BS2202

Module Leader: Claire Ancient.

Object Oriented Software Development

A 22/23(S2)

Word Count: 2050

Student ID: 2011182

Project Initiation

Project Brief

The project brief [Appendix A] is the comprehensive initiation document that reinforces the areas of interest and the deliverable statements required as part of the agreement of intent of the project between the client and the developer. These areas of interest have been defined within a Project Scope Document, which together creates a set of clear objectives and agreements for the project's direction and scope.

Project Scope

Introduction

LoanShark is a software project, to be created for a library to help the employees with loan management, by cataloguing the items supplied by the library and tracking or creating active loans.

Project Scope

The library software will be written in Java, using an SQL database to store information about the items, employees and customers of the library. Before the commencement of production, rigorous database, logic and UI planning will be designed and documented in line with the client's Specifications.

Project Deliverables

The Primary Deliverable will be a functional and reliable software system that matches the needs of the client, as well as other features and considerations in line with the project requirements. A secondary Deliverable is a comprehensive documentation of the design, production and testing of the project’s primary deliverable, to be completed alongside it by the 15th of May.

Project Acceptance Criteria

End-to-end testing to ensure intended results are found will ensure that the precision of the project is ensured, and then acceptance testing against the project requirements to show accurate functionality

Project Exclusions

The project will not become a customer-facing online portal, nor will it store the personal information of any customer or employee.

Project Constraints

The given design of the project may be subject to change as the solution evolves through production, but these changes will be noted.

Assumptions

An assumption made within the project is that each customer would be responsible for remembering their own customer number and that the payment of the loan fees will be handled externally by a separate program.

Client Requirements

The requirements of the client extend beyond the specific deliverables within the brief, such as implied expectations of a well-made project. For example, the prolonged reliability of the system is expected, and resources must be spent to ensure that this requirement is met when the project is completed. This need is an example of a stakeholder requirement. A stakeholder is any group that can affect or be affected by the realisation of an organisation or product's purpose. (Freeman, 2010)

The Client of the project as a whole is the institution of the library where the library management system is being implemented. The library itself is the major stakeholder, but it also has an interest that the needs of all stakeholders are appropriately met and that the product has high satisfaction and usage rates, which will affect different stakeholders in different ways. Therefore project requirements must be targeting these stakeholder requirements. Harer and Cole, 2005.

The Library's primary requirements are for a system to effectively manage and record the processing of items either being loaned out or returned to the library. This is to satisfy the main requirement of the Customer stakeholder, who is wanting a convenient way to receive and return the items. Additionally, they have an interest in any other material the library can provide, as well as the requirement for the process to be clear, understandable and efficient. The system's user, the Librarian, is a clear stakeholder in the system. The final stakeholder is the Developer, whose focus of the project is to create a reliable, malleable and comprehensible codebase that can be used to deliver objectives and be updated in the future if needed.

User Interface Diagrams

A user interface diagram lays out the look, operations and interactions of the project so that the client is able to approve or modify the direction of the project, and then it is useful as a visual wireframing tool for the developers to follow. A connected sequence of the Interface serves as an effective roadmap of the project functionality.

There are 10 Pages in the program. Each fulfils a stakeholder requirement, and specific care was given to ensure that the application could be moved through both forwards and backwards smoothly so that long sessions would not become disrupted by a linear, continually terminating workflow.

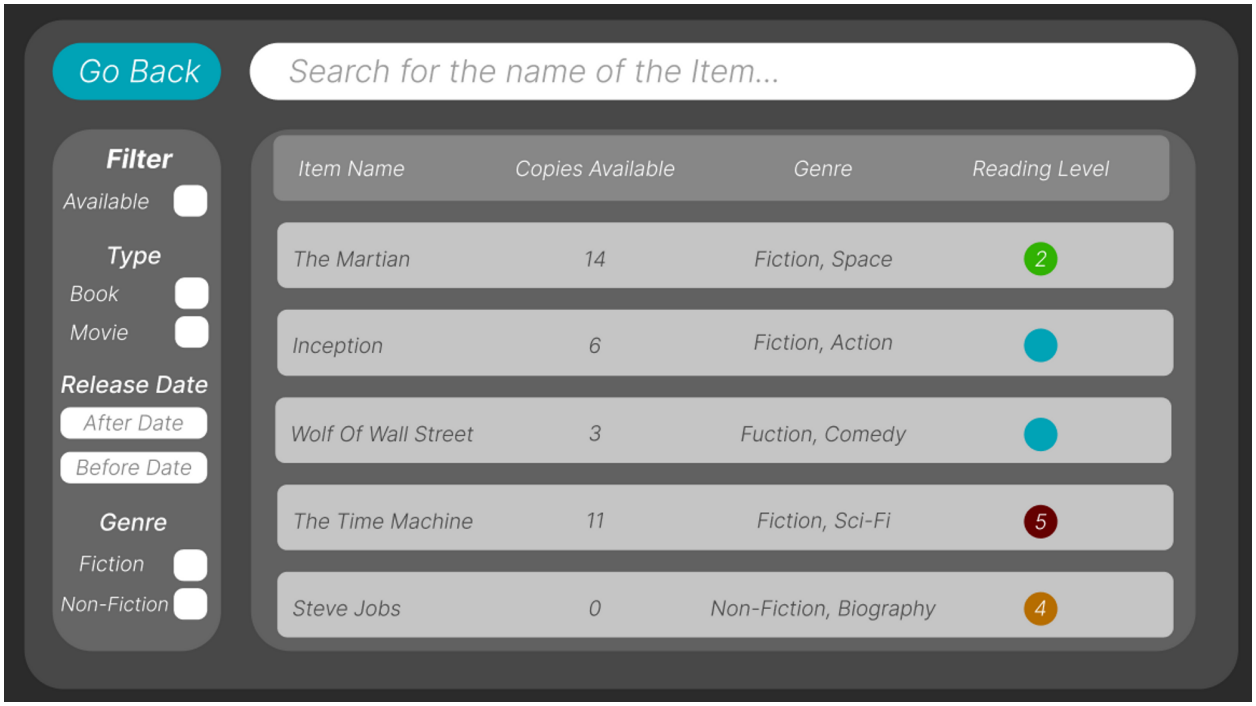
In the item page design, SDG10, (Panday 2020) is the goal to reduce inequality, the advantages of different methods to convey information were considered very carefully, with the implementation of the reading level, which allows the selection of books appropriate for the reader, as well as a colour indication to intuitively show any user who is still unsure. These resources are important to onboard people into a learning process, and the reading of books is shown to improve comprehension (Nation, 1998)

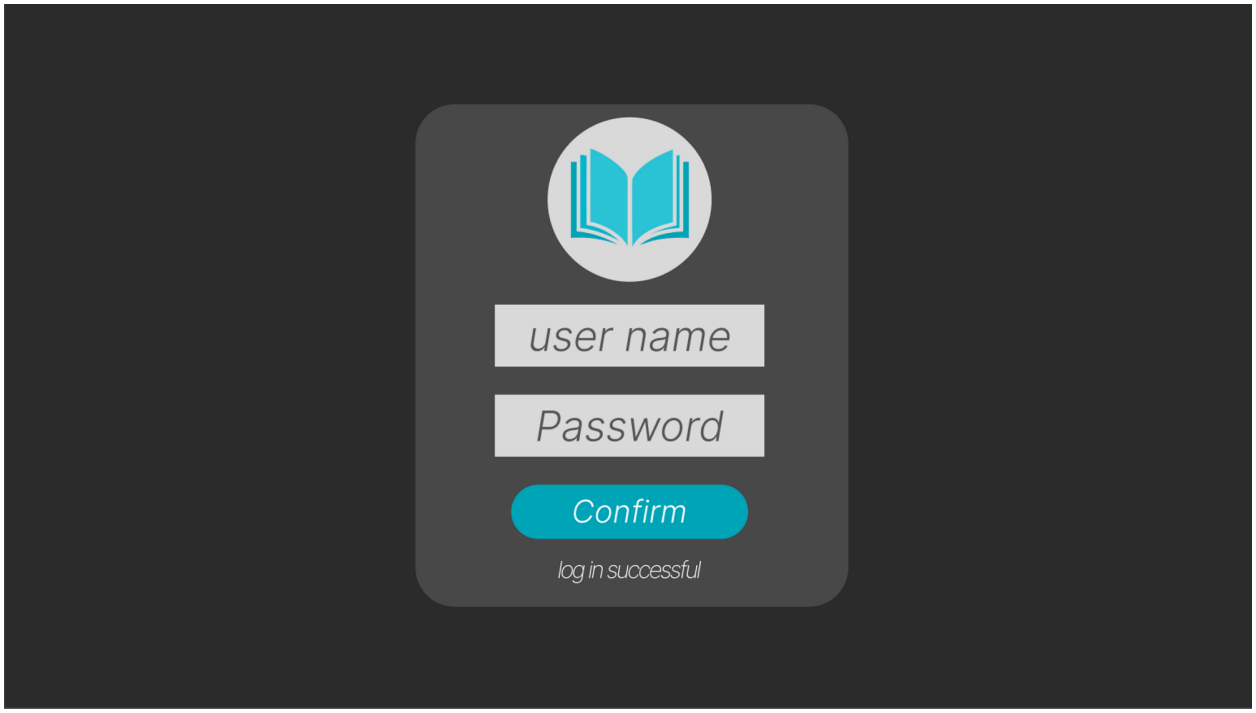
Below are the Complete wireframes interface diagrams designed to be user-friendly, as well as informative. Instead of using a large amount of space needlessly, the information is always centred and in an appropriately sized constrainer. In Figure 1, visual indicators show the level of difficulty for the books, and each page has responsive feedback to ensure that the user is never unsure of whether an action has been completed or not, and if anything needs rectifying.

There have been changes in production, for example, some results pages were amalgamated into the pages that they would have been activated from. The filter in Figure 1 was removed as well for simplicity, as a toggle being unintentionally switched would potentially lead to results not showing up. Buttons are always the same colour, and they point out the interactable elements of the application. Likewise, light grey is often used for text Fields or to show pieces of information, and careful consideration was given so that there would not be clashing of colours, which would damage usability.

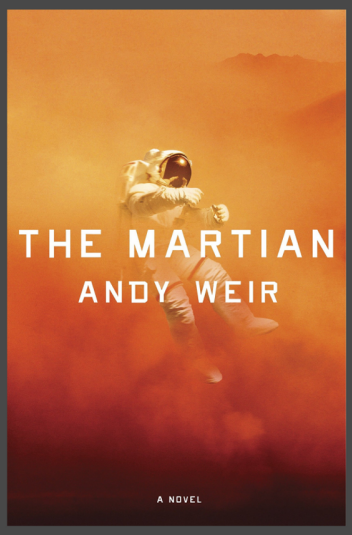
The user always works from the top of the application down, including messages regarding errors or conformations. if possible, the size of the elements also decreases as the user goes down the point, and so although the application has versatility and can be traversed forwards, backwards and with some jumps in workflow, such as the recommendation tab which can allow the user to both return and loan books, the nature of the funnelling effect creates a workflow that means you are not interrupted by having to travel all over the page.

The button is always in the same place, deliberately away from the window options in the top right, so the user does not have to worry about accidentally closing the program.


Go Back
Search for the name of the Item...
Filter
Available
Type
Book
Movie
Release Date
After Date
Before Date
Genre
Fiction
Non-Fiction
Item Name
Copies Available
Genre
Reading Level
The Martian
14
Fiction, Space
2
Inception
6
Fiction, Action
Wolf Of Wall Street
3
Fuction, Comedy
The Time Machine
11
Fiction, Sci-Fi
5
Steve Jobs
0
Non-Fiction, Biography
4


user name
Password
Confirm
log in successful

Go Back



Item Type: Book
Book Name: The Martian
Author: Andy Weir
Item Reference: IT86F309

Available Versions:
- Paperback
- Hardcover
- Audiobook

Tags:
- Fiction
- Space
- Adventure

Next Earliest Return: 23-04-2023

Next Selected Version Return: 23-04-2023

Any Version Available: Yes (14)
Selected Version Available: Yes (3)

Select A Book Version

Select A Loan Length

Enter Customer ID

Confirm Loan

Go Back

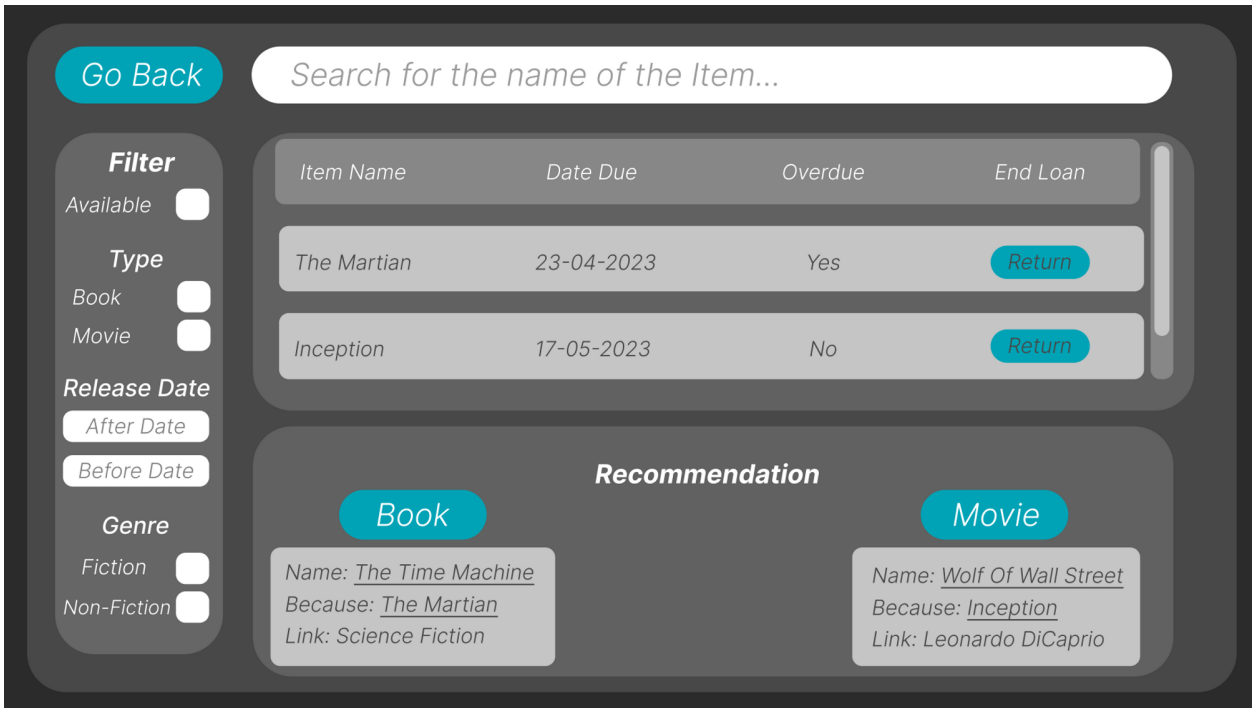
Type in a Customer ID to go to that page

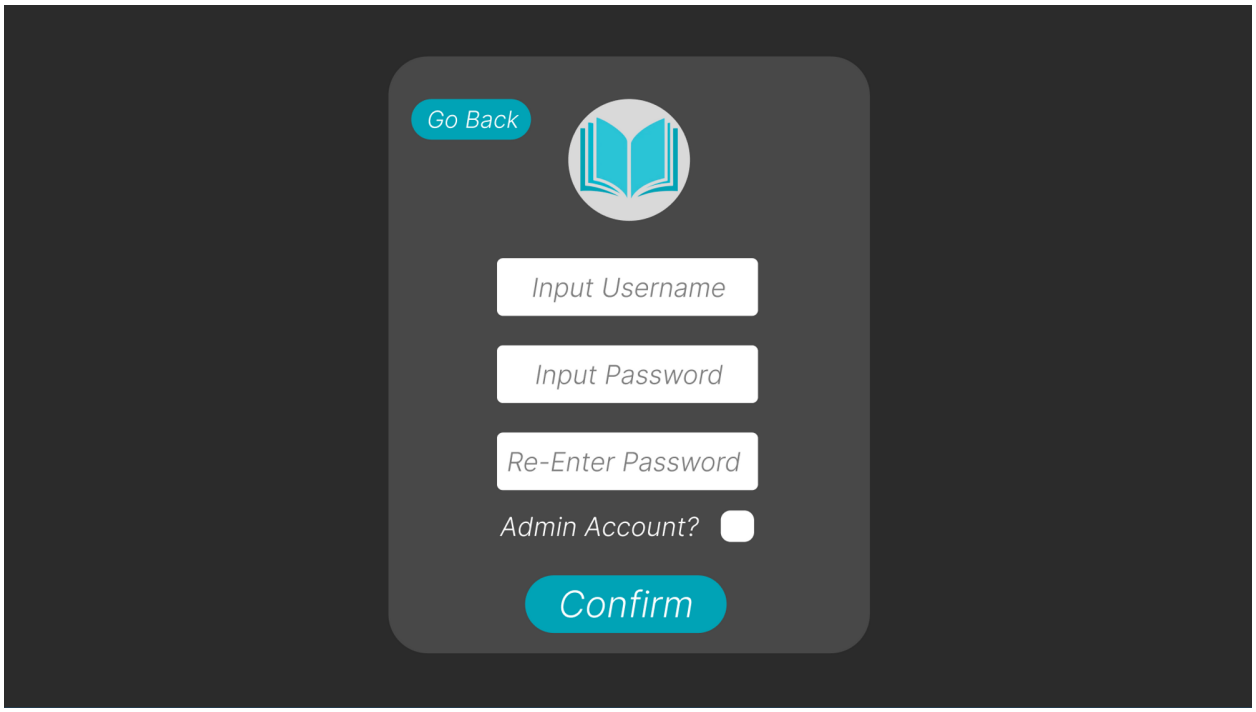
Enter Customer Number

Press button to generate a new Customer ID

Generate ID

New ID: CU203948


Go Back
Search for the name of the Item...
Filter
Available
Type
Book
Movie
Release Date
After Date
Before Date
Genre
Fiction
Non-Fiction
Item Name
Date Due
Overdue
End Loan
The Martian
23-04-2023
Yes
Return
Inception
17-05-2023
No
Return
Recommendation
Book
Name: The Time Machine
Because: The Martian
Link: Science Fiction
Movie
Name: Wolf Of Wall Street
Because: Inception
Link: Leonardo DiCaprio


Go Back
Input Username
Input Password
Re-Enter Password
Admin Account?
Confirm

Go Back

Results Page

Item Type: Book
Book Name: The Martian
Version: Paperback
Author: Andy Weir

Event Reference: EV2B3941
Customer Reference: CU10PS31
Item Reference: IT86F309

Date Loaned: 23-03-2023
Time Of Loan: 1 Month
Due Back: 23-04-2023

Save in File

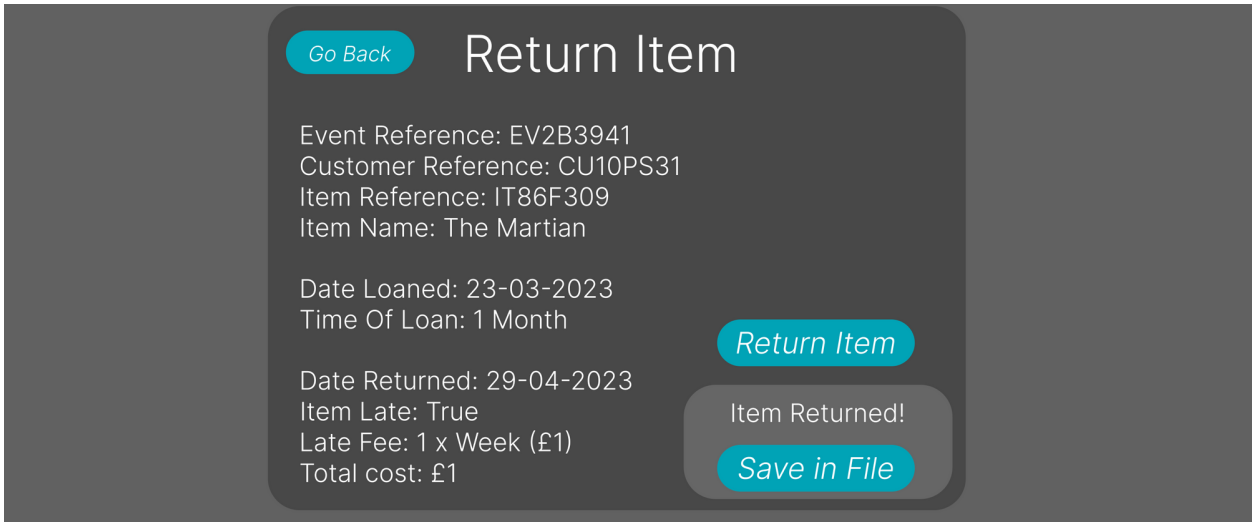
Go Back

Type In Event ID to
Return Item

Enter Event ID

Return Item

Event ID not found

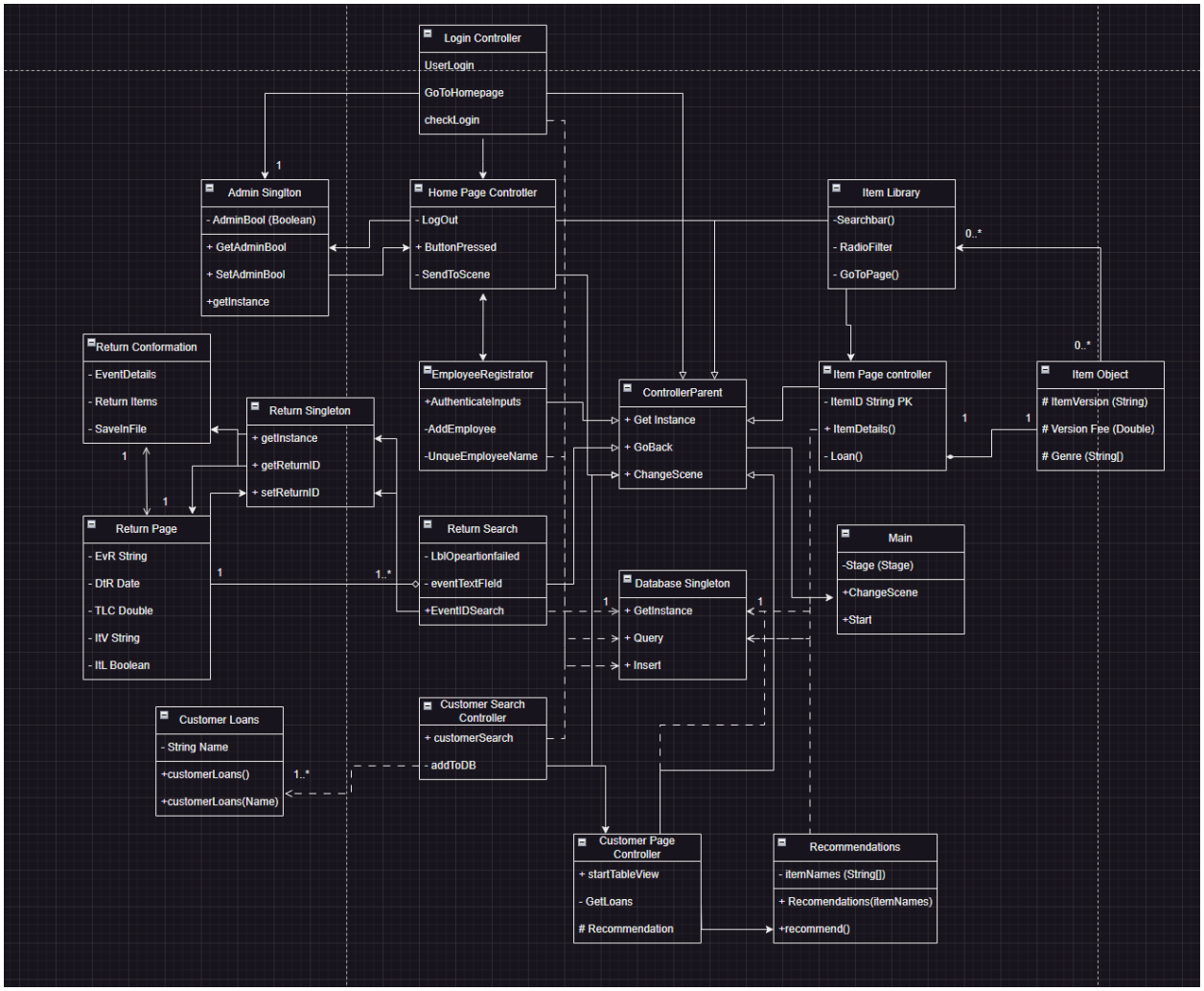


Class And Entity Diagrams

While the user interface diagrams provide insight and guidance for creating the Front end functionality and interactions of the application, Class diagrams are used to show the technical structure of the project, as well as list the resources that will be used to achieve the stated goals of the program.

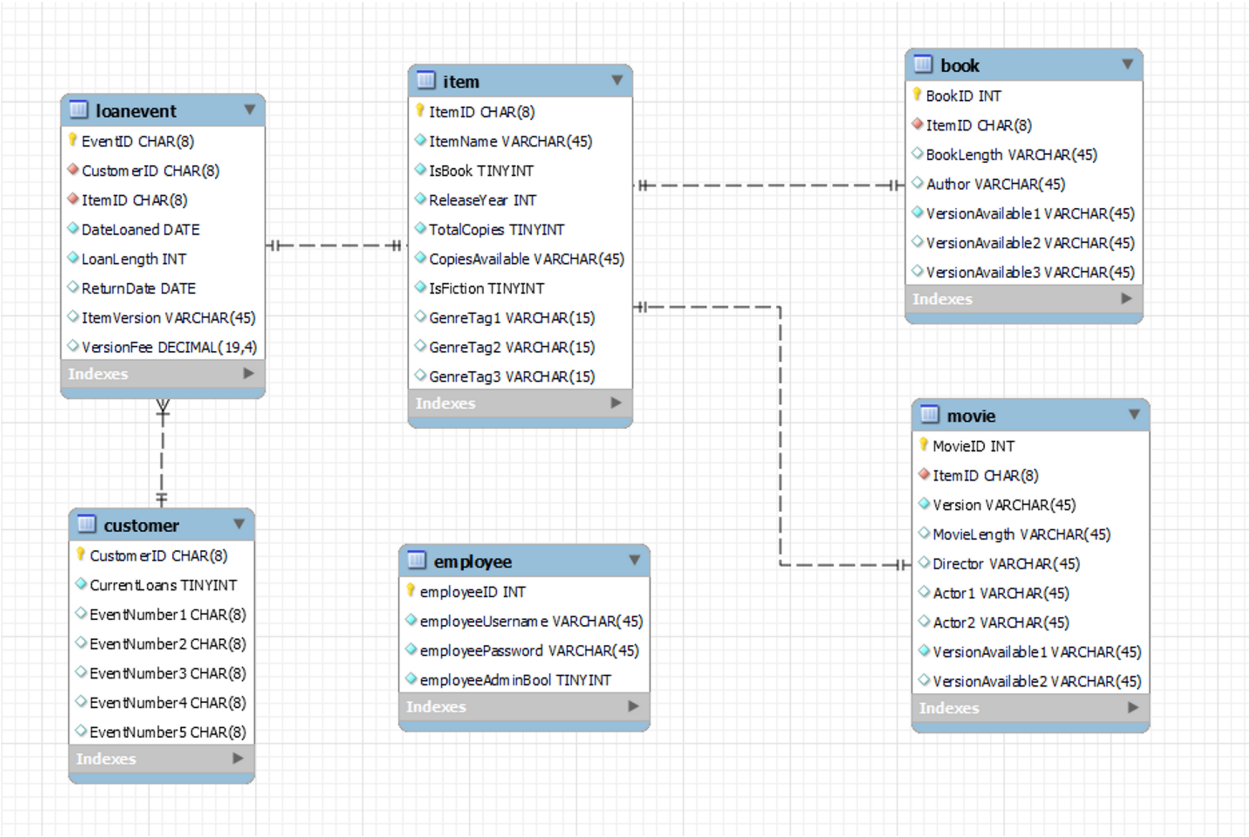
An example of another diagram that is used to show the structure of the information relating to the project is an Entity Relationship Diagram. This diagram is used to detail all the database tables used within the project, as well as the relationship of each table to each other. For Example, the use of one-to-one relationships can provide information to further understand the nature of the application's data structure.

There is a third UML diagram [See appendix B] to show a charting relationship of the front-end structure and map out the relationship of all pages with each other. As the class diagram displays the technical description of the components, classes and objects used to construct the program, and the ERD displays the structure of information underpinning the application, the third UML diagram is an interaction overview diagram, to map out the possible interactions the client can have with the application.



A design pattern (Cooper, 2000) is a structured way to reuse resources, therefore increasing productivity and driving the product forward. There are 3 main types of design patterns, but the LoanShark program follows A creational design pattern. This means that the program makes use of flexible design structures that protect against inefficiency. An example of this used in my project is the Singleton, which is used to move information around the program, but also to channel all database traffic through exactly one instance, therefore duplication or inefficiency is greatly reduced, as the connection and traffic will all flow through one point and it is not possible to have multiple iterations of this singleton running at the same time.

There are elements of other design patterns, such as structural inheritance, which are used to increase the efficiency of the development, allowing a developer to reuse the code for as many uses as applicable. For the ControllerParent class, which allows all pages to have a “GoBack” button all controlled from within the parent class.



This Entity relationship diagram details the connections between the tables within the databases. These connections come in the form of foreign keys, which contain common pieces of information, which are likely to be the primary key in one table, which can be found in other tables and can be used to link the tables together.

For example, within the program, a customer does not have records of the items that they are currently loaning, but they have the receipts for the loan events themselves, which in turn hold a foreign key for the Item identification. The customer can be linked to the details of any book being loaned by them, but only through the use of a well-organised and interwoven database.

Testing

The testing of a system is fundamental to both the operation and acceptance of the project upon its completion and should be carried out across the entire scope of development, often post-development and to accomplish a series of goals. Unit testing is carried out to ensure that individual components fulfil the role and requirements that they are intended and designed to carry out. Integration testing is used when multiple components are being run together, and this ensures that each is harmonious in terms of functionality. Ammann, 2016
From an acceptance perspective, end-to-end testing is carried out to ensure that stated deliverables have been achieved, as well as comprehensive performance testing to ensure that the metrics by which functionality is achieved are also at acceptable levels. A well-regimented testing plan will include all of these tools to deliver a comprehensive and complete understanding and tested system which can be accepted by the client and used with high reliability.

Test Case A: HashMap Testing

This test case is an example of unit testing. The unit being tested is the Homepage directing Hashmap, which is responsible for connecting all the branches of the program and allowing the user to access any of them. This case will include exploratory testing to ensure coverage of all likely inputs.

```
private void sendToScene(Object btnName) throws IOException {
    HashMap<String,String> SceneDestination = new HashMap<>();

                        //Which button , FXML file for that page
    SceneDestination.put("btnCustomers","customerSearch.fxml");
    SceneDestination.put("btnItemSearch","itemLibrary.fxml");
    SceneDestination.put("btnEmployeeReg","employeeRegister.fxml");
    SceneDestination.put("btnReturns","returnSearch.fxml");

    callChangeScene(SceneDestination.get(btnName));
}
```

TestCase	Description	Expected result	Result	Pass/Fail
1.01	Testing the Customers branch	CustomerSearch.fxml is loaded	CustomerSearch.fxml is loaded	Pass
1.02	Testing the Item	itemLibrary.fmxl is loaded	itemLibrary.fmxl is loaded	Pass

	Search Button			
1.03	Testing the Return branch	returnSearch.fxml is loaded	returnSearch.fxml is loaded	Pass
1.04	Testing employee registration	employeeRegister.fxml is loaded	employeeRegister.fxml is loaded	Pass

Despite the fact the unit testing was successful and there were no failures, exhaustive testing is impossible and not all inputs have been tested, so the unit can never be declared 100% reliable.

Test Case B: Generalised SQL Query

This test case is an example of integration testing. The 2 components being tested are a Controller for the application and the SQL database. The testing involves querying for results only belonging to a particular custom ID. This testing will involve edge case testing, to ensure that there are no exceptions that can be triggered from inputting irregular or incompatible data.

TestCase	Description	Expected result	Result	Pass/Fail	Changes
1.01	Querying for an Integer	ResultSet queryResult	ResultSet queryResult	Pass	
1.02	Querying for a String	ResultSet queryResul	Fatal Error	Fail	Introduction of “”/” quotation marks around String queries
1.02	Querying for a String	ResultSet queryResul	ResultSet queryResul	Pass	
1.03	Querying for a Date	ResultSet queryResul	ResultSet queryResul	Pass	
1.04	Querying for a Boolean type	ResultSet queryResul	ResultSet queryResul	Pass	

An error was found in test case 1.02, where when the testing had a String type as input, as it has to be in quotation marks for SQL, these have to be added to integrate the components properly

Test Case C: Routing Testing

This test case is an example of end-to-end testing. This is testing the process for searching for a specific Loan Event, in the scenario of an item being returned. As much as the technical solution is imported, end-to-end testing is primarily focussed on a set input, and the expected outcome, which if met would satisfy a client requirement of the project.

TestCase	Description	Expected result	Result	Pass/Fail
1.01	No Input	“No ID Entered”	“No ID Entered”	Pass
1.02	CU	“Customer ID not found”	“Customer ID not found”	Pass
1.03	%3!)£(*$!00000 0000000A	“Customer ID not found”	“Customer ID not found”	Pass
1.04	CU942302	“Customer ID not found”	“Customer ID not found”	Pass
1.05	CU023830	Taken to customer page for CU023830	Taken to customer page for CU023830	Pass

These late-stage testing scenarios are often conducted with, or by the clients, also known as acceptance testing. With the client's requirements met, the project can be declared as ready for production, as the stated goals and criteria of the project have been met. Miller,2001

Appendix

Appendix A: Project Proposal

Introduction:

Library System:	LoanShark
Library System Description:	LoanShark is used by employees to complete a number of tasks necessary to running a library system. These actions include the loaning and returning of items, the cataloguing of available items that can be loaned, tracking active loan events and management of customer accounts. Employees are required to log in, and new employees can be added by designated administrator accounts. There is a database which contains multiple tables which contain relevant information such as storing employee login information, as well as a list of Loan Events, showing all currently outstanding loans, which item they are on and which customer is currently loaning the item. There is a graphical user interface to deliver the information to the user in a convenient and accessible way. The navigation of the application is helped by a consistent and helpful platform of page interconnectedness. In order to increase user satisfaction, recommendations based on the library activity of a customer would be able to provide new catered reading experiences to all customers. In accordance with SDG10, visual aids will help reduce academic inequality, and so both images and visual guides are implemented to assist in non-written ways for the library program. Users are able to receive receipts of the transactions, particularly the loaning and returning of items by having the information saved to a text file.

Key System Features:

- Database
  - Ability to see a customer's outstanding loans, and whether they are late or not
  - Ability to see an item and check the availability of it at that time.
  - Ability to look up any outstanding loan event tracker and be able to terminate the loan if an employee sees fit
- Graphical User Interface
  - Seamless integration of database functions into the front end
  - Ability to intuitively select options for loaning items using a GUI.
  - Ability to see photos of books or movies that could interest the customer
  - Ability to see a wide range of options for loanable items in a collection search screen

    - Ability to learn important details about each item and its author, director or length
- Login System
    - Security for employees only
    - Ability to register employees into the system
    - Limiting this registration ability to only administrators
    - Removal of elements for those not able to interact with them for clean design
    - Responsive database checks to not cause a delay in system startup.
- Item Images
    - Sourced from application SRC, corresponding to the correct property
    - Uniform size so each item frame is the same size and therefore continuity is preserved
- Search Functionality
    - Ability to enter the name of an item and the result returned to you, either as a ID for a customer or loan event or the name of an item that the user would want to loan out.
    - Quick and efficient searching to deliver satisfaction without significant delay
- File Output
    - Comprehensive report on item loaning or returning, output in a file for convenience
- Other Appropriate Functionality
    - Recommending similar items to a user to increase library engagement.
    - Large range of options, so a limited amount of info can still be used productively.

Core Item List:

Item Description	**Late Fees (Per Day)**
Book (Paperback)	£0.50
Book (Hardback)	£1.00
Book (Audiobook)	£2.00
Movie (Standard)	£1.00
Movie (Directors Cut)	£1.50

Functional Requirements:

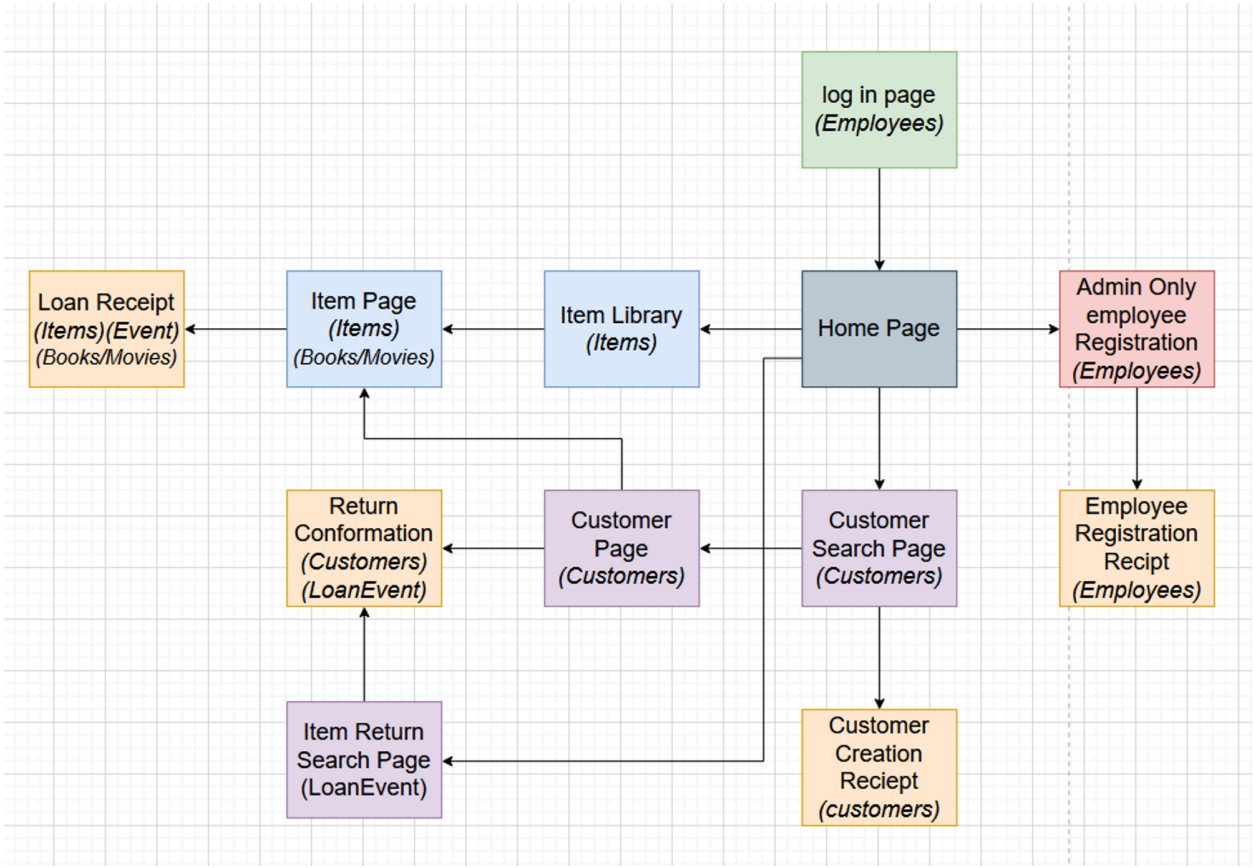
Req ID	**Requirement Description**	**Priority**
1	Ability To Loan an item	1 (Critical)
2	Ability to return an item	1 (Critical)

3	Constant accurate recording of library resources	2 (Very High)
4	Ability to search or filter for targeted interests	3 (Very High)
5	Ease of use of the system for library worker and customer interaction, for all people	4 (High)
6	Management of Employee Login and security	4 (High)
7	Receipts for personal proof of loan/return	5 (Medium)
8	Reasonable Efficiency of speed and computing resource	6 (Medium)
9	Recommendations for similar items based on user interests	7 (Low)

Non-Functional Requirements

Req ID	**Requirement Description**
10	Data security for usernames and passwords. Non-functional due to no GDPR qualifying information being stored or even entered into the program. However, hashing and protecting any passwords would still be ideal if possible.
11	Further Exception handling to make it more intuitive if incorrect inputs or non-feasible operations. One such way would be small separate stage windows for error messages
12	Scalable components to make the UI more accessible and suited for a range of non-standard full-screen window sizes
13	Expansion of the number of items that can be loaned by one customer at any given time
14	Wider scope and depth of recommendation algorithm, to take into account all previous loans, not currently active loan. As well as this, to establish and implement other metrics to compare books..

Appendix B: UML Diagram



References

Citation in text	Citation in full
Ammann, 2016	Ammann, P. and Offutt, J., 2016. *Introduction to software testing*. Cambridge University Press.
Cooper, 2000	Cooper, J.W., 2000. Java design patterns: a tutorial.
Freeman, 2010	Freeman, R.E., Harrison, J.S., Wicks, A.C., Parmar, B.L. and De Colle, S., 2010. Stakeholder theory: The state of the art.
Harer and Cole, 2005	Harer, J.B. and Cole, B.R., 2005. The importance of the stakeholder in performance measurement: critical processes and performance measures for assessing and improving academic library services and programs. *College & Research Libraries*, *66*(2), pp.149-170.
Miller, 2001	Miller, R. and Collins, C.T., 2001. Acceptance testing. *Proc. XPUniverse*, *238*.
Nation, 1998	Nation, P. and Coady, J., 1988. Vocabulary and reading. *Vocabulary and language teaching*, *97*, p.110.
Pandey, 2020	Pandey, U.C., Kumar, C., Ayanore, M. and Shalaby, H.R., 2020. *SDG10–Reduce inequality within and among countries*. Emerald Group Publishing.