

SOFTWARE REQUIREMENT ANALYSIS DOCUMENT



CSE3112: Software Engineering Lab

Title: Boikini

Submitted By:
Shahanaz Sharmin (07)
Anika Tabassum (61)



Contents

1	Introduction	4
1.1	Purpose of the system	4
1.2	Scope of the system	4
1.3	Objective and success criteria of the project	4
1.4	Definitions and Acronyms and Abbreviations	5
1.5	References	6
1.6	Overview	6
1.6.1	Introduction	6
1.6.2	Overall Description	6
1.6.3	Specific Requirement	6
1.6.4	Supporting Information	6
2	Overall Description	6
2.1	Product perspective	6
2.2	Product functions	7
2.3	User Profiles	8
2.4	Constraints	8
2.5	Assumptions	9
2.6	Dependencies	9
3	Specific Requirement	9
3.1	Overview	9
3.2	Functional Requirement	10
3.3	Non Functional requirement	13
3.3.1	Usability	13
3.3.2	Reliability	13
3.3.3	Performance	14
3.3.4	Supportability	15
3.3.5	Implementation	15
3.3.6	Scalability	16
3.3.7	Security	17
3.3.8	Maintainability	17
3.3.9	Testability	18
3.4	System Models	19
3.4.1	Scenarios	19
3.4.1.1	User registration	19
3.4.1.2	Searching for books	19
3.4.1.3	Placing an order	19



3.4.1.4	Tracking an order	19
3.4.1.5	Writing a review	19
3.4.1.6	Managing a user account	19
3.4.1.7	Seller login	19
3.4.1.8	Adding a book to inventory	19
3.4.1.9	Setting prices and discounts	19
3.4.1.10	Managing orders	20
3.4.2	Use cases	20
3.4.3	Use case model	23
3.4.4	Dynamic Model	24
3.4.4.1	Sequence Diagram	24
3.4.4.2	Statechart Diagram	25
3.4.4.3	Activity Diagram	26
3.4.5	Interfaces	26
3.4.5.1	User Interface	26
3.4.5.2	Software Interface	27
3.4.5.3	Hardware Interface	27
4	Supporting Information	27



1 Introduction

1.1 Purpose of the system

The purpose of the Boikini system is to provide a user-friendly and convenient platform for buying and selling books online from anywhere in Bangladesh. The system will enable users to browse and search for books, place orders, and have the books delivered to a specified location. It will provide a wide range of books to users without the need to visit physical stores. The system will provide a simple and intuitive user interface, making it easy to use for even the most novice users. It will also provide a secure and reliable platform for conducting transactions, with safeguards in place to prevent fraudulent activities. The corporate goals and business strategies of the Boikini system are to create a profitable and scalable online marketplace for books. This system will help to create a more efficient and accessible marketplace. By creating an easy-to-use system that provides excellent customer service and support, the Boikini system aims to become a market leader in the online book-selling industry. A separate vision and scope document is not available at this time, but the objectives and goals of the project are to create a reliable and user-friendly online platform for buying and selling books. The system aims to provide a high-quality service that exceeds the expectations of its users, and to continuously improve and innovate the system to maintain its position as a market leader.

1.2 Scope of the system

The product covered by this RAD includes the entire Boikini system, including all features and subsystems that enable users to browse and search for books, place orders, and have the books delivered to a specified location. This RAD does not describe only part of the system or a single subsystem, but rather encompasses the entire system as a whole. The Boikini system will have two main user roles: buyers and sellers. Buyers will be able to browse books, search for specific titles or authors, and place orders for delivery to a specified location. Sellers will be able to upload books for sale, manage their inventory, and fulfill orders. The system will allow buyers and sellers to communicate with each other and to rate each other's transactions. The system will also have an administration section, which will allow the system administrators to manage users, transactions, and the overall system configuration. The system will be accessible from any device with an internet connection, such as a desktop computer, laptop, tablet, or mobile phone.

1.3 Objective and success criteria of the project

The general objective of the Boikini project is to provide a reliable , user-friendly and easily accessible online platform for buying and selling books from anywhere in Bangladesh. The specific objectives of the project include:

- Enabling users to search for books and place orders easily.
- Creating a secure and reliable system that protects users' information and transactions from fraud.
- Building a scalable and adaptable system that can handle a growing volume of users and books.



- Providing excellent customer service and support to ensure a positive user experience.

The success of the Boikini project will be measured by its ability to meet its objectives and achieve the following criteria:

- **User satisfaction:**

The system must be easy to use, provide a wide range of books, and ensure secure and reliable transactions.

- **Increased sales:**

The system should help increase sales for book sellers and provide a growing range of books for buyers.

- **Scalability and Adaptability:**

The system should be able to handle a growing volume of users and books, as well as be adaptable to changes in the industry and technology.

- **Timely delivery:**

The system should ensure that orders are delivered to the specified locations in a timely and efficient manner.

- **Efficient Customer Support:**

The system should provide excellent customer support and respond to any issues and inquiries in a timely manner.

The achievement of these success criteria will ensure that the Boikini system is effective, efficient, and successful in providing a reliable and user-friendly platform for buying and selling books online.

1.4 Definitions and Acronyms and Abbreviations

- **Boikini:**

The name of the online platform for buying and selling books.

- **Buyers:**

Users who purchase books on the Boikini platform.

- **Sellers:**

Users who offer books for sale on the Boikini platform.

- **Administrators:**

Users with system-level access to manage the Boikini platform.

- **UI:**

User interface.

- **API:**

Application programming interface.

- **SSL:**

Secure Sockets Layer, a protocol for securing online communications.



1.5 References

- ASTM 1340-90, Standard Guide for Prototyping of Computerized Systems
- IEEE Std. 610.12-1990 IEEE Software glossary for Software Engineering Terminology
- <http://www.utdallas.edu/chung/RE/IEEE830-1993.pdf>
- <https://app.creately.com>

1.6 Overview

1.6.1 Introduction

This section provides a general overview of this Requirement Analysis Document (RAD). It tells us about the purpose and scope of the project undertaken by the developers, and tries to describe very briefly about its functionalities and goals.

1.6.2 Overall Description

This section describes the perspective and functionalities of the project. It provides us with a basic overview of the entire project as well as the constraints, assumptions and dependencies that come with it.

1.6.3 Specific Requirement

In this section, the functional and non-functional requirements of the project have been pointed out. All the necessary models and diagrams have also been depicted. This section is basically a broad illustration and detailed overview of the project.

1.6.4 Supporting Information

This section is for the supporting information that might be necessary.

2 Overall Description

2.1 Product perspective

Boikini is a new, self-contained online platform for buying and selling books. It is a web-based system that allows users to browse, search, and purchase books from sellers, who can list their books for sale on the platform.

The system is designed to be user-friendly, secure, and scalable, with features that enable users to search for books based on various criteria, place orders, and track the delivery of their orders. The platform will also allow sellers to manage their inventory, process orders, and receive payments securely.

Boikini is not part of an existing product family, nor is it designed to replace any existing system. Instead, it is a new product that seeks to meet the growing demand for online book sales, offering a convenient and accessible platform to buy and sell books.



As a web-based platform, Boikini will be developed on React on frontend and Python and Django on backend. The system will also interact with external databases, such as Structured Query Language (SQL) and Non-Structured Query Language (NoSQL), and exchange data in various formats, including JavaScript Object Notation (JSON) and Unified Modeling Language (UML). Any device with modern browser and stable internet connection will be able to access the website.

2.2 Product functions

The Boikini online platform must perform the following major functions to meet the needs of its users:

- **Book search:**

The system should allow users to search for books based on various criteria, including title, author and subject.

- **Book ordering:**

The system should enable users to place orders for books they want to buy, including the ability to add items to a shopping cart, review and edit the cart before checkout, and complete the purchase using a secure payment system.

- **Order tracking:**

The system should allow users to track the status of their orders, including estimated delivery times, shipment tracking information, and order history.

- **Seller inventory management:**

The system should provide sellers with the ability to list their books for sale, manage their inventory, update pricing, and view order history.

- **User account management:**

The system should enable users to create and manage their accounts, including setting up a profile, managing saved searches, and reviewing past purchases.

- **Reviews and ratings:**

The system should allow users to leave reviews and ratings for books they have purchased, and also provide users with the ability to view and filter reviews from other users.

- **Notifications:**

The system should provide users with automated notifications related to their orders, including order confirmations, shipping updates, and purchase history.

- **Contacting Sellers or Developers:**

The system should provide necessary interface for the buyers to contact the sellers and to report an issue or provide feedback to the developers in case of any technical difficulties.



2.3 User Profiles

- **Book Buyers:**

These are the primary users of the system who will browse and purchase books online. But they will need to create an account first to make a purchase. They will have basic computer knowledge and will use the system frequently.

- **Book Sellers:**

These users will sell books on the platform. They will be able to add or remove books, confirm orders, view customer information and contact with any customer. They will have expertise in listing books and will use the system frequently.

- **Administrators:**

These users will manage the system managing user accounts and monitoring the system. They will have the authority to delete any user account and books directly from the database. They will have technical expertise and will use the system frequently.

- **Guests:**

These are users who do not have an account on the platform. They will be able to browse books but will not have the ability to purchase or sell. They will have basic computer knowledge and will use the system infrequently.

The most important user classes for this product are the book buyers and book sellers. The system must be user-friendly, secure, and reliable to ensure their satisfaction. The administrators must also have access to a comprehensive management system to ensure smooth operations. The guest users are less important, but still require an easy-to-use browsing experience to attract them to the platform.

2.4 Constraints

- **Small geographical Scope:**

Only users from Bangladesh can buy books. The system can't be used all over the world.

- **Less options:**

The system is like an online book store. Users are buying books from a particular place. This might lessen the numbers and varieties of books.

- **Payment Restrictions:**

Buyers will be able to pay only using particular methods. Users will need those payment methods to make a purchase.

- **Language Restrictions:**

The system will be developed using only English language.

- **Third Party dependence:**

The system will be dependent on a third party for financial transaction and delivery purpose.



These constraints will limit the options available to the developers and will need to be considered during the development process.

2.5 Assumptions

- The users have access to a stable internet connection to access the online platform.
- The users have basic computer skills and can operate an online platform.
- The users have a valid payment method to purchase the books.
- The users provide accurate information regarding their location and contact information.

2.6 Dependencies

- The project is dependent on the availability of the third-party payment gateway to process the online transactions.
- The project is dependent on the availability of the third-party shipping services to deliver the books to the users.
- The project is dependent on many APIs.
- The project will need a technical team identified as admins to manage the technical side of the website.

3 Specific Requirement

3.1 Overview

- **User Registration:**
The system will allow users to register with their basic information, including their name, email address, and a password.
- **User Login:**
The system will provide a login page for registered users to access their accounts.
- **Book Search:**
The system will provide a search functionality for users to search for books by title, author, or keyword.
- **Book Details:**
The system will display book details such as the title, author, publisher, price and availability.
- **Order Placement:**
The system will allow users to place orders for books by adding them to their shopping cart and providing their shipping address and payment information.



- **Order Tracking:**

The system will provide users with the ability to track their order status, including the estimated delivery date and shipping carrier information.

- **Payment Gateway:**

The system will integrate with a payment gateway to securely process online payments for orders.

- **User Reviews:**

The system will allow users to write and view reviews for books.

- **Sellers:**

The system will enable sellers to manage book listings, orders, users, and reviews.

- **Security:**

The system will implement industry-standard security measures to protect user data and prevent unauthorized access.

3.2 Functional Requirement

ID	3.2.1
Name	User Authentication
Description	The system must allow users to create an account and log in to access the platform. Authentication must be secure and use strong password policies, and the system must ensure that only registered users can purchase books.
Priority	High

ID	3.2.2
Name	Search Function
Description	The system must have a search function that allows users to search for books according to names, authors, publishers or genres. The search function should return results that are relevant to the user's query and sort them by relevance.
Priority	High



ID	3.2.3
Name	User Profile Management
Description	Once logged in, users must be able to edit their profiles, update their personal information, and add a profile picture. The system must also allow users to delete their accounts if desired.
Priority	Low

ID	3.2.4
Name	Payment Gateway
Description	The system must implement payment gateways to provide payment methods to users.
Priority	High

ID	3.2.5
Name	Notification System
Description	The system must have a notification system that alerts buyers about the confirmation of their orders and about new books. It must notify the sellers about new orders so that they can confirm them. The system must also allow users to customize their notification settings.
Priority	High

ID	3.2.6
Name	Review
Description	The users must be able to review books and rate them. They can also read other people's reviews.
Priority	Low



ID	3.2.7
Name	Contacting
Description	Buyers and sellers must be able to contact with each other in case of any queries.
Priority	Low

ID	3.2.8
Name	Reporting Function
Description	The system must have a reporting function that allows users to report technical issues of the platform. The system must have a clear process for handling reports and enforcing platform guidelines.
Priority	Low

ID	3.2.9
Name	Order Status
Description	The system should give the buyers the current status of their orders, such as, if the order is accepted by the buyers or it is shipped. It also show information about the payment is accepted or not.
Priority	Low

ID	3.2.10
Name	Order and customer information
Description	The system should give the sellers information about what orders have been placed and necessary information like the location of the customer and the payment method.
Priority	High



3.3 Non Functional requirement

3.3.1 Usability

1. The software will have a user-friendly interface that is easy to navigate and understand by the target users.
2. The software will be intuitive, requiring minimal training for new users. The product will be easily customizable, allowing users to tailor their experience to their specific needs.
3. The software will have clear and concise documentation available to users.
4. The product will be consistent in its design and functionality throughout all features and components.
5. The product will comply with established industry standards and best practices for usability.
6. The product will provide clear and concise error messages and recovery options in the event of errors or failures.
7. The product will have fast response times and minimal downtime to ensure maximum user productivity.
8. The product will be able to handle concurrent users without significant decrease in performance.

3.3.2 Reliability

Reliability refers to the ability of the system to perform its functions without failure, and with consistent and accurate results. The following non-functional requirements related to reliability should be considered:

- **Availability:**

The system should be available for use as required. The percentage of uptime should be clearly defined.

- **Consistency:**

The product must be consistent in its performance. It should notify customers and sellers in the right time about their orders.

- **Reliable Payment:**

The system should implement reliable payment and delivery methods for the users.

- **Error handling:**

The system should be able to detect errors and take appropriate action. Errors should be logged and made available for analysis.

- **Fault tolerance:**

The system should be able to continue functioning in the event of a failure. It should be able to recover quickly and gracefully from faults.



- **Recoverability:**

The system should be able to recover from failures with minimal loss of data. The time required to recover should be specified.

- **Scalability:**

The system should be able to scale as the number of users or transactions increases. The performance degradation should be defined for each level of scaling.

3.3.3 Performance

Performance is a crucial aspect of any software application, especially for an online marketplace like "Boikini". To ensure that the website runs smoothly and efficiently, the following performance requirements have been identified:

- **Page Load Time:**

The maximum page load time for any page on the website should be no more than 2 seconds, as slow page load times can lead to frustration for users and a decrease in overall user satisfaction.

- **Concurrent Users:**

The website must be able to handle at least 100 concurrent users without any noticeable decrease in performance. This is important to ensure that the website can handle high traffic volumes during peak periods without slowing down or crashing.

- **Search Time:**

The search functionality of the website should be able to return relevant results in no more than 1.5 seconds, as users may quickly lose interest and move on to another website if the search takes too long.

- **Order Processing Time:**

Once an order is placed, the system should be able to process the order within 24 hours. This includes verifying the payment information, updating the inventory, and shipping the order to the correct location.

- **Scalability:**

The website should be designed to be scalable so that it can accommodate future growth without any major performance issues. This includes the ability to handle an increase in concurrent users, larger databases, and increased traffic volumes.

- **Error Handling:**

The system should be able to handle errors gracefully and provide appropriate error messages to users. This includes handling errors related to database connections, network connectivity, and other system-level errors.

By adhering to these performance requirements, Boikini can ensure that its website is fast, reliable, and user-friendly, resulting in a positive user experience and increased user satisfaction.



3.3.4 Supportability

Supportability refers to the ease and efficiency with which the software can be maintained, upgraded, and supported throughout its lifecycle. Some supportability requirements for the Boikini project might include:

- **Modularity:**

The software should be designed in a modular way, with clear separation of functionality and clear APIs, to facilitate future updates and maintenance.

- **Documentation:**

Comprehensive documentation should be provided to aid in the maintenance, configuration, and troubleshooting of the software.

- **Error logging and reporting:**

The system should be capable of logging and reporting errors in a clear and concise manner, allowing developers to easily identify and fix any issues.

- **Scalability:**

The system should be designed to easily scale up or down to handle changing demand, without significant modifications to the underlying architecture.

- **Compatibility:**

The software should be compatible with a range of operating systems and browsers, to ensure maximum accessibility for users.

- **Support channels:**

Clear and accessible support channels should be provided to users, including email, phone, and chat support, as well as online documentation and FAQs. Users should be able to contact the technical team in case of major technical issues.

- **Upgradability:**

The software should be designed with a clear upgrade path, allowing users to easily upgrade to new versions of the software as they become available.

By focusing on supportability, the Boikini project can ensure that the software is easy to maintain and upgrade, while also providing clear support channels for users who may encounter issues or have questions.

3.3.5 Implementation

- **Programming Language:**

The web application will be built using a combination of front-end and back-end programming languages. For the front-end, the preferred language is JavaScript. For the back-end, the preferred programming language is Python.



- **Database:**

The web application will require a database to store user data, book details, order history, and payment information. The preferred database management system for this project is MySQL. The database will be hosted on a cloud-based platform to ensure easy access and scalability.

- **Framework and Libraries:**

The web application will be built using a combination of front-end and back-end frameworks and libraries. The front-end framework will be React, while the back-end framework will be Django.

- **Deployment and Hosting:**

The web application will be hosted on a cloud-based platform to ensure easy access and scalability.

- **Security:**

Security will be a top priority during the implementation phase. The web application will use HTTPS for all communication between the client and the server. User data and payment information will be encrypted using industry-standard encryption algorithms.

3.3.6 Scalability

Scalability is an important aspect of the Boikini project that should be considered. The system should be designed to handle a growing number of users, transactions, and data with minimal performance impact. The following are the scalability requirements for the system:

- The system should be able to handle at least 100,000 active users and 1000 concurrent users without experiencing any downtime or significant performance degradation.
- The system should be able to handle at least 1,000 book listings, and 1000 transaction requests daily, without any data loss, transaction errors or downtime.
- The system should be designed to allow for easy addition of new features, modules or APIs to meet the growing needs of the business without significant performance impact.
- The system should be designed to support horizontal scalability, allowing for the addition of more servers, resources or nodes to improve system performance without significant re-design or configuration changes.
- The system should be designed to support load balancing and failover, ensuring that the system remains available even when a server or resource fails.
- The system should be designed to optimize performance by caching frequently accessed data, using appropriate database indices, and implementing efficient algorithms and coding practices.

By considering these scalability requirements, the Boikini system will be able to grow with the business, accommodate a growing user base, and deliver optimal performance and user experience.



3.3.7 Security

Security Requirements for Boikini:

- **User Authentication:**

The system shall require users to create an account and provide valid login credentials to access the system. The system shall ensure that all login credentials are encrypted and stored securely.

- **Data Encryption:**

The system shall use encryption to secure all data transmissions over the internet. The encryption shall use industry-standard algorithms to prevent unauthorized access to user information.

- **Secure Payment Processing:**

The system shall implement a secure payment gateway to process transactions.

- **Protecting User Data:**

The system will ensure the safety of users' data from any outside access.

- **Privacy Policy:**

The system shall have a clear and concise privacy policy that outlines the types of data collected, how the data is used, and who has access to the data. The privacy policy shall be readily available to all users.

- **Password Complexity Requirements:**

The system shall require users to create strong passwords with a minimum length and a combination of characters, including upper and lowercase letters, numbers, and special characters. The system shall also enforce password change policies to ensure that users change their passwords regularly.

3.3.8 Maintainability

Maintainability is an important aspect of software development, as it ensures that the product can be easily maintained and updated over time. The following are some specific maintainability requirements for the "Boikini" project:

- **Modular design:**

The system should be designed with a modular structure that allows for easy maintenance and updates. Each module should be independent and loosely coupled, so that changes to one module do not affect the others.

- **Clear documentation:**

All code, including comments and documentation, should be clear and well-organized. This will make it easier for developers to understand and modify the code as necessary.

- **Standard coding practices:**

The system should follow standard coding practices, such as naming conventions, code formatting, and error handling. This will make it easier for developers to maintain and update the code, even if they did not originally develop it.



- **Version control:**

The system should be stored in a version control system, such as Git, to track changes and allow for easy rollbacks. This will ensure that any changes made to the system can be easily tracked and reverted if necessary.

- **Unit testing:**

The system should be designed with unit tests that cover all major functionality. This will help developers catch bugs and errors early, and make it easier to maintain the system over time.

- **Error reporting:**

The system should include an error reporting system that logs errors and provides detailed information about the issue. This will help developers quickly identify and fix any issues that arise in the system.

By adhering to these maintainability requirements, the "Boikini" project will be easy to maintain and update over time, ensuring that the system remains reliable and functional for users.

3.3.9 Testability

Testability is a critical aspect of software development to ensure that the product meets the functional and non-functional requirements. The following are the testability requirements for the Boikini project:

- **Test Case Management:**

The system should have a robust and reliable test case management system that enables developers to create, maintain, and execute test cases effectively. The system should have the capability to track and manage test results, as well as identify and report defects.

- **Automated Testing:**

The system should be designed to allow for automated testing of the various functions and features. This will enable developers to execute a large number of test cases in a short amount of time and identify defects earlier in the development cycle.

- **Performance Testing:**

The system should be designed to allow for performance testing, which will help developers to identify any bottlenecks and optimize the system. Performance testing should be done under realistic load conditions.

- **Security Testing:**

The system should have a robust security testing plan that verifies the security of the system against various threats.

- **User Acceptance Testing:**

The system should be designed to allow for user acceptance testing, which involves testing the system with actual end-users. This will ensure that the system meets the needs and expectations of the users and is easy to use.



Overall, the Boikini project should be designed with testability in mind, to ensure that the product meets the functional and non-functional requirements, is robust and reliable, and meets the needs and expectations of the users.

3.4 System Models

3.4.1 Scenarios

3.4.1.1 User registration

A user accesses the system for the first time and is prompted to register. The user provides their name, email address, password and a delivery address.

3.4.1.2 Searching for books

A user can search for books by entering the book title or author name in the search box. The system displays the books that match the search criteria.

3.4.1.3 Placing an order

A user selects the books they want to purchase, and enters their payment details. The system validates the payment and confirms the order.

3.4.1.4 Tracking an order

A user can track the status of their order from the order history section of their account. The system displays the current status and estimated delivery date.

3.4.1.5 Writing a review

A user can write a review for a book they have purchased. The system verifies the review for authenticity and publishes it on the product page.

3.4.1.6 Managing a user account

A user can manage their personal information, including name, email, password, and delivery address, and view their order history.

3.4.1.7 Seller login

A seller can log in to the system to manage the product inventory, prices, and discounts.

3.4.1.8 Adding a book to inventory

A seller can add new books to the inventory by entering the book details and uploading an image.

3.4.1.9 Setting prices and discounts

A seller can set prices and discounts for each book in the inventory.



3.4.1.10 Managing orders

A seller can view and manage orders, including confirming the order, changing the order status, and canceling an order.

3.4.2 Use cases

Name	Creating a User account
Actor	Buyer,Boikini
Flow of events	The users will be able to create an account. While creating the account the users will need to give necessary information like their delivery location and payment information.
Entry Condition	Users will give information to create the account
Exit Condition	The account will be created and the information will be saved in the database

Name	Searching for books
Actor	Buyer,Boikini
Flow of events	The users will be able to search books based on the name, author, publishers or genres
Entry Condition	Users will give the name or author of the book in the searchbar
Exit Condition	The system will retrieve the information about the book and display it

Name	Add book to the cart and make a Purchase
Actor	Buyer,Boikini
Flow of events	The users will be able to add their desired book to the cart and continue shopping or finish shopping by confirming the order.
Entry Condition	Users will click add to cart and confirm the order by giving necessary information
Exit Condition	The system will retrieve the information and send the order to sellers to customer for confirmation



Name	Get and confirm a new order
Actor	Seller,Boikini,Buyer
Flow of events	The seller will get the new orders and can view necessary information related to the order and can confirm the order.
Entry Condition	The seller will get notified about a new order and will confirm or decline the order
Exit Condition	The confirmed order will be saved in the database and the buyers will get notified about their order

Name	Add or remove book
Actor	Seller,Boikini
Flow of events	The seller will be able to add new books or remove books when they are out of stock.
Entry Condition	The seller add or remove books
Exit Condition	The books will be saved or removed from the system database and will be available or unavailable for purchase

Name	Order Status
Actor	Buyers,Boikini,sellers
Flow of events	The buyers will be able to get their order status such as if the order is shipped , if the payment is received and the estimated days of delivery.
Entry Condition	The seller will update the status of the order
Exit Condition	The order status will be saved will be displayed to the users

Name	Reviews
Actor	Buyers,Boikini
Flow of events	The buyers will be able to give reviews about a particular book. Other buyers can see it when they will search for the book.
Entry Condition	The buyers will give a review about a book
Exit Condition	The system will save it along with the book information and display it when other buyers search it



Name	Contact
Actor	Buyers, Sellers
Flow of events	The buyers and seller will be able to contact each other in case of any failure such as payment not received or the book is not delivered in the estimated time.
Entry Condition	The buyers or sellers will send a message
Exit Condition	The buyers or sellers will receive a message

Name	Feedback
Actor	Buyers, Admins
Flow of events	The buyers will be able to report any technical issues or any feedback to the developer team.
Entry Condition	The buyers will report or give feedback
Exit Condition	The developer team known as admin will receive it



3.4.3 Use case model

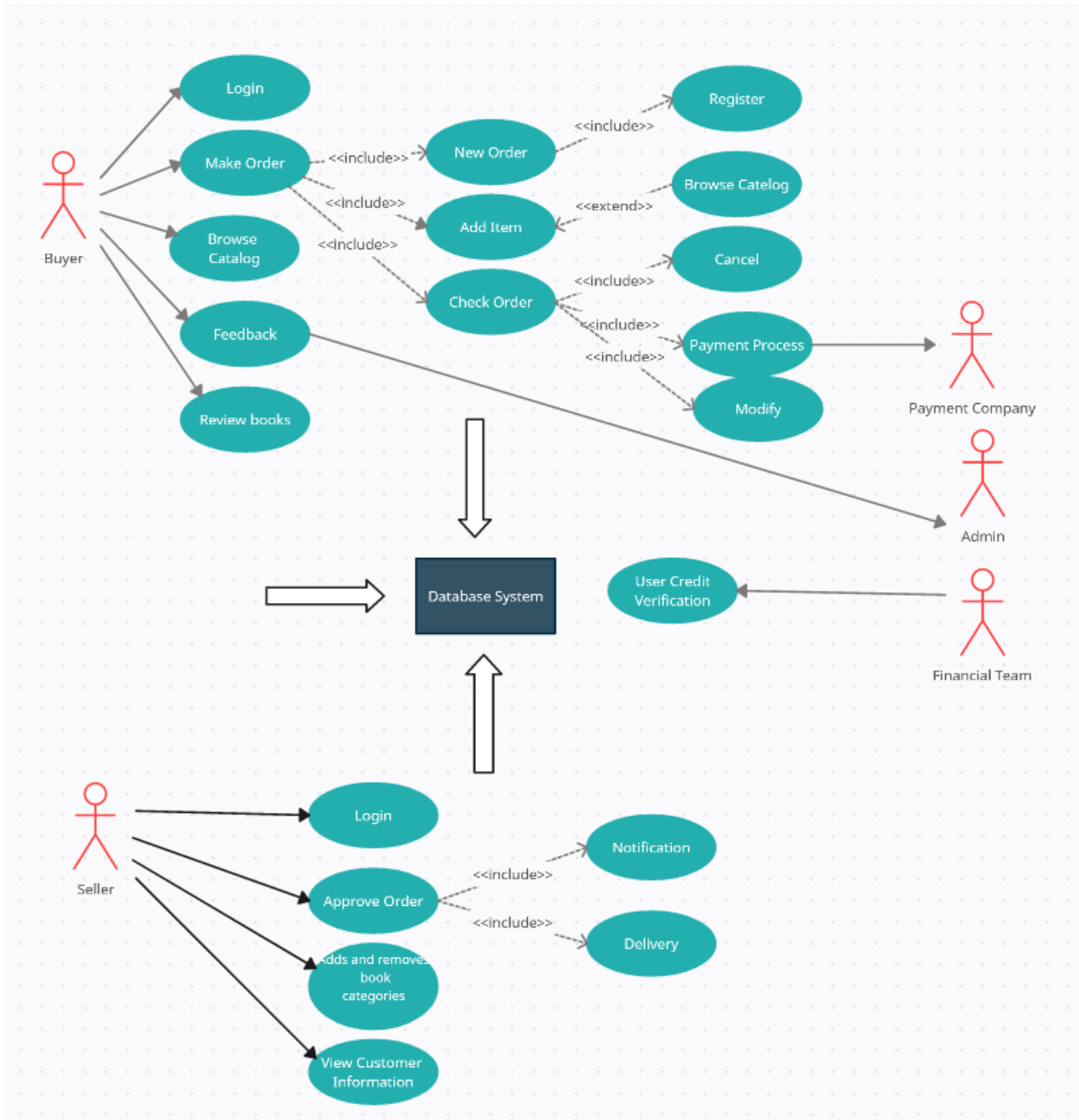


Figure 1: Use Case Diagram



3.4.4 Dynamic Model

3.4.4.1 Sequence Diagram

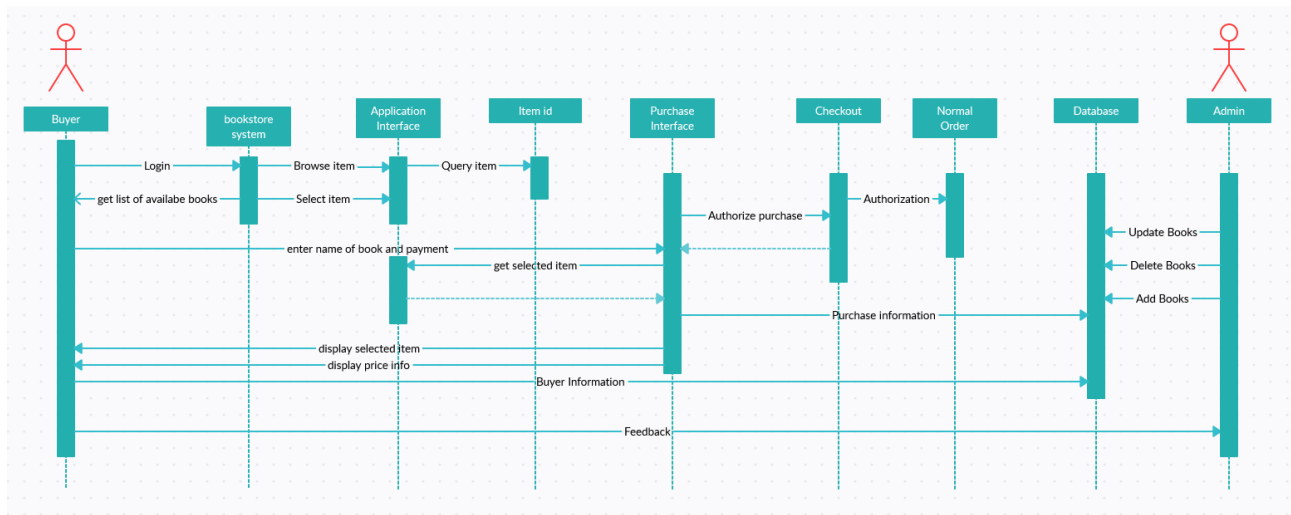


Figure 2: Sequence Diagram



3.4.4.2 Statechart Diagram

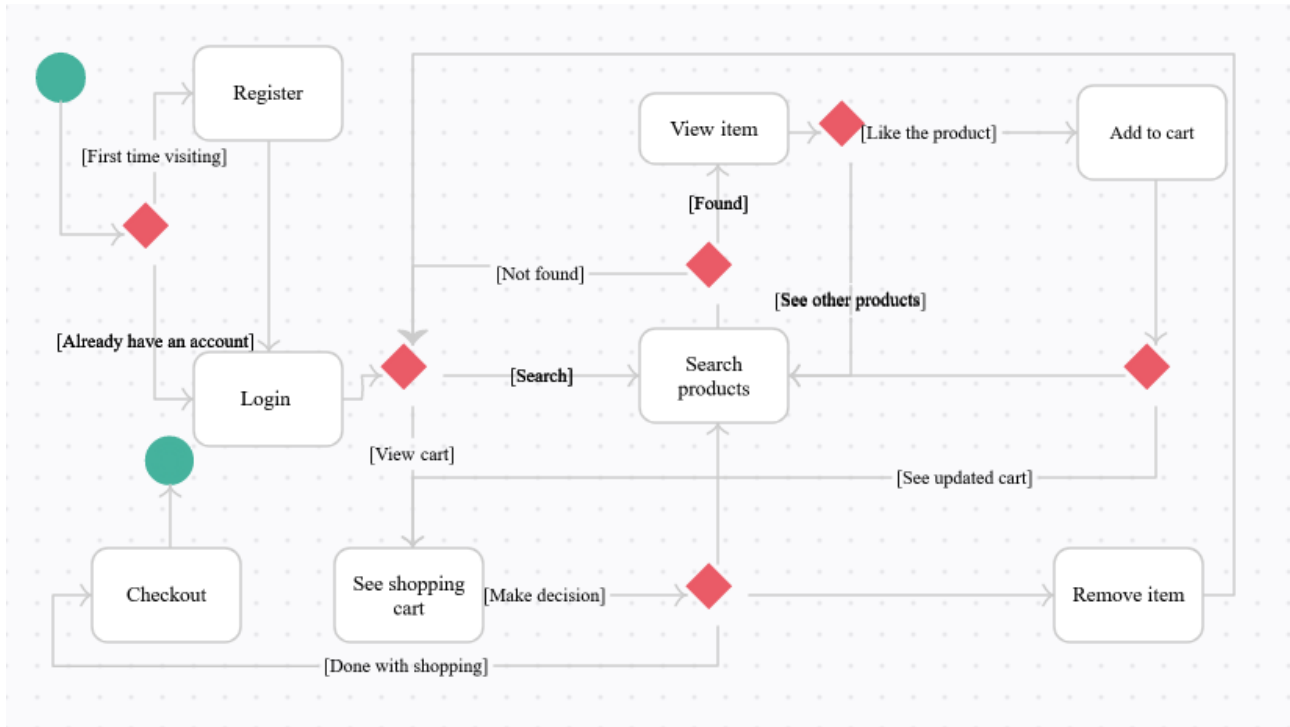


Figure 3: Statechart Diagram



3.4.4.3 Activity Diagram

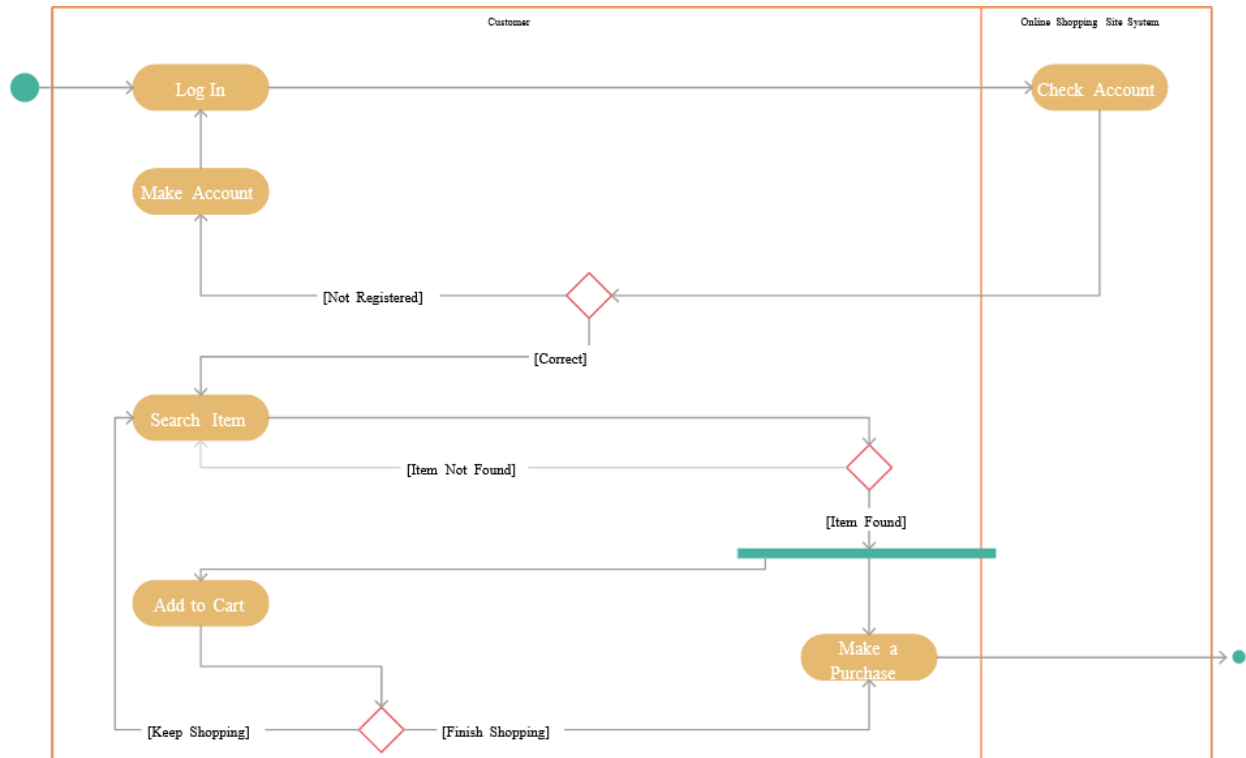


Figure 4: Activity Diagram

3.4.5 Interfaces

3.4.5.1 User Interface

- **Main Page**

The main page shows all the options for the system. It includes everything from adding a new books, search option, book suggestion, etc. Users will be able to navigate to all of these pages. The interface should be intuitive and easy to navigate to allow users to access the desired functionalities without confusion. A well-organized navigation bar with links to different pages of the website should be available on every page.

- **Search functionality**

The interface should have a search bar that allows users to search for books based on various search criteria such as author, title, genre, and keywords.

- **Book details page**

The book details page should contain all the relevant information about the book, including the title,



author, price, description, reviews, and related books.

- **Shopping cart**

The interface should include a shopping cart that allows users to keep track of the books they intend to purchase, view the total cost of their orders, and modify or remove items from their cart.

- **Checkout process**

The checkout process should be streamlined and user-friendly to enable users to complete their purchases without any hassle. The interface should include a secure payment gateway that accepts various payment methods.

- **User profile**

The interface should allow users to create and manage their profiles, view their purchase history, and manage their preferences and notifications.

- **Feedback**

This tab is for providing feedback to the developers over possible improvements to the app.

3.4.5.2 Software Interface

The project is mainly a web application. So necessary web APIs will be used. The project will use ReactJS in the frontend and Django in the backend. The project will need to connect to a database to store and retrieve book information, user profiles, and order details. The database system will be MySQL version 8. The system will use a standard API for database access and will use secure communication protocols to ensure that user data is protected.

3.4.5.3 Hardware Interface

Any device with a modern browser should be sufficient to browse "Boikini". It must have a working and stable Internet connection in order to access our web application.

4 Supporting Information

As of this moment, we do not have any supporting information that needs to be documented. If we face such an information, we will update it with the necessary information.