# A BOOK RECOMMENDATION APPLICATION

## A BRIEF INTRODUCTION

For this year's software engineering project, our team has decided to create a book recommendation application. We believe that there is enough potential in this application to make a success of it for our clients which will solve their reading needs.

As this is application is focused primarily at book readers, we have therefore identified two main clients or users of this application:

. Students – This is because books are used as a material resource for their subjects and studies

. General readers – This is because many people read books outside of their working hours and this tool will be able to aid these fellow readers.

Students and general readers have a common link, they both need to use books for their desired purpose whether this is a student studying for his unit or a general reader interested in the works of Sir Arthur Conan Doyle. Each user in either case would find such an application like ours useful because it can easily allow the two users to either expand their knowledge of their subject area or indulge further into related works of another author.

By either circumstance, we hope that this application will serve the purpose of provide meaningful suggestions to related books so it makes finding new material simple and efficient. This application has arisen because of a recent user pole of how they would like to make their reading experience easier and more intuitive.

Our motivation for this project is based on this pole where several common thoughts emerged throughout the study:

."What are books are like this one?"

. "If only there were more books like this!"

. "So what can I read next?"

. "I know what I want but I still can't find it"

The thoughts above present to us the issues that people have been experiencing whilst reading either for leisure or for educational purposes.

## OUR OBJECTIVES PROPOSED

In spite of the recent discovery of existing systems and current user thoughts on the matter, we feel it would be best to listen to the users and provide them with a tool that will support their book reading experience not hinder it which executing are unfortunately contributing to.

For the user's problems to be properly actioned we have provided a list of objectives that we want to satisfy by the end of our project.

1. Address the issues raised by our user groups:
    (1) – Give meaningful suggestions to a particular book
    (2) – Provide meaningful suggestions to related works of the currently reading book
    (3) – Provide details on the book
    (4) – Provide reviews of the book that is currently being viewed
    (5) - Give meaningful information about pricing and where to find the book
2. Provide a system that is simple to use
3. Ensure that our system supports users first and is not cumbersome
4. Provide an organization facility for users
5. Ensure that a user always has material to read
6. Provide a recommendation facility for the user – This can be in the form of reviews and tagging

7. Provide an efficient system that puts the user first
8. Give routine suggestions to new books easy time the system is used

## OUR CONSTRAINTS

With most projects there will be several constraints that we will need to work with on order to successfully complete this project, we have outlined our constraints below:

1. Completing the work in the given time – We need to make sure that we meet our deadlines and milestones or otherwise the project will start to stagger and slow down and it may lead to the project becoming incomplete and unsuccessful.
2. Group management – We need to ensure that each member is allocated work that they can complete and that they actually complete this work set by the team leader. If a group member does not attend, we will have to allocate another team member to his/her position to for fill his/her role.
3. Resources – If we do not have the necessary resources to complete our tasks, then it would be difficult to continue with the project.
4. Quality – We need to ensure that enough time is spent with the right people using the right resources to ensure quality, this will be aided by a team leader.
5. Goal – We need to ensure that we stay true to our objectives and allow little deviation from them or otherwise we could never finish the project so it is important to follow our objectives.
6. Risks – We need to be weary of risk and therefore we will create a contingency plan in order to deal with risks and must follow this plan so risks are properly dealt with.

## PROPOSED SOLUTION

With the needs of our users above, we would like to address these problem by designing a book recommendation application that solves these problems. We would like to use the Waterfall development model as it provides us with a simple step-by-step process to follow in order to complete this project and is therefore the most suitable model for our project.

The book recommendation application will come in the form of a website where we will use the following languages to implement the solution to this problem:

. HTML – Website structure

. CSS –Website look and style

. JavaScript –Website interactivity and database

. AJAX – Dynamic database querying, part of our JavaScript

## PROJECT ORGANISATION

In this section we shall now outline the role of the various project members. The roles assigned will initially not change but are subject to.

We have identified 2 main roles for members in this project and our team is organized in the following manor:

1. Project/team leader – The team leader is responsible for the management of the various stages in our development model, settles disputes between team members and has the final say on any decision made by the project members. The team leader will also be accompanying the project members in each stage of development.
2. Operational/delegate – This member will carry the tasks instructed by the team leader in each stage of the development model and will report to the team leader weekly about progress made throughout the project and time spent.

   Role assignment:

| Name: | Role: |
|---|---|
| Spencer Murray | Team leader, delegate |
| Snehath Christopher | delegate |

| | |
|---|---|
| Ka Cheuk Yam | delegate |
| Mark Villar | delegate |
| Peter Russel | delegate |
| Oscar Beare | delegate |

## RISK ANALYSIS

In this section we shall state the risks that may come with this project and also provide solutions or plans to address these issues as well as their consequences on the team members and the project.

1. Time management – If we do not complete our work by the given deadlines and hit our milestones then we will be unable to complete the project. In order to mitigate this risk, we will meet up once a week (subject to change) so that the team leader is able to check-up on how we are progressing with each of the team members/delegates jobs.
2. Quality – If we do not ensure that there is quality is given by the amount of effort each member gives then the project will be incomplete and a basic version would be given and they may not satisfy our objectives. In order to mitigate this risk, the team leader in the meetings will check to see that objectives are being met and that we do not deviate from our objectives and tasks in the different section of our development model.
3. Group management – Communication is very important within teams, this is why we have all created a GitHub account to put our work on for discussion and have created a slack workspace in order for the team to communicate to each other at any time.
4. Resources – We need to think about what resources we have and be sure that we can use them well. If we had weak resources then completing the project would be difficult and cumbersome. This is why we have identify what text editors, browsers and server software we need in order to create and deploy this application unsuccessfully and to a minimum adequate standard.
5. Member sickness – If a member unfortunately reports as sick, then the work that was meant to be completed for that day or amount of time that the person is sick will not be completed. In order to mitigate this issue, each team member, including team leader, will be briefed about each task individually so that if a member does not attend due to a sickness, another one or more teams members will be assigned to the sick members' work for the time being and the sick member will be notified.
6. Designing the system - If we do not design the system properly, then we are open to vulnerabilities and design faults. This will mean that we will need to stop and rethink our design which will cost more time to get the system up and running. In order to mitigate this risk, we will check each decision with the team members and scrutinize as best we can to find any issues with our design before any implementation is undertaken.
7. Lack of change - If we are not open to change about any part of the project and do not let it evolve naturally then we will have limited visibility into potential impacts of our designs for example. In order to mitigate this issue, the team leader will employ change a core methodology for this project and encourage new ideas to be discussed and not let a question or idea pass by.
8. User data mining – If we do not properly mine for data from our users, then it would be impossible to create a complete and meaningful system. In order to mitigate this issue, we will carry out data collection on our users for this very reason which can be in the form of interviews or questionnaires.
9. Requirements do not fit objectives – We need to make sure that our requirements reflect our objectives. In order to achieve this, we will be asking questions in interviews, polls and questionnaires that mirror our objectives for the project and do not deviate.
10. Decision making – If we do not take on board each team members' decision and thoroughly think about and consider it i.e. Analysis, then we are open to delays of these designs that would impact the project. In order to adhere to a solution to this risk, we will discuss each decision and make an accurate assessment on whether to

undertake the decision or to declare it not useful or worthwhile to undertake and measure the possible impacts that it may bring.

## RESOURCE REQUIREMENTS

Here we will declare the resources that we need to complete this project.

Types of resource:

- Hardware
- Software

| Resource (Type is above): | Reason: |
|---|---|
| Desktop computer | To Design, document, implement, test and finally deploy the application. |
| Smartphone | Team communication, data recording (Requirements) |
| Camera | Record deployment and testing |
| Text editor – using Atom, Sublime | To code our website application |
| Code libraries | To allow us to write efficient code |
| VM – Virtual machine | To deploy our website and for testing |
| Browser – Chrome, Firefox, MS Edge, Apple Safari | To test and run our website, deployment facility |
| A word package | To document our project |
| A planning tool | To easily plan our project e.g. scheduling and work breakdown |

### Monitoring Mechanisms

We have identified several monitoring mechanisms that will allow us to meet our deadlines and milestones and manage the project.

Slack - This is a team communication tool that allows our team to easily communicate with each other in one place. Therefore we are able to discuss ideas, planning in the different parts of the project and serve as a place to go in order to support the management of this project.

Git & Github - A version control gives us the ability to track every version of the software that we are going to develop, it also gives us the flexibility to share and download code from each other.

Roles - Everyone in the group understands their role well and therefore it makes it easy to complete work as we all have a clear idea of our requirements from the project. Each persons is assigned a particular role or more more if needed from them and our team members are open to change and very adaptable to new ideas or changing requirements.

Contact and single-point of call - We have all agreed to meet at least once a week set out by our team leader in order to catch up on the work we have been completing. The team leader serves and a go to person for any issues/concerns or ideas that would be of concern/interest in the project.

A shared goal - Each team member has the same goal and wants to fulfil that goal by the end of the project, we would like to fulfil our objectives set out previously in this document and ensure that the project is completed on time and to a good quality.

# Project Schedule

| Project proposal and Plan - (03/10/2017 - 06/11/2017) - 25 Days | | | | |
|---|---|---|---|---|
| Activity: | Members Assigned: | Approximate time required: (Days) | Dependancy: | ID: |
| Introduction - Application domain, motivation, objectives and constraints | Spencer, Snehath Christoher | 3.1 | N/A | 1 |
| Introduction - Problem and proposed solution | Mark Villar | 1.24 | 1 | 2 |
| Project organisation - Roles and responsibilities | Peter Russell | 1.24 | N/A | 3 |
| Risk analysis - List of possible risks, consequences and strategies | Peter Russell, Oscar Beare | 3.72 | 1, 2 | 4 |
| Resource requirements - Hardware and software | Ka Cheuk Yam<br>Mark Villar | 2.48 | 1, 2 | 5 |
| Work breakdown | Peter Russell<br>Spencer Murray<br>Oscar Beare | 5.58 | 1, 2, 3 | 6 |
| Project schedule | Spencer Murray<br>Snehath Christopher<br>Mark Villar | 5.58 | 6 | 7 |
| Monitoring mechanisms | Mark Villar<br>Peter Russell | 1.86 | 4, 5 | 8 |
| **Requirements elicitation - (07/11/2017 - 22/11/2017) - 12 Days** | | | | |
| Gather user requirements | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 3.72 | Project proposal and Plan | 9 |

| | | | | |
|---|---|---|---|---|
| Translate to system requirements | Mark Villar<br>Peter Russell | 1.24 | 9 | 10 |
| SRS - Introduction, existing systems and outline of SRS definition | Snehath Christopher<br>Peter Russell<br>Spencer Murray | 2.48 | 10 | 11 |
| Requirements method | Oscar Beare<br>Ka Cheuk Yam<br>Peter Russell | 1.24 | 10 | 12 |
| Requirements definition | Snehath Christopher<br>Mark Villar | 3.72 | 9, 10, 12 | 13 |
| **Design - Part 1 - Before Christmas Break - (23/11/2017 - 23/12/2017) - 32 Days - Including weekends** | | | | |
| Website structure and functionality - HTML Modules | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 12.4 | N/A | 14 |
| Javascript- Interactivity Modules | Mark Villar<br>Peter Russell<br>Spencer Murray | 4.96 | 14 | 15 |
| Layout | Oscar Beare<br>Ka Cheuk Yam<br>Spencer Murray<br>Snehath Christopher | 6.2 | 14 | 16 |
| Overall architecture design and use cases | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 5.96 | 14, 15 | 17 |
| **Design - Part 2 - Semester 2 - (01/02/2018 - 19/02/2018) - 18 Days - Including weekends** | | | | |
| Database query module - Javascript | Mark Villar<br>Peter Russell<br>Spencer Murray | 4.58 | 17 | 18 |
| Website styling | Snehath Christopher<br>Ka Cheuk Yam<br>Oscar Beare | 6.2 | 14, 16, | 19 |
| DFD | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 3.72 | 14, 15, 17 | 20 |

| | | | | |
|---|---|---|---|---|
| ERD | Snehath Christopher<br>Spencer Murray | 3.72 | 20 | 21 |
| Use case | Snehath Christopher<br>Spencer Murray | 2.48 | 21 | 22 |
| **Implementation - (20/01/2018 - 15/02/2018) - 19 Days** | | | | |
| Create Javascript modules | Mark Villar<br>Peter Russell<br>Spencer Murray | 4.34 | 15, 17, 18 | 23 |
| Create database and query modules | Snehath Christopher<br>Mark Villar<br>Peter Russell<br>Spencer Murray | 4.34 | 15, 17, 18, 20, 21 | 24 |
| Create website structure, styling and layout (All HTML and CSS module designs implemented. | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 8.06 | 14, 15, 16, 17, 18, 19, 20, 23, 24 | 25 |
| JQuery and Javascript linking | Spencer Murray | 1.24 | 25 | 26 |
| **Prototyping - (16/02/2018 - 05/03/2018) - 12 Days** | | | | |
| Run Javascript module(s) and record results and test runs | Mark Villar<br>Peter Russell<br>Spencer Murray | 3.1 | 25, 26 | 27 |
| Run JQuery module(s) and record results and test runs from supplied data | Mark Villar<br>Peter Russell<br>Spencer Murray | 3.1 | 25, 26 | 28 |
| Run website and check scripts are linked and test and record results | Spencer Murray | 0.62 | 25, 26 | 29 |
| Run website and check, test and record functionality | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 5.58 | 29 | 30 |
| **Deployment - (06/03/2018 - 13/03/2018) - 6 Days** | | | | |
| Deploy website on VM and check it works through HTTP | Spencer Murray | 0.62 | 30 | 31 |

| | | | | |
|---|---|---|---|---|
| Run website on VM and check for issues | Snehath Christopher<br>Ka Cheuk Yam<br>Mark Villar<br>Peter Russell<br>Spencer Murray<br>Oscar Beare | 5.58 | 31 | 32 |
| | | | | |

# Work breakdown

- Constructing a book reccomendation applictaion — Work: 100%
  - 1. Project Proposal and plan — Work: 20%
    - 1.1 - Introduction
      - 1.1.1 - Applictaion domain, objetcives, motivation and constraints — Work: 2.5%
      - 1.1.2 - Problem and proposed solution — Work: 1%
    - 1.2 - Project oragnisation
      - 1.2.1 - Roles and responsibilities — Work: 1%
    - 1.3 - Risk Analysis
      - 1.3.1 - List of possible risks, consequences and strategies — Work: 3%
    - 1.4 - Resource requirements
      - 1.4.1 - Hardware and software requirments — Work: 2%
    - 1.5 - Work breakdown
      - 1.5.1 - Work breakdown diagram — Work: 4.5%
    - 1.6 - Project schedule
      - 1.6.1 - Project schedule outlining activity dependancies — Work: 4.5%
    - 1.7 - Monitoring mechanisms
      - 1.7.1 - Strategies to meet deadlines — Work: 1.5%
  - 2. Requirements — Work: 10%
    - 2.1 - Gather requiemenrents
      - 2.1.1 - User requirements — Work: 3%
      - 2.1.2 - Translate to system requirements — Work: 1%
    - 2.2 - SRS Document
      - 2.2.1 - Introdution — Work: 2%
      - 2.2.2 - Requirements method — Work: 1%
      - 2.2.3 - Requirements definition — Work: 3%
  - 3. Design and Implement — Work: 55%
    - 3.1 Design — Work: 40%
      - 3.1.1 - Architecture
        - 3.1.1.1 Overall Architecture design — Work: 4.8%
        - 3.1.1.1 Back end — Work: 20%
          - 3.1.1.1.1 - Javascript
            - 3.1.1.1.1.1 - Interactivity modules — Work: 4%
            - 3.1.1.1.1.2 - Database query modules — Work: 3.7%
        - 3.1.1.1.3 - Database
          - 3.1.1.1.3.1 - DFD — Work: 3%
          - 3.1.1.1.3.2 - ERD — Work: 3%
          - 3.1.1.1.3.2 - Use case — Work: 2%
      - 3.1.1.2 Front end — Work: 20%
        - 3.1.1.2.1 - User Interface
          - 3.1.1.2.1.1 - HTML
            - 3.1.1.2.1.1.1 - Website structure and functionality — Work: 10%
          - 3.1.1.2.1.2 - CSS
            - 3.1.1.2.1.2.1 - Layout — Work: 5%
            - 3.1.1.2.1.2.2 - Styling — Work: 5%
    - 3.2 Implement
      - 3.2.1 Back end — Work: 6.5%
        - 3.2.1.1 Create interactivity modules — Work: 3.5%
        - 3.2.1.2 Create database query modules — Work: 3.5%
      - 3.2.2 Front end
        - 3.2.2.1 Create website - Structure and styling and layout — Work: 6.5%
        - 3.2.2.3 Link database using Javascript module — Work: 1%
  - 4. Prototype/testing — Work: 10%
    - 4.1 - Back end — Work: 5%
      - 4.1.1 - Run Javascript module, record results and change if nescessary — Work: 2.5%
      - 4.1.2 - Run Javascript with data module, record results and change if nescessary — Work: 2.5%
    - 4.2 - Front end — Work: 5%
      - 4.2.1 - Check Webiste loads including pages and scripts — Work: 0.5%
      - 4.2.1 - Check page by page functionality, record test runs and change if necessary — Work: 4.5%
  - 5. Deployment — Work: 5%
    - 5.1 - Run site on VM server, check for issues and correct — Work: 4.5%
    - 5.1 - Check site works through http — Work: 0.5%


Activities Schedule - 2017-2018
Start dates and durations
03-Oct-17
23-Oct-17
12-Nov-17
02-Dec-17
22-Dec-17
11-Jan-18
31-Jan-18
20-Feb-18
12-Mar-18
Activities
Introduction - Application domain, motivation, objectives and constraints
3.1
Introduction - Problem and proposed solution
1.24
Project organisation - Roles and responsibilities
1.24
Risk analysis - List of possible risks, consequences and strategies
3.72
Resource requirements - Hardware and software
2.48
Work breakdown
5.58
Project schedule
5.58
Monitoring mechanisms
1.86
Gather user requirements
3.72
Translate to system requirements
1.24
SRS - Introduction, existing systems and outline of SRS definition
2.48
Requirements method
1.24
Requirements definition
3.72
Website structure and functionality - HTML Modules
12.4
Javascript- Interactivity Modules
4.96
Layout
6.2
Overall architecture design and use cases
5.96
Database query module - JavaScript
4.58
Website styling
6.2
DFD
3.72
ERD
3.72
Use case
2.48
Create Javascript modules
4.34
Create database and query modules
4.34
Create website structure, styling and layout (All HTML and CSS module designs...
8.06
Database and Javascript linking
1.24
Run Javascript module(s) and record results and test runs
3.1
Run JQuery module(s) and record results and test runs from supplied data
3.1
Run website and check scripts are linked and test and record results
0.62
Run website and check, test and record functionality
5.58
Deploy website on VM and check it works through HTTP
5.58
Run website on VM and check for issues
0.62

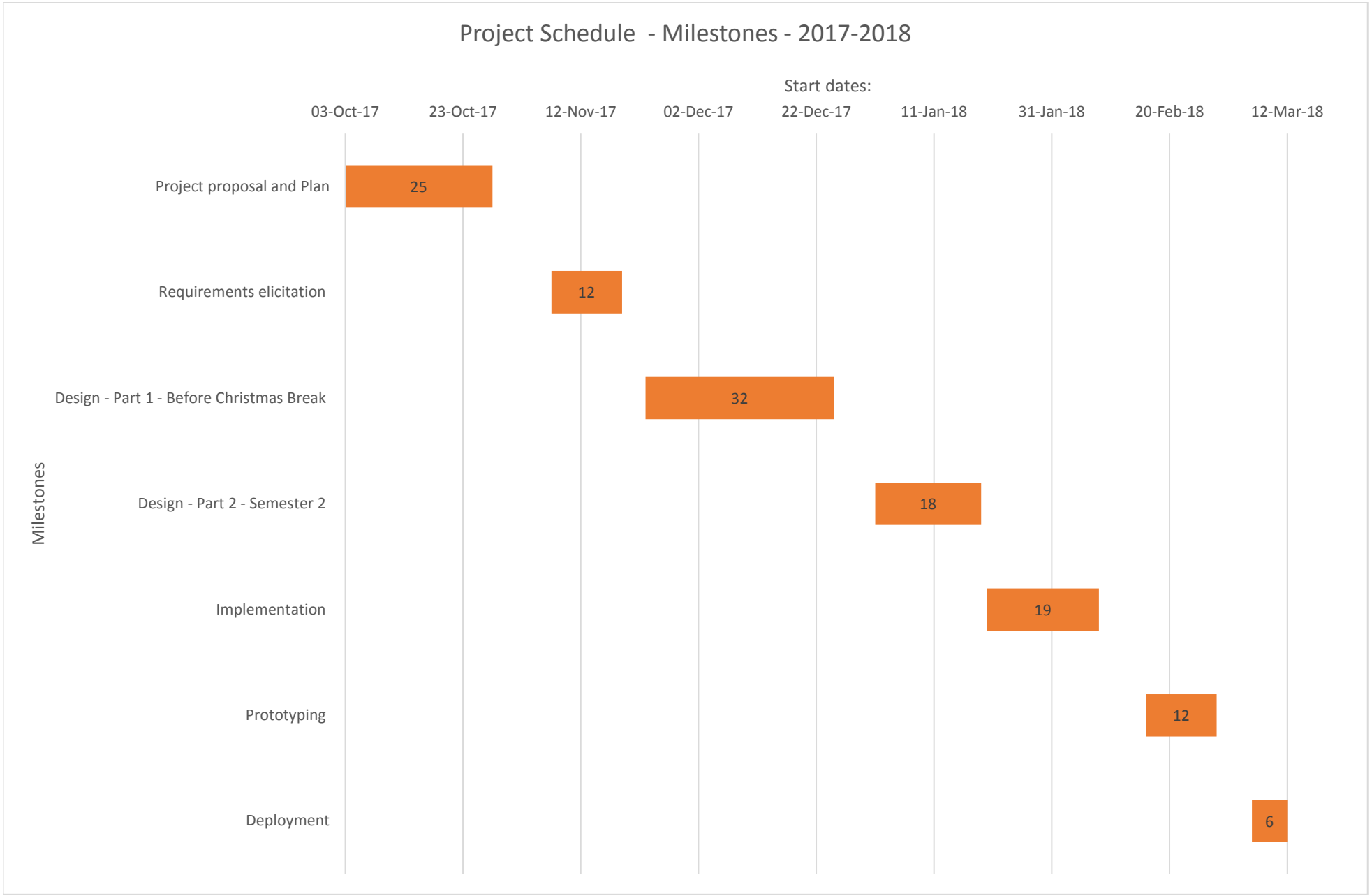

Project Schedule - Milestones - 2017-2018
Start dates:
03-Oct-17
23-Oct-17
12-Nov-17
02-Dec-17
22-Dec-17
11-Jan-18
31-Jan-18
20-Feb-18
12-Mar-18
Milestones
Project proposal and Plan
25
Requirements elicitation
12
Design - Part 1 - Before Christmas Break
32
Design - Part 2 - Semester 2
18
Implementation
19
Prototyping
12
Deployment
6

Table of Contents:



## Authors and contributors:

| Name: | Contact: | Organisation: | Description: |
|---|---|---|---|
| Snehath Christoher | Up691978@myport.ac.uk | MetBook | Author, Introduction , preface |
| Ka Cheuk Yam | Up8757588@myport.ac.uk | MetBook | Author, Introduction |
| Mark Villar | Up773229@myport.ac.uk | MetBook | Author, User requirements |
| Peter Russell | Up806925@myport.ac.uk | MetBook | Author, User requirements |
| Spencer Murray | Up811216@myport.ac.uk | MetBook | Author, Functional and non-functional requirements |
| Oscar Beare | Up825319@myport.ac.uk | MetBook | Author, Functional and non-functional requirements, requirement methods |

## Revision history:

| Version: | Revision: | Date: | Description: | Author: |
|---|---|---|---|---|
| 1 | 1 | 18/11/2017 | First draft | 811216, 825319, 806925 |
| 1 | 3 | 23/11/2017 | Added requirements | 811216, 825319 |
| 1 | 4 | 25/11/2017 | Update user requirement | 811216, 825319,773229 |
| 1 | 5 | 27/11/2017 | Reviewed requirement methods | 811216, 825319, 8757588 |
| 1 | 7 | 28/11/2017 | Completed review | 811216, 825319, 691978 |

## Introduction:

Purpose of this Document

The purpose of this Software Requirements Specification (SRS) is to illustrate the requirements for the “Book recommendation” software. The SRS document is primarily intended to be used as a proposal to the client. It will outline the idea and need for the development of the software. Furthermore, describe the functions and constraints of the system.

Scope

From our research, we have found that primary goal of this software is to keep people interested in reading. In trying to achieve that goal, the software surrounds itself around the idea of “giving people exactly what they like”. Simply, recommend books that one will want to read.

The “Book recommendation” is a web-based application. It suggests and recommends books that users may love reading based on their search results, search history and, books that are related to ones currently in their personal library. Another goal is to reduce the troubles people face in trying to find the details of a book. Furthermore, the primary target audience for the software are students, professors, and book lovers.

Additionally, the software requires internet and a database (of books) to obtain and display results. All information such as interests, likes and dislikes, as well as personal information of users will be stored on a client database. This will allow the system to recommend books that are better suited for each person.

## System functions:

User will be able to use a search facility with its filtering options to find related searches, the user will then meet a results screen. From here the user will be able to view the results and order them, and filter the results to their desire. The user will be able click on any results and the view the information about the book. The information includes:

. Title
. Description
. Image
. Rating
. Reviews

For account users, they will have a dashboard where they will be able to view recent recommendation, new recommendation, their reviews, their ratings and their suggestions. The user will be able to have a settings menu where they can change the look of results i.e. how many are shown and filter results and searches based on their suggestions, recommendation lists.

Users will be able to write reviews and rate books from 1-5 (5 Being most liked). User information such as their searches will be saved for further searches for non-account users and for account users, to assist in recommending new books on the user-account dashboard. Users with accounts can have more options on the content they wish to see, including the breadth of the search function. They may search for books based on their own recommend categories and suggestion list or choose standard search functionality and instead just use the search term and any other filtering options applied to the search.

## Glossary of terms:

| Term: | Meaning: |
|---|---|
| Dashboard | A central location or home area that acts a user’s main point of call. |
| Search facility | A input field used to find books and obtain results from the book library |
| HTTP | Hyper-text-transfer-protocol – Used to allow websites to connect to a serve rand obtain webpages. |
| Suggestions list | A list of search terms/categories that have been recommend to the user. |
| Header | The top area of a webpage containing useful navigational features. |
| Footer | The bottom area of a webpage containing navigational and help information. |

## Current Systems:

Currently there are a number of existing systems that are in place to serve this purpose for readers. However several problems have been identified which are common to most systems right now.

. System does not provide meaningful related material – For example, the google play store or the apple store do have such a feature but often, the suggestions that are given are quite poor and often unrelated to the user

. System provides a poor suggestion response to currently reading book – Again with these two stores as an example, the current book that you are reading does not give the user other meaningful suggestions to read other material.

. System fails to differentiate between general user and student – The problem that both user groups are facing here is that as they are effectively treated as the same, it is difficult to organize a student’s books that they are reading with more general books so it makes organization of a users’ books for certain purposes more difficult to manage therefore leading to a more difficult user experience and interface.

# Requirement methods

## One-to-one interviews

It is important that we gather information from users that will be using our software, this is because at the end of the day they are the ones that will be using the system. One-to-one interviews are a fantastic way in order to gain the relevant information to make our system the way that users will find useful.

It is important that the right questions are asked when conducting the one-to-one interviews, this is to avoid getting the wrong or irrelevant information out of the interviewees and making sure that we are receiving the relevant information or new information. Asking the wrong questions will only lead to wasted time and in the real world lost money.

We will conduct through setting up interviews and asking thoughtful and meaningful questions. If the interviews are a success we will have a bank of user needs. This is also meaningful as we will have real users tell us what they want. The interviews will be a lengthy process and so it should be, this is where we will be assigning our requirements and also confirming many user requirements.

### Questionnaires

This is not a particularly good method at gaining information; this is due to the fact that the users can lie on the questionnaires. It is also a possibility that the developer could be asking the wrong questions and not know it until the questionnaires are collected, this problem would mean that the questionnaire is null as there will be no useful information on it. Or information that could be betrayed in the wrong way. Questionnaires are not seen as good requirement method building methods; this is why we will not be implementing this method.

### Use cases

During this method you create multiple personas in order to implement a user; this will then be used to see what requirements these types of people may need. The personas need to be many and varied greatly in order to see the most crucial requirements.

This requirement method is all down to the developer, it is up to them to create enough personas and make them varied from one another; if the developer hasn't created enough personas than there could be many different requirements missed out. This is why this method can be a poor example of the users of the system.

This could prove to be an interesting way of creating user requirements. Allowing us - the developers - to get an idea of how possible users may use the system. This method will also be saving the developer's time as they will be able to create a vast amount of personas in the time that it would have taken to interview one person.

The limitation to this method is that the developers may miss out multiple personas, there for missing out on many requirements. Another limitation is that the developer will have to be thorough in their research of each persona as they may miss out many needs that the user may have. This method is completely down to the developer.

This is a requirement method that will be implemented by creating multiple personas, the more personas the more information that we will create. The personas will have to be complex in order to make the most out of this method and gain all the necessary requirements that users will have. We will be able to create multiple user requirements and these will be turned into technical requirements.

### Prototype

Prototype gives our users a very rough look and feel of the functionalities and what the final product would look like, this also helps our fellow developers to show proof of concept to other developers (if need be).

### Social & Interface analysis

We have passively observed our target users from using existing tools, applications, and services from the internet that provides somewhat similar functionality, what features needs to be added or can be improved in order to improve our target users' experience. This helped us to understand how the users use existing products, understand the flaws and restrictions of current existing tools i.e. what works and what does not work, how we would create the user interface to make it as easy and accessible for our users to navigate as possible, which functionalities are appropriate, this also taught us some ideas that users would otherwise would not be able to point out or would find it hard to point out because of technicalities involved.

## User Requirements:

| ID: | Requirement: | Reason: | Source: |
|---|---|---|---|
| 1 | There should be a search facility which allows multiple filter options such as publisher, ISBN. | To gain more accurate searches. | Interview question response – "What features must be included?" |
| 2 | The home page should include information about new books and the categories that the library has to offer. | To keep the site fresh and always provides new recommendations to users. | Interview question response – ."What features must be included?" . "What do you expect the home page to include" |
| 3 | The home page should include a search facility which must include its own filtering options. | Users should be able to search from anywhere in the site. | Interview question response – "What do you expect the home page to include" |

| | | | |
|---|---|---|---|
| 4 | A header and footer is needed with a search bar and accessibly bar in the header. | This is standard practice with most modern websites. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 5 | There must be an accessibility options menu in the header. | This is standard practice with most modern websites. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 6 | The website should return correct and accurate, related results and work as expected. | Our aim to provide a smooth experience that does function as expected. | Interview question response – "What are your expectations when you search for an item?" |
| 7 | Must be allowed to change a search term or remove one and then search again. | The user must be in charge at all times and modify their search as they desire. | Interview question response – "What are your expectations when you search for an item?" |
| 8 | Search options must include category filtering, search term filtering and publisher, ISBN etc. filtering. | To give more related searches that help the user's journey on to finding new books. | Interview question response – "What kinds of options should you have when searching for a book?" |
| 9 | There should be appropriate error messages for all errors on that can occur e.g. a search returns false and an appropriate error message must show. | We need to tell the user when something goes wrong and these messages must help the user to move on. | Interview question response – "What types of errors do you expect to be presented to you?" |
| 10 | Results should show a quick section of the results found and then should be put into separate pages for viewing. | This is standard practice with most modern websites. | Interview question response – "What are your expectation for the layout of the results page, how should it respond?" |
| 11 | The information shown on the results page should be minimal, only a partial amount of information about a book should be shown. | This is standard practice with most modern websites. It important to show only a snapshot of information as the user is only viewing results. | Interview question response – "What are your expectation for the layout of the results page, how should it respond?"<br><br>Google play store reviews |
| 12 | At all times, the user must be able to return to the previous page or home page. | This is standard practice with most modern websites. | Google play store reviews – "Difficulty with navigation" |
| 13 | Item result should be one full page and show, an image, some description and any other necessary information. | We need to show all information on each item here. | Interview question response – "What are your expectation for the layout of the results page, how should it respond?" |
| 14 | The rating and reviews must be shown on each book. | We need to show all information on each item here. | Interview question response – "In terms of ratings and reviews, how should this be presented to the user?" |
| 15 | Each account user must see their recommendations, rated books and new books. | Account users must be able to access their information in one place. | Interview question response – "What information do you expect to see in your account from this type of website?" |
| 16 | There should be a settings menu where can change the type of search you want i.e. in-depth. | Account users are entitled to a settings, configuration menu to change the site to best suit their preferences. | Interview question response – "What information do you expect to see in your account from this type of website?" |
| 17 | There should be a simple login and log out button that is clear and easy to find. | This is standard practice with most modern websites. | Interview question response – "What information do you expect to see in your account from this type of website?" |
| 18 | Extra features like pdf must be shown but unable to use when in non-user mode. | Premium users or account users are entitled to this feature. | Interview question response – "We are thinking of a premium feature such as pdf viewing for account users, what do you think of this from an account users perspective and a non-user accounts' perspective?" |
| 19 | A user should be able to save suggestions and then view them later on for searching. | Account users have extended functionality. | Interview question response – "What types of account functionality could a user have against a non-user account?" |
| 20 | The site should accept new users and their accounts. | This is standard practice with most modern websites. | Interview question response – "Would it be useful if we had accounts, and why should we have this?" |

| | | | |
|---|---|---|---|
| | | | Google play book store reviews – "Accounts and improvements" |
| 21 | Account users can write reviews and rate book and also recommend. | Account users have extended functionality. | Interview question response – "What types of account functionality could a user have against a non-user account?" |
| 22 | The site must work on all devices and re-flow itself accordingly. | This is standard practice with most modern websites. | Interview question response – "In terms of scope of multiple devices, should the site be usable on all devices and what do you expect from these different views of the site?" |
| 23 | The layout must be easy to use and clear. It must only show it is must and nothing more. | This is standard practice with most modern websites. | Interview question response – "What kind of layout do you expect from this type of site"? |
| 24 | It should be clear that I am a non-account user and account user. Different versions of the site should be displayed depending on user type. | This is standard practice with most modern websites. | Interview question response – "What types of account functionality could a user have against a non-user account?" |
| 25 | Multiple users may use the site. | This is standard practice with most modern websites. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 26 | The site should load quickly enough and respond to search in an efficient amount of time. | This is standard practice with most modern websites. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 27 | The site should include accessibility and this function should be easy to use. | This is standard practice with most modern websites. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 28 | Searching should happen at any time no matter where you are in the website. | This is standard practice with most modern websites. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 29 | Hide and show information on each page should be optional. | To allow the user to control what they want to see. | Interview question response – "What are your expectation for the layout of the results page, how should it respond?" |
| 30 | The site should allow users to visit from external locations on the internet and work in a similar way when updates are applied. | This is standard practice with most modern websites. User experience is not affected. | Interview question response – "What do all websites have in common and what do you expect to include in these common areas?" |
| 31 | There should be a help section for new users and existing users. | This is standard practice with most modern websites. | Interview questions response – "What do all sites have in common?" |

## Functional requirements

Continue on next page

## Home dashboard:

| ID: | Requirement: | Reason: | Source: |
|---|---|---|---|
| 1 | Must include a list of 10 of the most recently recommended books based on user rating. | To give users an idea of what books are in the top rating. Gives user an idea of what new books are available. | 2 |
| 2 | Must include a search facility. | To enable users to find new books from the home screen. | 3 |
| 3 | Must include the categories of books that the library holds. | To enable simple and efficient and accurate filtering. (Pages) | 2 |
| 4 | Must show the next 10 new books that have been added to the library. | To provide new ideas of new books that the user can view. | 2 |
| 5 | Must show a snapshot of each category on the home dashboard<br>And include 10 books from this category to the user. | To provide the user with a flavor of each type of book in each category. | 2 |
| 6 | Must include an accessibility settings bar. | To allow all users to user to site to its full extent. | 5 |
| 7 | Must include a filter to search for a category on the home dashboard. | Accurate and efficient search filtering. (Search and results) | 3 |
| 8 | Must include a header and footer. The header must include a title, its category links, and accessibility bar, search bar. The footer must include site information including, categories for books, link to top of page and to its help pages. | Ease of user, efficient search, all users can gain full extent from the site. Standard website practice. | 4 |

## Search facility:

| | | | |
|---|---|---|---|
| 9 | Must include filtering options. This includes filtering by, publisher, author, and ISBN and search term. | Reliable, accurate and efficient search to achieve retrieving related books. | 1, 3 |
| 10 | Must return the appropriate response to the search. The search function must only retrieve related findings which are related to the search request only. | The search function must only work as intended and achieve, at minimum, related pages, items and links to the desir3d search term and its filtering options. | 6 |
| 11 | Must allow the user to modify their search at any time. The user must be allowed to re-order their filtering settings or change the search term. | To allow the user to find what they want and retrieve reliable results. | 7, 8 |
| 12 | Must allow the user to enter multiple search terms in one search. | Reliable and related results. | 7, 8 |

| | | | |
|---|---|---|---|
| 13 | Must allow the user to change their search term at any time in the site. | Standard practice in most websites, ease of user for the user. | 7 |
| 14 | Must save the user's search for that session and show these recent searches. (Temporary session) | Standard practice in most websites, ease of user for the user. | 8 |
| 15 | Must be allowed to search from a category. | Accurate and reliable related results finding. | 8 |

Results page:

| | | | |
|---|---|---|---|
| 16 | Must display an appropriate error message if no books have been found and then provide related search that are similar to the user's search term. | Standard practice in most websites. | 9 |
| 17 | Must show the first 10 results. | Standard practice in most websites, ease of user for the user. We do not want to overload the user with results. | 10 |
| 18 | Must allow the user to view from 10 to 100 results per page. | Providing options to the user to help them with their search. | 10 |
| 19 | Must provide a snapshot of the book information in each result found. The information in each search result must include: title, brief description, rating, recommended number of times and its image. | Providing only necessary information at this stage for searching, the user can go the results screen for more information. | 11 |
| 20 | Must allow the user to filter the results by ISBN, author, title, rating, and publisher. | Accurate, reliable fileting of results for related items to search term. | 1, 7, 8 |
| 21 | Must allow the user to return back to the home screen with a simple button. | Standard practice in modern websites, ease of use. | 12 |
| 22 | Must show the amount of pages of results that have been found. | Standard practice in modern websites, ease of use. | 10 |
| 23 | Must provide related search terms after a search has been made. | Ease of use, more accurate filtering for results. | 1, 6 |
| 24 | Must allow the user to filter the results by category, site re-displays results | Filtering results so the user can easily find a book by category. | 1, 7, 8 |
| 25 | Must display the results in several forms. These forms are tile, list. | User preference, the user can display the results in the form that they prefer most. Ease of user accessibility. | 11 |
| 26 | Must display the header and footer at all times. | Standard website practice. | 4 |
| 27 | Must allow multiple searches to take place. | User experience, not making the use return to the home screen to make a new search. Ease of use. | 1, 3, 7, 8 |

| | | | |
|---|---|---|---|
| 28 | Must show the image of the book. | Providing appropriate and expected information to the user. | 13 |
| 29 | Must show a description of the book including its blurb. | Providing appropriate and expected information to the user. | 13 |
| 30 | Must show its rating. | Help in recommending a book and it being viewed by the user. | 14 |
| 31 | Must show its reviews. The reviews show are 5 at first then more if necessary. | To ensure that the user receives a balanced snapshot of reviews. Aids recommendation of the book and user finding a book. | 14 |
| 32 | Must show places where this book can be found. | To allow the user to easily find the book after using the website. | 13 |
| 33 | Must show the app's availability of this book. Users in this mode cannot use the book in the app, only the availability is shown. This is a PDF version. | To show that we do have the PDF version available but premium users are entitled to this feature. | 13, 18 |
| 34 | Must allow the user to return the results page and the home dashboard. | Standard website practice. | 12 |

Account features:

Account dashboard:

| | | | |
|---|---|---|---|
| 35 | Must provide a list of the books that the user has recommended. | To provide the user with their history of recommended books. | 15 |
| 36 | Must provide a list of books that have been rated by the user. | To provide the user with their history of rated books. Filtering for previous books that have recommended. | 15 |
| 37 | Must provide a list of new books that are related to books that the user has already recommended. This list includes 5 new books. | To allow the user to easily find new books, can expand to show more books if required. | 15 |
| 38 | Must include a suggestions section. | To allow the user to find new books but with their related history in the suggestions. | 15, 19 |
| 39 | Must include a settings menu where the user can change, the amount of suggestions, layout, colors, level of search term depth (absolute, similar and loosely-related. | Standard website practice, allows the user gain the full experience from the site. | 16 |
| 40 | Must include a log-out button. | Standard website practice. | 17 |

Login and sign-up:

| | | | |
|---|---|---|---|
| 41 - Login | Must include appropriate error handling for logging in. If a user does not exist, this message must be shown. | Standard website practice. | 9 |
| 42 - Login | Must load the user's account dashboard upon login | Standard website practice. This is for existing users only. | 6, 9 |
| 43 - Register | Must include appropriate error handing for registering. All fields must be validated before the information is sent to the server. | Standard website practice. If no issues and can be added to database, then add, else a validity check is wrong. | 20 |

Search facility:

| | | | |
|---|---|---|---|
| 44 | Must allow user to search by their recommended list categories. | To save user time and allow finding new books with their own recommend list. Ease of use. | 15, 19 |
| 45 | Must allow user to search by suggestions. | To save user time and allow finding new books with their own suggested list. Ease of use. | 19 |
| 46 | Must allow a custom amount of search results to appear. | The user should be allowed to change how many search they wish to see, top 5, 30, all etc. | 16, 19 |
| 47 | Must be allowed to filter by suggestions, user recommendations and user saved search terms. | To save user time and allow finding new books with their saved search terms, suggestions and recommendation list. Ease of use, faster filtering. | 19 |

Results page:

| | | | |
|---|---|---|---|
| 48 | Result screen - must allow user to write reviews and rate the book and also recommended | User account extended functionality. | 21 |

## Non-functional requirements:

| ID: | Requirement: | Reason: | Source: |
|---|---|---|---|
| 1 | The dashboard must load in an efficient amount of time. This is within 2-5 seconds. | Standard website practice and expectation. | 26 |
| 2 | Must include a predictable interface. | Standard website practice and expectation. | 23 |
| 3 | Must include a responsive design | Standard website practice and expectation. | 22 |
| 4 | The layout must display items clear and readable manner | Standard website practice and expectation. | 23 |
| 5 | The site must work on all types of devices. | Standard website practice and expectation. | 22 |
| 6 | An account user must only be able to access their profile and not any others. | Standard website practice and expectation. | 24 |

| | | | |
|---|---|---|---|
| 7 | A search must return related results and become more accurate as more filters are applied. | To allow users to quickly and easily find new books. | 6 |
| 8 | The site must account for a large number of users at the same time. The server must be able to respond to at least 100 users simultaneous. | Standard website practice and expectation. | 25 |
| 9 | A user must be able to change accessibility options. These include: high-contrast, larger text size, simple colors | Standard website practice and expectation. User experience. | 27 |
| 10 | A user must be able to search for a book at any time. | Standard website practice and expectation. | 28 |
| 11 | A user must be able to ask for help and refer to a help page. | Standard website practice and expectation. | 31 |
| 12 | Account dashboard - User must be able to view the information that they desire and hide any information that they do not wish to see. | User experience, providing the appropriate levels of options to the user. | 29 |
| 13 | Account dashboard - Each section shows a snapshot of the full section, expansion allowed to full page of section. | User experience, providing the appropriate levels of options to the user. | 10 |
| 14 | The site should perform the same no matter updates | Standard website practice and expectation. | 30 |
| 15 | Each users' data can only be accessed through their details | Standard website practice and expectation. | 24 |
| 16 | Each section of the website must only display information that the user expects, a results page should not show the home dashboard. | Standard website practice and expectation. The user should only view what they require, the site should not give unexpected response. | 6 |
| 17 | Each mode of operation for each user must show. The site must display this difference. | To enable the difference between non-user account and account user. | 24 |
| 18 | The site must be able to run through http and display the appropriate page for what the user expects. | Standard website practice and expectation. | 20 |
| 19 | The user must be able to change the size of text and items on each page to their desire. Accessibility provides automatic enhancement. | Accessibility, ease of use, and user options. | 27 |
| 20 | The website should let users to decide the type of search they want, accurate using previous information or a standard search for the library using the standard functions. | Provide user options on search to retrieve new and different results than a previous search. Increase breadth of search. | 8 |

# Design

For our architectural pattern for our web application, we need to consider the following problems:

. How will the user access our services?

. How can the system account for multiple users?

. How will data be controlled from one entity to another?

. How will security be handled in the system?

If we are to develop a capable web application then we need to design a system that will:

. Allow users to easily access services

. Allow multiple users at the same time

. Ensure that data is efficiently handled in the system.

. Incorporate the right level of security for our system.

It wouldn't be sufficient for our system to allow 3 factors for example, letting users access services; allowing multiple users and efficiently handling data without the right level of security as this would mean that our system would be at risk to malicious behaviour and user's data would be at risk.

To address these issues we thought of 3 different design models:

. MVC – Model View Controller

. Client-server architecture

. Layered

MVC – This architecture does look suitable for our needs as we could easily find sufficient components to model this type of system e.g. the interface could be the view, the model and controller would fall down to our servers but the problem with this architecture for our needs is that it would require extra complexity where it is not needed. We would have separate model-view-controllers for user front and back end which is unnecessary.

The layered architecture is somewhat interesting as we could have our database at the base layer/bottom and data is passed upward to the users and request downwards. Even though this also seems a viable option for us, it is also adds unnecessary complexity and does not help to ensure we meet our user requirements well because our last design idea offers a healthy grouping of components whereas this architecture does not.

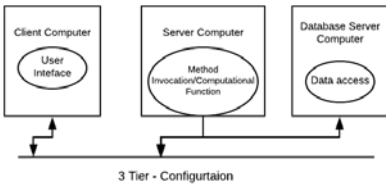
Our final architectural design is the client-server style which even though it was our initial design, the standard client-server architecture was not suitable for us because in a 2 tier client-server architecture, the database is handled along with the web server requests on the same server, this poses security risks therefore we have decide to design a 3 tier-client server architecture to ensure that security of user's data is sufficient i.e. our data is handled on a separate server so this adds another level of security. This means that we have our Front-end interface, a web application server and a database server.

We went with this architecture for the following reasons:

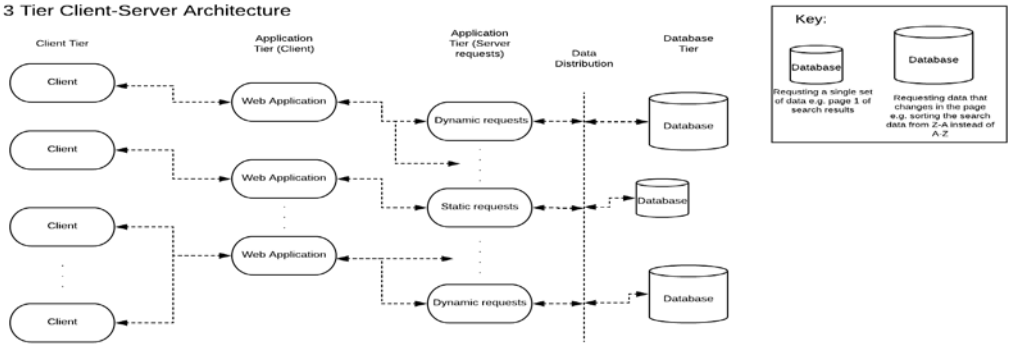
. It address our problems above efficiently

. It allows us to meet our user requirements

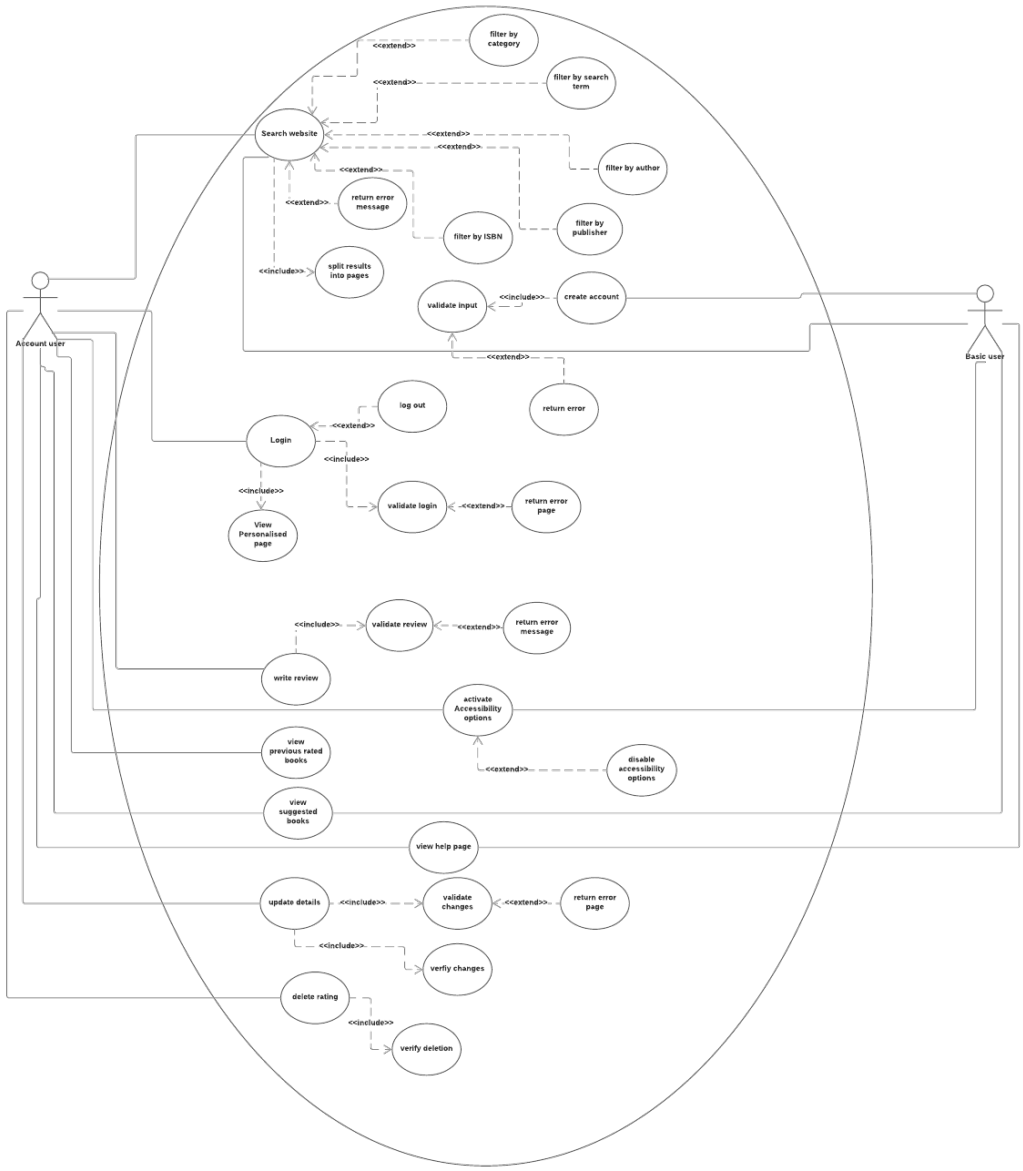
. It answers the question on security




Proposed architecture



Use cases.




**Use case name:** login
**Brief description:**
This use case describes how the user will login to their account
Actors using this use case : Account user
**Flow of events: The use case begins when the user types their login details into the login form**
**Basic flow:**
1.The system will validate the users login details on the login form
2.The system will display the user's personalised page and the use case will end
**Alternative flows:**
1.invalid login
If the system cannot find an account matching the provided login details an error message will appear. The actor can either attempt to login with different details or cancel the operation, ending the use case
**Special requirements: n/a, Pre conditions:n/a, Post conditions: n/a, Extension points: n/a**

**Use case name: Adjust size of site**
**Brief description:**
This use case describes how the user can just the size of the website
Actors using this use case : Account user, Basic user
**Flow of events: The use case begins when the user adjusts the size of the window**
**Basic flow:**
1.The website will reflow and the formatting will remain unchanged
2.The system will display the current page with the correct styling and the use case will end
**Alternative flows: n/a, Special requirements: n/a, Pre conditions: n/a, Post conditions: n/a, Extension points: n/a**

**Use case name:** write review
**Brief description:**
This use case describes how the user can add a review for a book
Actors using this use case : Account user
**Flow of events:**
The use case begins when the user presses submit on the review page
**Basic flow:**
1. The system will validate the review checking for invalid characters, duplicate reviews and character limit
2. The system add the review to the book and the use case will end
**Alternative flows:1**
1.Invalid review
If the system find that the review is invalid the system will display an error message detailing what was wrong with the review. The user can adjust their review and attempt to submit it again or the user can cancel the review ending the use case

**Special requirements: n/a, Pre conditions: The account user must be logged in, Post conditions: n/a, Extension points: n/a**

**Use case name:** view previous rated books

**Brief description:**

This use case describes how the user can view reviews that they have written

Actors using this use case : Account user

**Flow of events:**

The use case begins when the user views the reviews on their account or views a book that they have previously written a review on

**Basic flow:**

1. The system pull reviews that the user has written from the database
2. The system append the reviews to the webpage, ending the use case

**Alternative flows: n/a, Special requirements: n/a, Pre conditions: The account user must be logged in, Post conditions: n/a, Extension points: n/a**

**Use case name:** view help page

**Brief description:**

This use case describes how the user can view the help page to assist with using the web app

Actors using this use case : Account user, basic user

**Flow of events: The use case begins when the user clicks on the help button**

**Basic flow:**

1) The system will display the help page, ending the use case

**Alternative flows: n/a, Special requirements: n/a, Pre conditions: n/a, Post conditions: n/a, Extension points: n/a**

**Use case name:** update details

**Brief description:**

This use case describes how the user change the details and settings of their account

Actors using this use case : Account user

**Flow of events:**

The use case begins when saves any changes they have made

**Basic flow:**

1.The system will perform checks to prevent invalid data from being entered into the system
2.The system will verify the changes the user made by displaying an “are you sure?” prompt
3.the system will update the database storing the user's data ending the use case

**Alternative flows:**

1.Invalid changes
The user will be prompted to re-enter the invalid data where they can either re-enter the data or cancel out of the operation ending the use case.
2.unverified changes
The user selects no for the prompt allowing them to either re-enter data or cancel out of the operation ending the use case.

**Special requirements: n/a, Pre conditions: Account user must be logged in, Post conditions: n/a, Extension points: n/a**

**Use case name:** activate accessibility options

**Brief description:** This use case describes how the user can enable accessibility options

Actors using this use case : Account user, Basic user

**Flow of events: The use case begins by the user clicking the accessibility button**

**Basic flow:**

1.The system adjust its content to provide better usability ending the use case

**Alternative flows**:

1.Disable accessibility options
The system will revert to its normal format ending the use case

**Special requirements: n/a, Pre conditions: n/a, Post conditions:n/a, Extension points: n/a**

**Use case name:** delete rating

**Brief description:**

This use case describes how the user will delete a review from their account

Actors using this use case : Account user

**Flow of events: The use case begins when the user presses the delete button on any of their reviews**

**Basic flow:**

1.The system will prompt the user with a yes/no form to verify if they want to actually delete their review
2.The system will remove the review from the database ending the use case

**Alternative flows**:

1.The deletion is not verified
After pressing no on the prompt the user will be able to carry on browsing the site as before ending the use case

**Special requirements: n/a, Pre conditions: Account user must be logged in, Post conditions: n/a, Extension points: n/a**

**Use case name:** view suggested book

**Brief description:**

This use case describes how the user can view books recommended by the website

Actors using this use case : Account user

**Flow of events:**

The use case begins when the user visits their homepage

**Basic flow:**

1.The system will pull recommended books from the database and append them to the webpage ending the use case

**Alternative flows**: **n/a, Special requirements: n/a, Pre conditions: Account user must be logged in.,Post conditions: n/a, Extension points: n/a**

**Use case name:** search website

**Brief description:**

This use case describes how the user can use the search function of the website

Actors using this use case : Account user, basic user

**Flow of events:**

The use case begins when the user presses the search button after inputting a search term

**Basic flow:**

1.The system will pull the results of the search from the database
2.The system will split the results into separate pages and display them to the user ending the use case

**Alternative flows**:

1.There are no results
The system will display an error page to the user where the user can search again or use the rest of the website ending the use case
2.The user filters the search
The user can filter the search results given back by the system by: category, search term, author publisher and/ or ISBN, the system will display the filtered results ending the use case

**Special requirements:.n/a, Pre conditions: n/a, Post conditions: n/a, Extension points: n/a**

**Use case name:** create account

**Brief description:**

This use case describes how a user can create a new account

Actors using this use case : Account user, basic user

**Flow of events: The use case begins when the user presses the create account button**

**Basic flow:**

1.The system will validate the details that the user has submitted
2.The system will create the account in the database and send the user to their homepage ending the use case

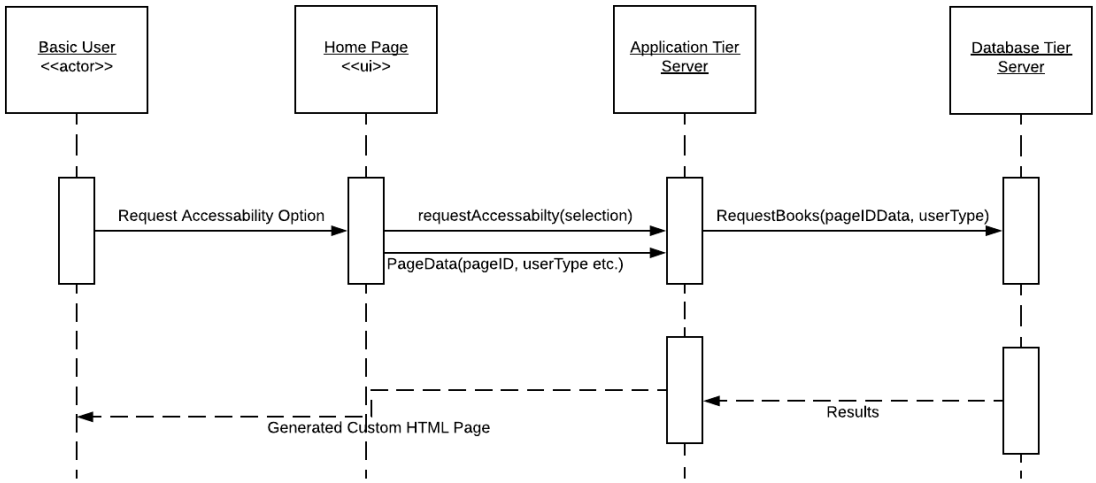
**Alternative flows**:

1.Invalid data
The user will receive an error prompt and will be able to either change the data or cancel out of the process ending the use case

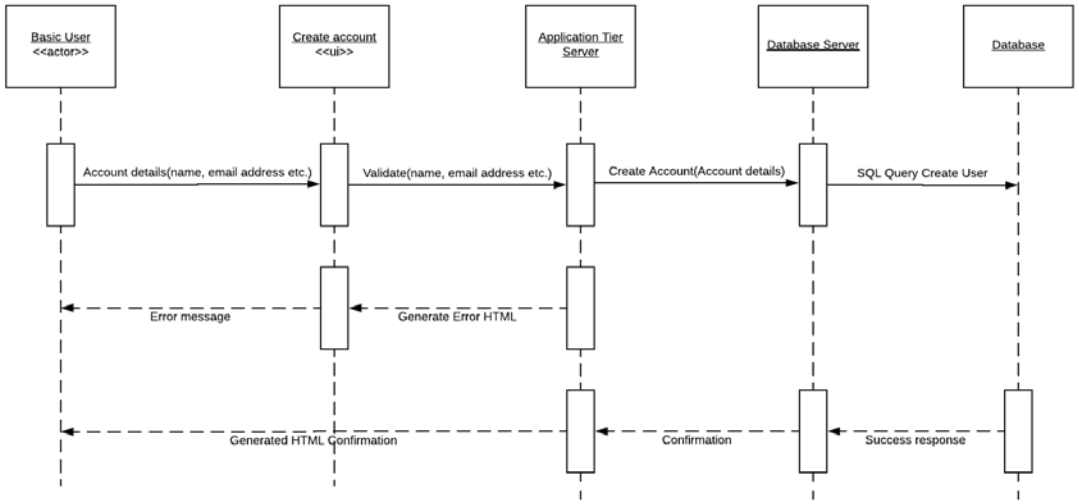
**Special requirements: n/a, Pre conditions: n/a, Post conditions: n/a, Extension points: n/a**

Sequence diagrams:

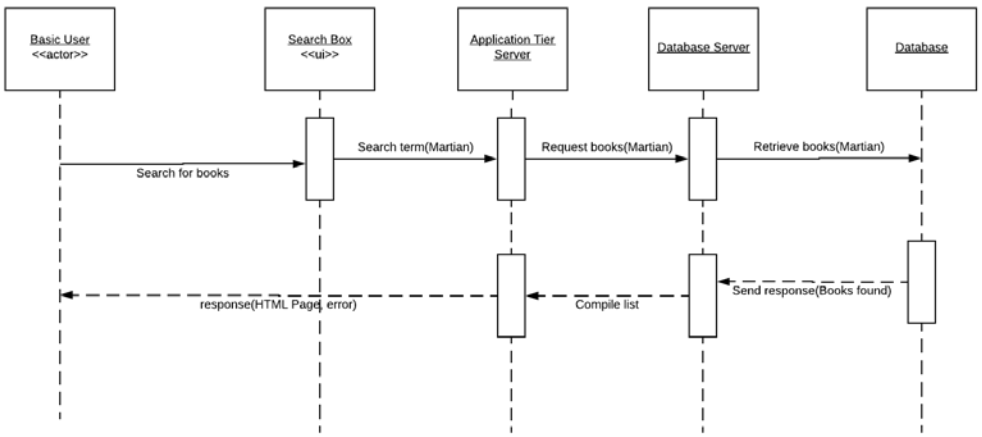
Accessibility options.



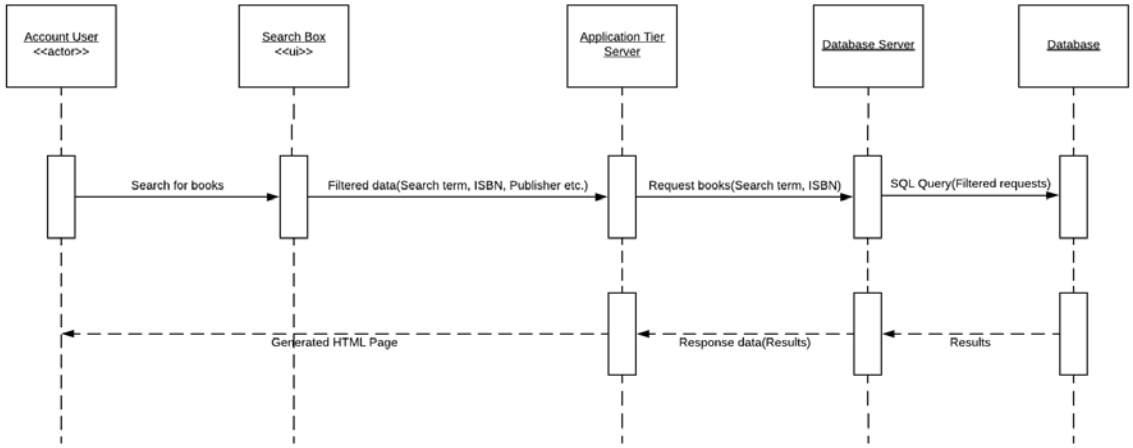

Create an account.



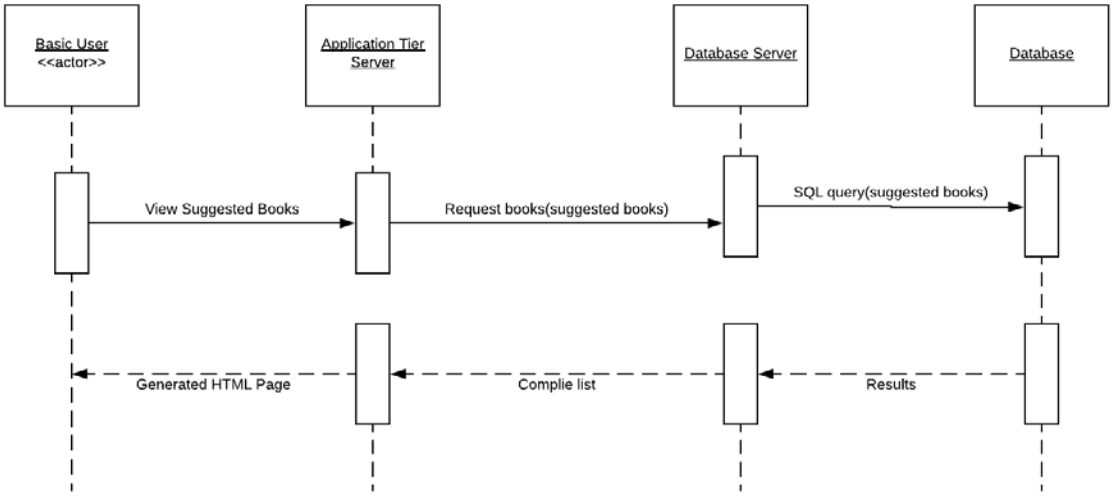

Basic user search



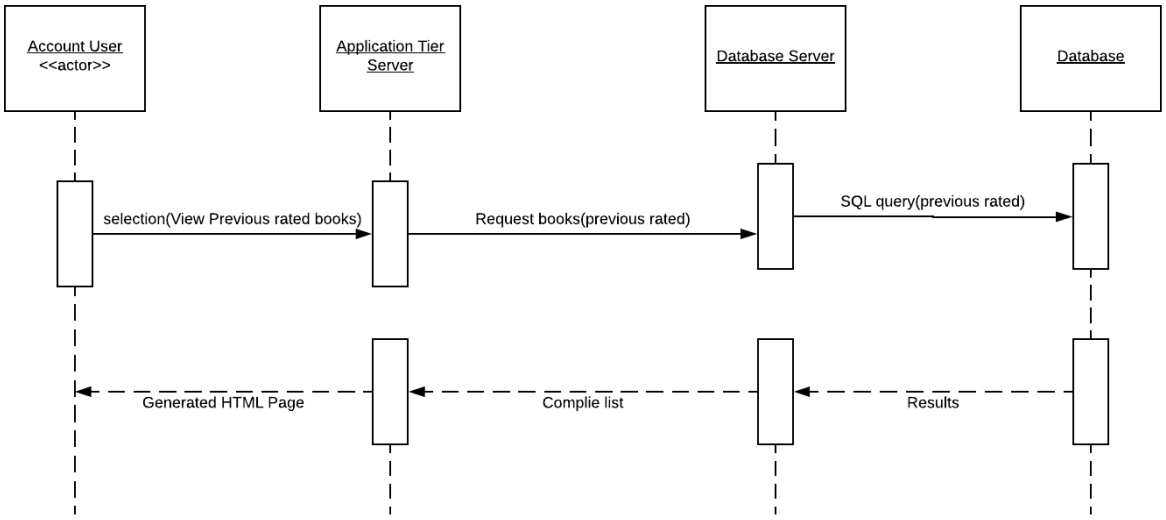
Account user – Search for books



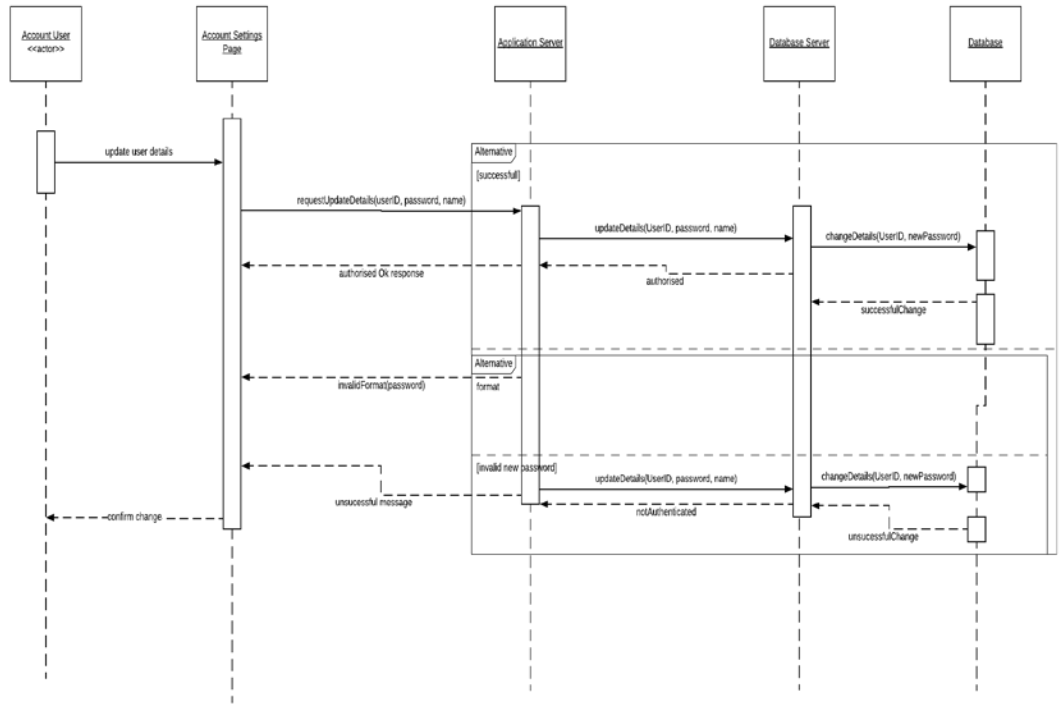

View suggested books.



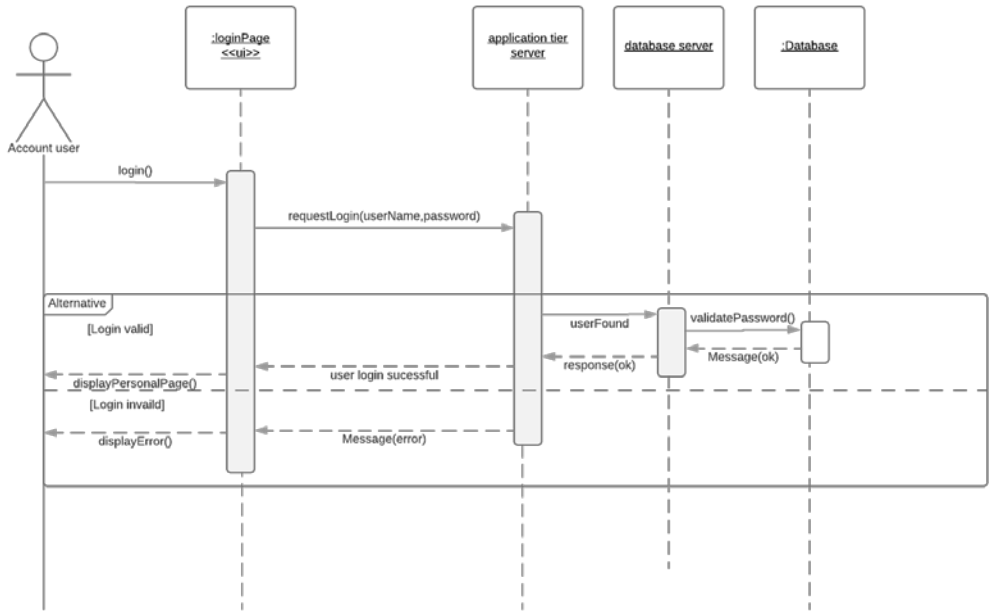

View previous rated.



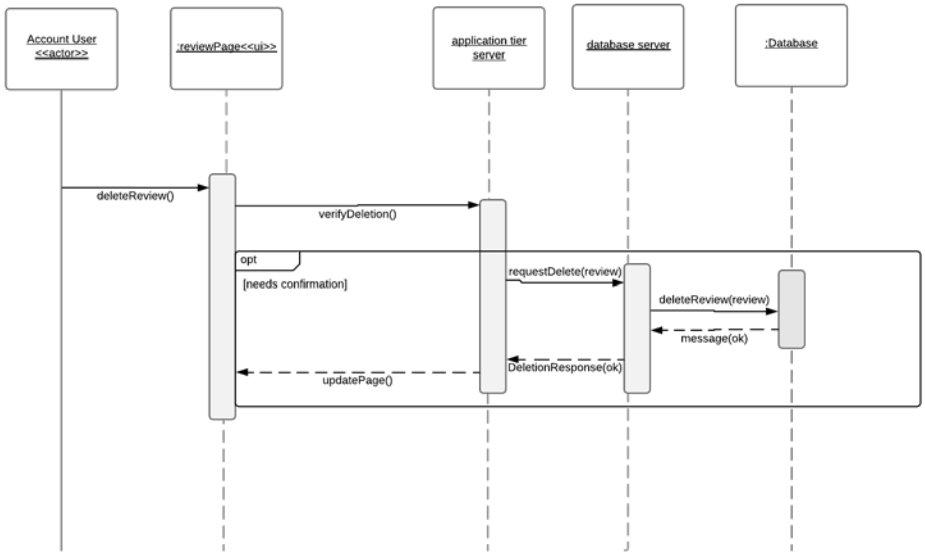
Update details.



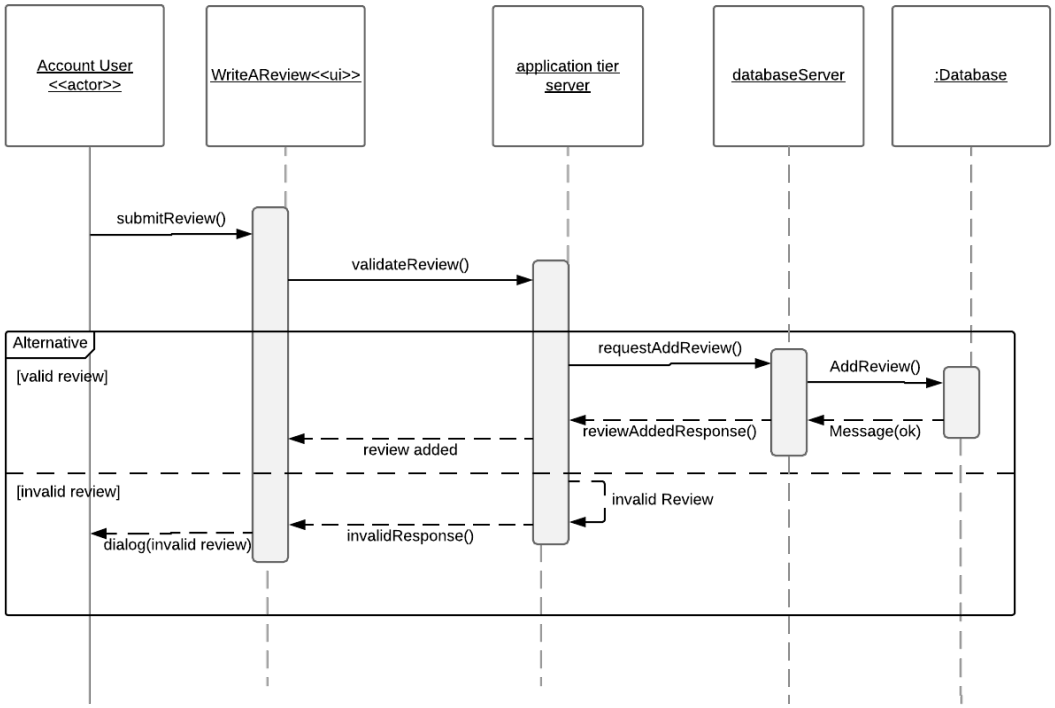

Login.



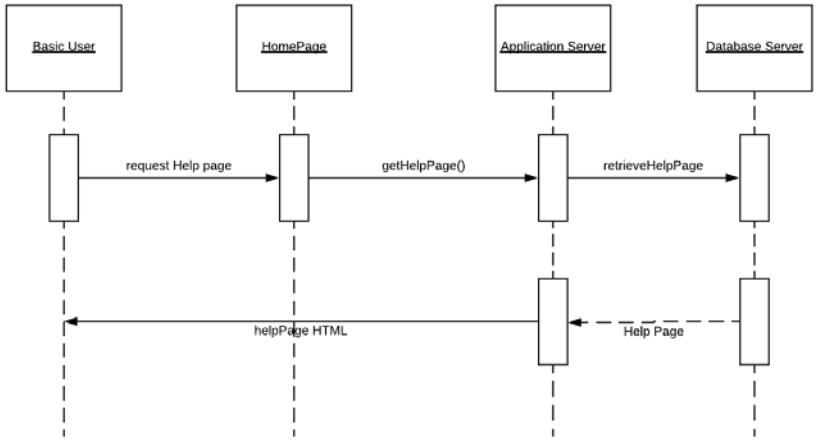
Delete a review.




Write a review.

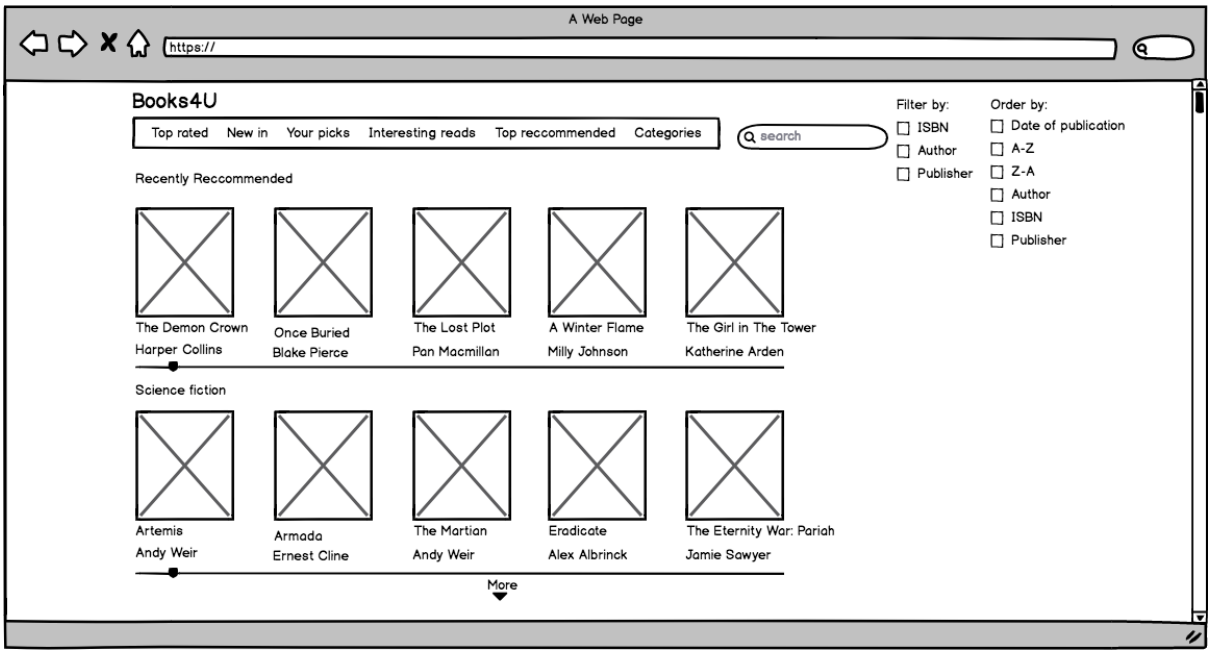


Help form.

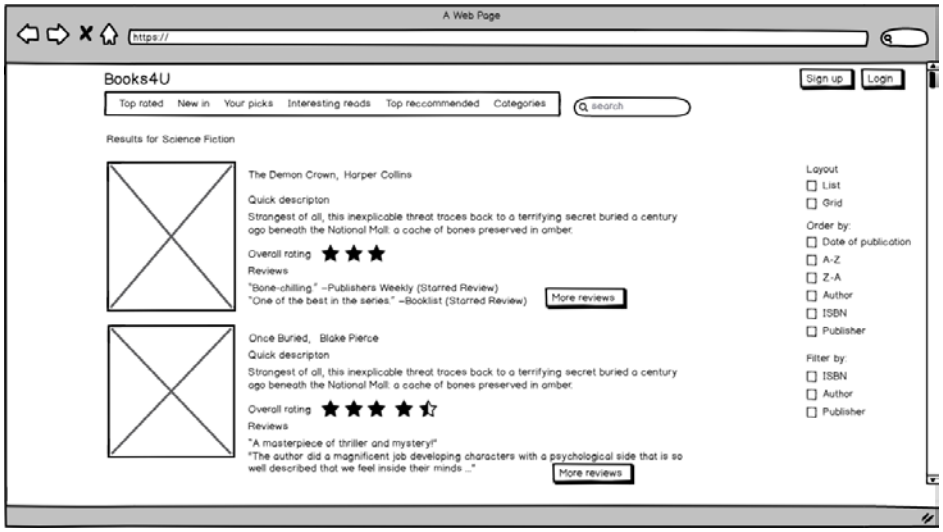



UI Mock-ups.

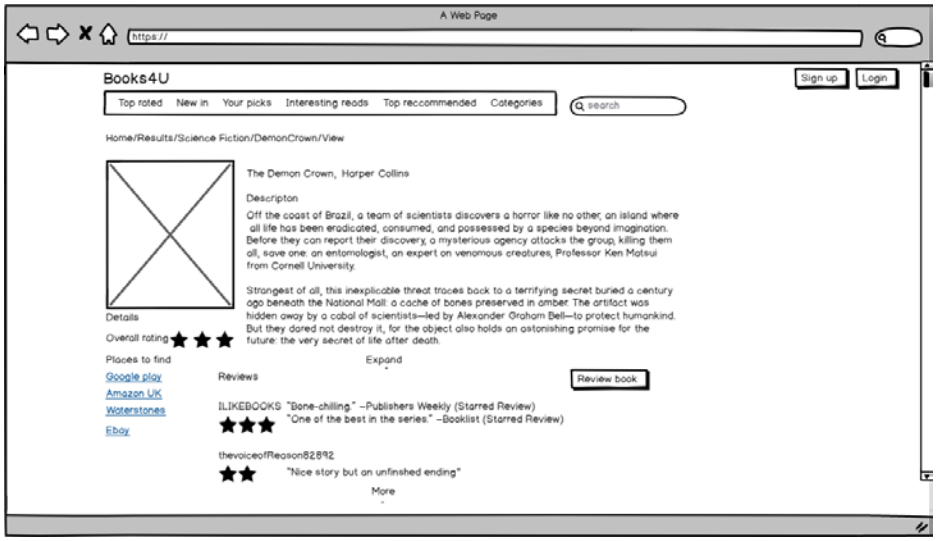
Home page view



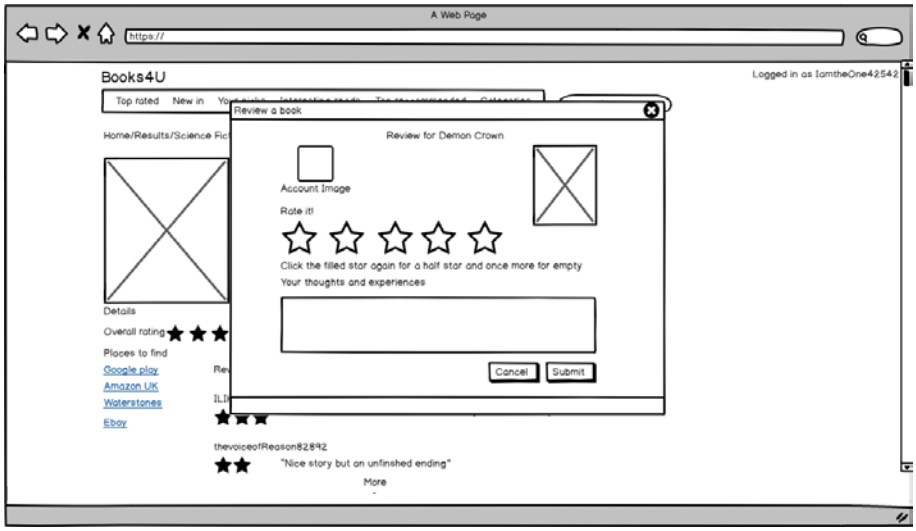

Search Results

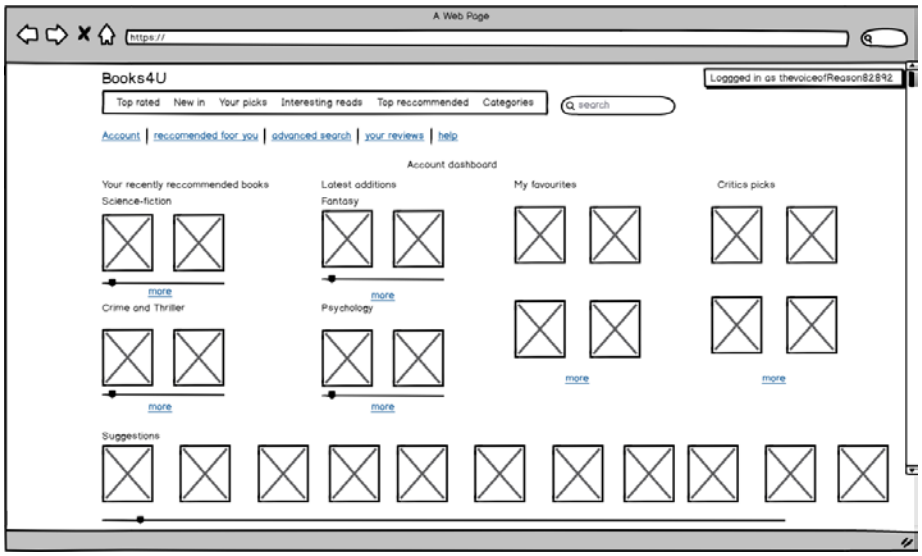


Result view.

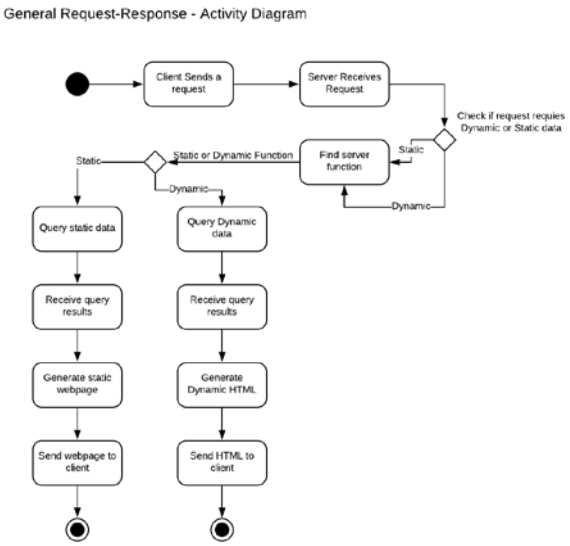


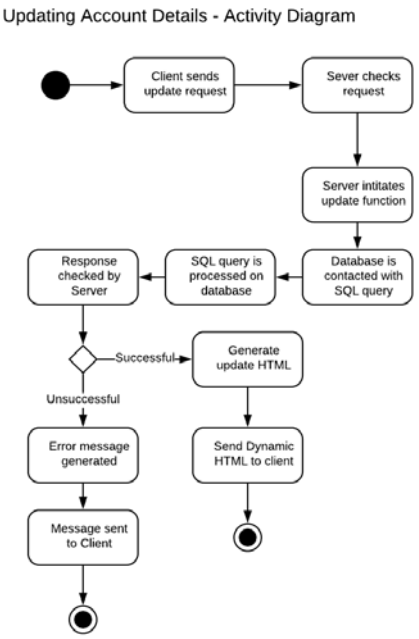

Review a book.



Activity diagrams




State diagrams.