Faculty of Engineering and Information Technology

Department of Computer Science

Introduction to Software Engineering

Instructor: Dr. Amjad Rattrout

Team members:

Nowar Daraghma	202010557
Nooraldein Asia	202011016
Zakaria Ghazal	201912096

TABLE OF CONTENTS

GOODS RENRAL SOFTWARE

1. Preface

1.1. Problem Statement

Often people find themselves in situations where they need a specific product that they might not be able to afford, or they simply don’t feel the need to buy just to use once or for a limited amount of time. For example, many university projects require the students to use some items that not everyone find necessarily affordable or worth buying just to use on one project then throw it out.

Or in other cases, people might want to rent a product to test before making full investment in it. Renting provides the ability to perform product tests at a pace that suits the users.

1.2. System Overview

Our application will provide a platform for people to rent products (renters) or offer products for rent (retailers).

It allows users to rent items for a predefined amount of time in exchange for money. Not only does it help renters save money and reduce consumption, but it also allows retailers to gain some money and help their community members.

The system allows users to have an account in order to access the services securely without having their information stolen. It allows users to easily find the desired item by searching for it on the search option. It also allows users to rent any available item using credit card or other online transactional methods. A shipping option is also available for those who would like to receive the rented item without having to leave their homes.

2. Introduction

2.1. Advantages of the Software

Reasons why you might consider using this rental software:

- Save money.
- Reduce consumption.
- Test a product before using it, or fully investing in it.
- Avoid major repairs and expenses.
- You can get all the products you need in one place.
- Avoid long-term commitment.
- Have multiple options.
- Avoid storage.

2.2. Functions of the Software

The app's main functionality is processing the user's inputted search and looking for the best match. Besides actively recommending a list of items the user might be interested in buying in order to keep them active on the app.

2.3. Feasibility Study

- ***Hardware*:** we will only need a device on which the application will be programmed.
- ***Software*:**
  - Sufficient knowledge in programming languages used to build web apps is required. For example, JavaScript, since it's one of the most versatile programming languages. PHP/MySQL is also needed for database design and implementation.
  - We need access to a hosting server/website, and payment services.
- ***Economic Feasibility*:** 30000$, to be spent on teaching the team app design and UI implementation as well as purchase all necessary programs such as payment services and licensing the software as well as hiring a people for testing and analysis.
- ***Schedule***: 26 weeks, around the end of the 3rd University year.

2.4. Stakeholders

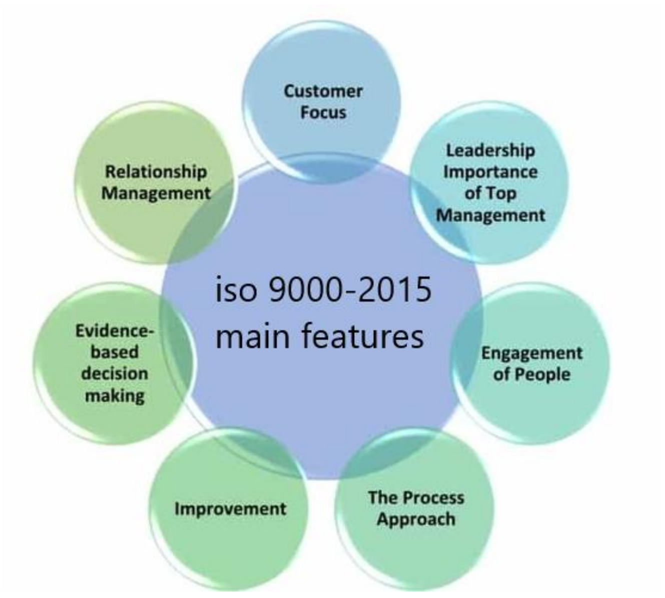
- Any person interested in earning money and offering a product for rent.
- Any person interested in renting a product in order to save money or reduce consumption.
- Any organization/company that is willing to advertise their products for the right price.

2.5. Risks

- Since the idea behind the app is fairly new, the success of the product is hard to determine.
- It will be very hard to respond to changing requirements due to the low budget.
- Since the software is a web app and not a website, customers may not be interested in using up space for an e-commerce app.
- Since the range of our application is limited to the West Bank in Palestine only, the market share could be lower than expected.

2.5. Standard to be used

The standard we have decided to use is the ISO 9000-2015 standard as we want to work towards a well-managed system with great quality assurance that provides confidence to users with the ability to provide products and services as well as the users who seek these products and services.


iso 9000-2015 main features
Customer Focus
Leadership Importance of Top Management
Engagement of People
The Process Approach
Improvement
Evidence-based decision making
Relationship Management

3. Glossary

Hosting server/website: is the business of housing, serving, and maintaining files for one or more Web sites.

Web application: is application software that is accessed using a web browser.

Market share: the percentage of an industry's sales that a particular company owns.

Plan driven: is a software development method which attempts to plan and develop all of the features a user might want in the final product and determines how all those features are to be developed.

Dark-mode: is a system appearance setting that uses a dark color palette to provide a comfortable viewing experience tailored for low-light environments.

4. Requirements

4.1. Stories and Scenarios

4.1.1 First Story

Nowar is a new user who's looking forward to creating an account in order to securely access the application services. Upon opening the app, Nowar is directed to a page containing a login form with the option to register if the user is new. Being a new user, she clicks on the *Register* button. The system directs her to a register form page where she has to input all necessary information to sign up and confirm the account.
When Nowar enters all of the necessary information and the system confirms that all data is valid, the system enables the *Complete Registration* button. When creating an account, users are asked to input their email or phone number, based on Nowar's choice, she will be sent a confirmation code as either an email, or a text message to her mobile phone.
The system will actively check that all information inputted is valid and follows the correct format. It will also check that the inputted confirmation code matches what was sent.
Once Nowar confirms the account, they system will finally direct her to the login page, where she's given full access to the system after entering the sign up information.

4.1.1 Scenarios

1. As a user, I expect a secure login to the system so that all my information is private and only accessible by me.
2. As a user, I should be able to create an account fairly easily and safely using personal information. And I expect to be sent a confirmation code to verify my identity.
3. The system should immediately alert users if any invalid data was inputted.

4.1.1 Task Cards

Task one: secure login
In order to guarantee the privacy of users and protect their information, when logging in to the system users have to authenticate themselves to confirm their identity. Then they will immediately be sent to the main page of the app where they can take full advantage of the system's services.

Task two: secure sign up
Users will be asked to input some personal information in order to create an account. Then a confirmation code must be sent to them, using either their email or phone number, based on their choice.

4.1.2 Second Story

Noor is a renter who is looking for a product to rent. While looking for the product, Noor clicked on the *search* icon and a search bar appeared as well as a *category* button.
If Noor clicks on the search bar, the system will allow him to type in the name of the desired product. When he inputs the name of the item or a keyword associated with it and clicks on the

search icon again, the system will direct him to a page containing all products that has a matching name or keyword to the inputted ones. If Noor clicks on the *category* button, the system will display a bunch of hyperlinks/photos, each represents a category option. If Noor chooses a category from the hyperlinks, the app will direct him to a page containing a list of items of the chosen category. The system will actively check the available products of the specified name/keyword or chosen category. The system will notify the user when an item isn't available, and calculate when it would be available next.

4.1.2 Scenarios

- As a renter I expect the system's UI to include an easy-to-locate search icon/bar, through which I can easily find my desired product.
- As a renter, when I enter a name or a keyword in the search bar, the system should direct me to a page that contains a list of items that match my search.
- As a renter, when I click on the category button, I expect the system to showcase all created categories.
- As a renter, when I click on a category, the system should direct me to a page containing a list of all items that are classified under that category.
- As a renter, I expect the system to alert me when an item isn't currently available, and show me when, approximately, it'll be available.

4.1.2 Task Cards

Task one: designing an easy-to-use UI
In order to accommodate as much users of all ages as possible, our goal is to create a simple and convenient user interface, with a color that's comfortable to the eyes, and understandable font and icons size. In addition to offering custom options, like dark mode or night light.

Task two: product searching via name
The system shall allow users to input the name of their product of interest or any related keywords, and showcase a list of all available products whose name or descriptive keywords match the inputted ones.

Task three: product searching via category name
The system shall give the user an option to search using category names. Upon choosing that option, users will be provided with a list of all created categories. When they click on a category, a list of all available items that are classified into that category will be displayed.

Task four: availability of items
If the user's desired item is currently available, they can proceed and finish their purchase process. If not, users will be notified of that fact, and will be shown a list of available recommended products similar to the original.

4.1.3 Third Story

Zakaria is a renter who's looking to rent an item from a retailer. After clicking on the *checkout* button, he is directed to a page that contains 2 radio buttons, each represents a payment option, which are: *using card* (could be master card, visa card, debit card … etc.) and *other service* (could be Venmo, Jawwal pay or others).
If Zakaria selected *using card*, a form asking for all necessary information is shown, where he has to input his card's data. If he selects other service, a different form asking for the information of that service will appear.
If Zakaria inputs all valid data for either payment option, a checkbox will be enabled, and asks if he would like to use the system's *shipping service* as well. If Zakaria chooses to use the app's shipping service, he will be directed to a page containing a form, where he can input all essential shipping information. After entering all the shipping information, a window will show the new price that sums the item's price + shipping fees. At the end a *confirm purchase* button will be enabled.
The system shall constantly check if the inputted payment information is valid, in case of an error, the system shall alert users of any invalid data. The system should also check if the user's balance if enough to complete the purchase process, if it isn't, the system will notify users and ask for a different card's information.
Upon completing the purchase, Zakaria should be sent an email confirming that the purchase has been made as well as information about the time of arrival.

4.1.3 Scenarios

- As a user, I expect the app to provide multiple payment options, and a place where I can securely enter my information without fear of it being leaked/stolen.
- As a user, I expect a shipping service provided by the system.
- As a user, when I complete my purchase, I expect to be sent a confirmation email that informs me of the expected time of arrival.

4.1.3 Task Cards

Task one: multiple payment options
Upon checking out, the user must input their billing information in order to complete the transaction/s. The user will be presented with multiple options (in the form of a radio button) for card transactions, venmo and even Jawwal pay. Using the inputted billing information, system will check if the user can fully complete the transaction and receive the desired item. If the card information is fully valid, the button to confirm purchase will be enabled.

Task two: shipping service
After confirming the purchase, users can choose to use the shipping service the app provides, and they will be presented with a form to input their data. The system should calculate the new total that includes the product's price and the shipping fees. Upon the completion of the purchase process, users will be sent an email that confirms the process and informs them of the expected time of arrival of their product/s.

4.1.4 Fourth Story

Aladdin is a user who wants to check out the information of the products he finds interesting. Upon searching the name of the product and being directed to the page containing these products, he finds a picture of a product, along with the name, and price. And a *filter* button at the top left side of the page. If Aladdin clicks on the picture or the name of the product, the system will direct him to a page for the product that contains every detail about the product, more images, and a button to *add to cart.* If he clicks on *filter* button, the system will display a range where the users are asked to input 2 integers, and a *confirm* button. When he confirms the filter, he is directed to the same page but now it only contains the products that are priced within the entered range. The system will actively check that the items being displayed are within the specified range.

4.1.4 Scenarios

- As a user I expect the app's UI to show the basic information about each product right next to a picture of it.
- If the user wishes to check out more details about an item, they can do so by clicking on the picture.
- Users will be provided with a *filter* option where they can input a price range, and will be shown a list of items within that range.
- As a user, I expect an *add to cart* button, that adds desired products to my cart.

4.1.4 Task Cards

Task one: product information
The app's UI must offer basic information about product upon searching for it, like its name and price, along with a picture of it. And users wish to see more information about it, they can do so via clicking on the picture, where they will be provided with a full description of the product, the retailer name, how many times it's been rented, and more pictures of it.

Task two: filter products
If users have a certain budget and would like to only check out products within their budget, they can click the filter button that shows items within a price range. After clicking, they will have to input 2 integers, minimum price and maximum price, to indicate the range.

Task three: adding to cart
Each account has a cart, where users can save items they wish to buy. An add to cart button will be added next to each product, so that users can easily save their item of interest.

4.2. Testing Cards

Test 1: registration process.
Input: 1. User's first name as a string. 2. User's last name as a string. 3. User's address as a string. 4. User's phone number as an integer. 5. User's email as a string. 6. User selecting an option between retailer/renter. *Tests:* 1. Check if the phone number or the email already exists in the database. If so, users will shall be asked to enter a different one. 2. Check if the Email is in a valid format and if the password is above 8 characters and contains special characters. *Output:* 1. A message that an account has successfully been created if both tests passed. 2. An error message to alert the user that the email/phone number they've entered already exists, or that the confirmation code is incorrect. 3. An error message to alert the user that the email/password they've entered already is invalid or not strong enough.

Test 2.1: searching for a product by name.
Input: A product name as a string. *Tests:* Check if the entered name matches any of the available items in products' database.

Output:

1. Showcasing the product and all the information available about it.
2. An error message indicating the product isn’t available.

Test 2.2: searching by category.

Input:

A category name as a clickable link/button.

Tests:

1. Check if the clicked name matches any of the categories already inputted in the system.
2. Check if the category has any items available.

Output:

1. Showcasing the list of items classified under the entered category.
2. An error message indicating that the category is still empty.

Test 3: Products’ information.

Input:

Clicking on the product’s photo or name.

Tests:

1. Check if the product is currently available for renting or not.
2. Check if the correct product details are displayed on the screen.

Output:

1. The product’s page with all full details replaced with “Not available currently”.
2. The product’s page will the full details displayed and *Add to Cart* button enabled.

Test 4.1: Payment process by card.

Input:

1. Type of card selected.
2. User’s card number.
3. User’s card expiry date.
4. Card’s CVV number.
5. Users card name.

Tests:

1. Input a correct number, CVV number, expiry date and name but the expiry date has been passed already or not.
2. Input a correct number but not CVV number and vice versa as well as a correct name and expiry date.
3. Input the data of a credit card while the type of card selected is not credit card.
4. Input all correct data of the credit card but the card has no balance left or not enough to purchase the items.

Output:

1. A prompt that states that the purchase has successfully been made.
2. An error message to alert the user that the card is invalid (all tests shall display that).
3. An error message to alert the user that the card is already expired and cannot be used to complete the purchase.
4. An error message to alert the user that the card is balance is not enough to complete the purchase.

Test 4.2: Payment process by Venmo or Jawwal.

Input:

1. Type of other payment selected.
2. Email of the user’s payment account.
3. Password of the user’s payment account.

Tests:

1. Input the email and password for the wrong website/service.
2. Input the correct email but wrong password and vice versa.
3. Input the correct email address and password for the correct type of payment.

Output:

1. A prompt that states that the purchase has successfully been made.
2. An error message to alert the user that the account does not exist.
3. An error message to alert the user that the email address or password is incorrect.
4. An error message to alert the user that the account is balance is not enough to complete the purchase.

4.3. Functional Requirements

They are functions that developers *must* implement to enable users to accomplish their tasks. Generally, functional requirements describe system behavior under specific conditions.

4.3.1 System Requirements

1. The system shall allow users to check the full description of the item of interest by simply clicking on the picture of the item.
2. The system shall provide full shipping services after checkout for both the retailer and the customer by providing shipping information.
3. The syntax of the UI will be simple enough for everyone to find it easy and convenient to use.
4. The system shall record users' information and keep them on the cloud, so that customers can easily retrieve their data and access them from any device.

4.3.2 User Requirements

1. Upon creating an account, users shall be able to specify whether they're a renter or retailer.
2. Users shall be able to access the application from any device.
3. Users shall be provided with a great product management service.
4. Products will be classified into categories, and users will be able to search using categories' names.
5. Users shall be provided with multiple payment options (PayPal, venmo…)
6. Users shall be provided with a shipping service.
7. Users shall be provided with a rating system, where renters will be able to rate the retailer's customer service.
8. Users shall be recommended a list of products based on their previous searches / uses of the application.
9. The system must provide the users with a service that allows them to customize the app. For instance, providing a dark mode or night lightning option.

4.4. Non-Functional Requirements

They are requirements that are not related to the system functionality, rather define *how* the system should perform.

4.4.1 Product Requirements

These are requirements that describe the software's *behavior* and *constraints*.

1. The system must contain an *authentication process* that guarantees users will be able to access the system's services only after confirming the username and password.
2. The application should be user-friendly, where all users find it easy and convenient to use, regardless of their age or abilities.
3. The system must provide users with a *Feedback* service that allows the user to input any feedback. This will help the developers improve the application.
4. The system must be designed with customer satisfaction in mind where the user can achieve high levels of productivity without the struggle of learning the application. This will result in *great customer satisfaction*.
5. The system must be refactored constantly until the most optimal code with low memory usage and no memory leaks is found; As such, the system should require at most 500mbs-1gb of ram for use when running.
6. The system must be hosted by a great web server that will allow for over *100,000* user visits while maintaining *optimal performance*.
7. The system must perform *without failure* for at least *90% of use cases* during the month.
8. The time to *restore* the system after a *failure* (such as a server overload) must not be greater than *30 minutes*.
9. The *rate of errors* the users experience after submitting their payment info on the payment information page must not *exceed 15%.*
10. The *payment gateway* must be *SSL and TLS PCI compliant.*
11. The system must be *compatible* with *Windows 8 and up.*

4.4.2 Organizational Requirements

1. Environmental requirements: the software is a web app, whose user interface is convenient and easy to use, with colors that are comfortable to the eye, and an easy-to-read font size. The system shall also provide a dark mode or night lighting as an option for the comfort of users.
2. Operational requirements: the system will be used by people who are interested in renting a product or offering a product for rent, and they must be able to access the system on their mobile devices (Android & iOS). Users must also authenticate themselves using their usernames and passwords.
3. Development requirements: the system will be developed using mostly Java Script, as it is a very dynamic and versatile programming language, that allows for a richer interface and increased interactivity. But we will also need to use other languages like HTML and CSS, in order to produce a high-quality end product.

4.4.3 External Requirements

They cover all requirements that're derived from factors external to the software system and the development process.

1. *Regulatory Requirements:* The system shall implement many *privacy provisions* that will serve to protect the user's confidentiality and how the user accesses the programs within the system.
2. *Ethical Requirements:* The system shall be developed under the fundamental rules of software engineering that concern *ethics* and *Intellectual Property* rights.
3. *Legislative Requirements:* The software shall be licensed by the Palestinian Monetary Authority as to establish the rights for the end-users.
4. *Accounting Requirements:* The system shall keep track of the expenses and constantly report changes as to ensure that the money being received and spent is sent to the proper destinations or managed by the correct parties.

4.5. Customer Requirements

These are specifications of *features* or *services* that the customers find *necessary* in the software.

1. The customer must be able to create an account.
2. The customer must be able to confirm the account after creating it.
3. The customer must be able to login securely after confirming the account.
4. The customer must be able to search for a product comfortably using categories.
5. The customer must be able to find all product information quickly after clicking on the image of the product or name.
6. The customer must be able to access the billing methods after clicking the "checkout" button.
7. The customer must be able to choose the payment method and input all billing info after selecting the payment method.
8. The customer must be able to choose whether they want to receive the product through shipping or not.

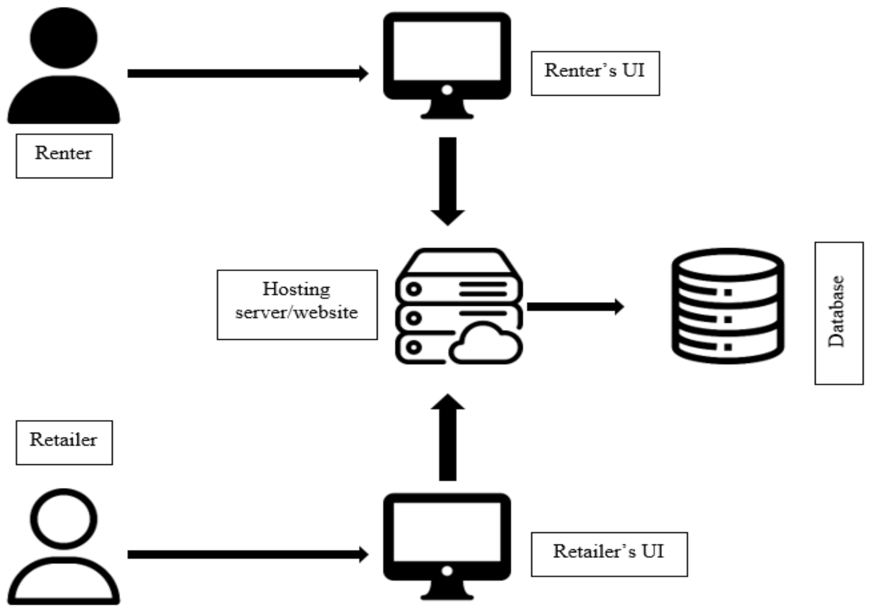
4.6. Requirements Specifications

Goods Rental Software/recommending products/4.3.2.4	
Function	Searching for items.
Description	The software must take the users' inputted keywords and scan the database for the best matches.
Inputs	A product name or keywords related to it that the user input in the search text field.
Source	System database.
Output	A list of products that best fit the entered keywords.
Destination	
Action	When a user enters the name of a product they're interested in buying, or any keywords that describe / related to it, the system takes that input and scan the system's database to find the items that best match the inputted data.
Requires	User's input in the search text field.
Pre-condition	The system database must contain all products' information to display suitable matches to the user's search.
Post-condition	The system database must constantly be updated to add better and more suitable description of products.
Sides effects	None.

Goods Rental Software/recommending products/4.3.2.7	
Function	Providing a Rating system
Description	Ask the users about their opinion on the retailer's customer service and provide them with numeric values that they can select.
Inputs	The numeric value that the customer choose and any comments they left behind.
Source	Renters experiences.
Output	Giving each retailer a rating based on the renters' inputs.
Destination	System database.
Action	After a user completes a purchase process, a window will pop up and ask them to rate their experience with retailer out of a numerical value. And add any comments or feedback they have. Then the system will calculates the average rating of the retailer, and display the result next to their names.
Requires	Completing a purchase process.
Pre-condition	The system database must contain all retailers' information alongside their previous rating,
Post-condition	Actively updating retailers' rating.
Sides effects	None.

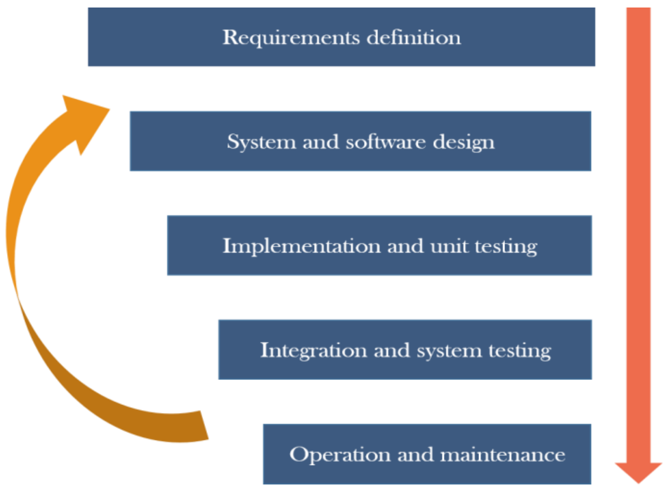
Goods Rental Software/recommending products/4.3.2.8	
Function	Recommending items to users.
Description	Check the user’s search history and based on the result recommend a list of items that seem to fit the customer’s taste.
Inputs	All products that the user searched for, all categories they were interested in browsing, and all products they purchased.
Source	Search history / memory.
Output	A list of products that the user might be interested in buying.
Destination	System database.
Action	When the customers first start using the app, their recommended items will be general, and the current popular items on the application. But after a while of using the software, it will become more customized, as the system gets to know the user’s preferences, and become able to recommend new items based on these preferences.
Requires	A user search history so that a list of products can be recommended.
Pre-condition	The system database must contain all users’ information alongside their searches and browsing history.
Post-condition	Recommend list of items must get updated constantly to accommodate users’ changing preferences.
Sides effects	None.

5. System Architecture

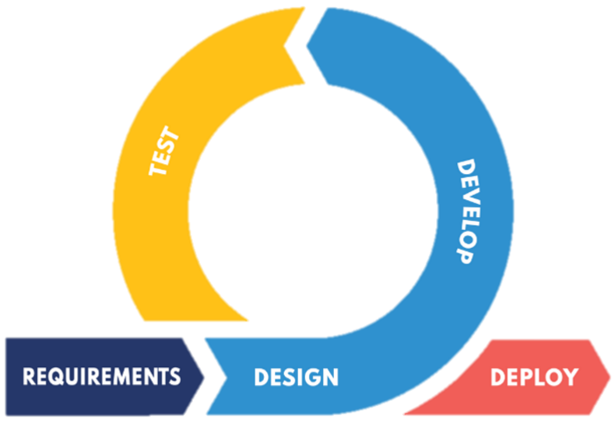


6. System Models

We are going to use the Waterfall model in a plan driven process for the majority of the development process, as we have found it simpler and helps us work towards a meticulously thought-out plan with mindfulness and perfection rather than encounter many problems out of the blue and be expected to fix them within the given timeframe.


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

However, since encountering changes in system and user requirements is inevitable during the process, we will be using agile development methods in some parts of the development, during the UI design process for instance. Agile methods are based on creating multiple versions of a project called increments. Where each increment is reviewed and worked on alongside stakeholders or end-users to deliver a more proficient version, until the project is fully completed. Dividing the project this way allows engineers to easily adapt to changes. Besides, the continuous customers' involvement guarantees great customer satisfaction and high-quality end products.



7. System Evolution

If GRS is successful in achieving a great spot in the digital ecommerce market; the software will be enhanced. Enhancements could include more UI options, such as more themes to the main page. Or implementing a messaging system, where renters and retailers can contact each other directly through the software.

If GRS only sees moderate success in the ecommerce digital market, no new features will be added.

Bugs will be fixed throughout the software's life cycle.

8. Appendices

Appendix A

Abbreviations & Acronyms:

CVV: Card Verification Value

CSS: Cascading Style Sheets

GRS: Goods Renting Software

HTML: Hypertext Markup Language

ISO: International Organization for Standardization

PCI: Peripheral Component Interconnect

PHP: Hypertext Pre-processor

SSL: Secure Sockets Layer

SQL: Structured Query Language

TLS: Transport Layer Security

UI: User interface

Appendix B

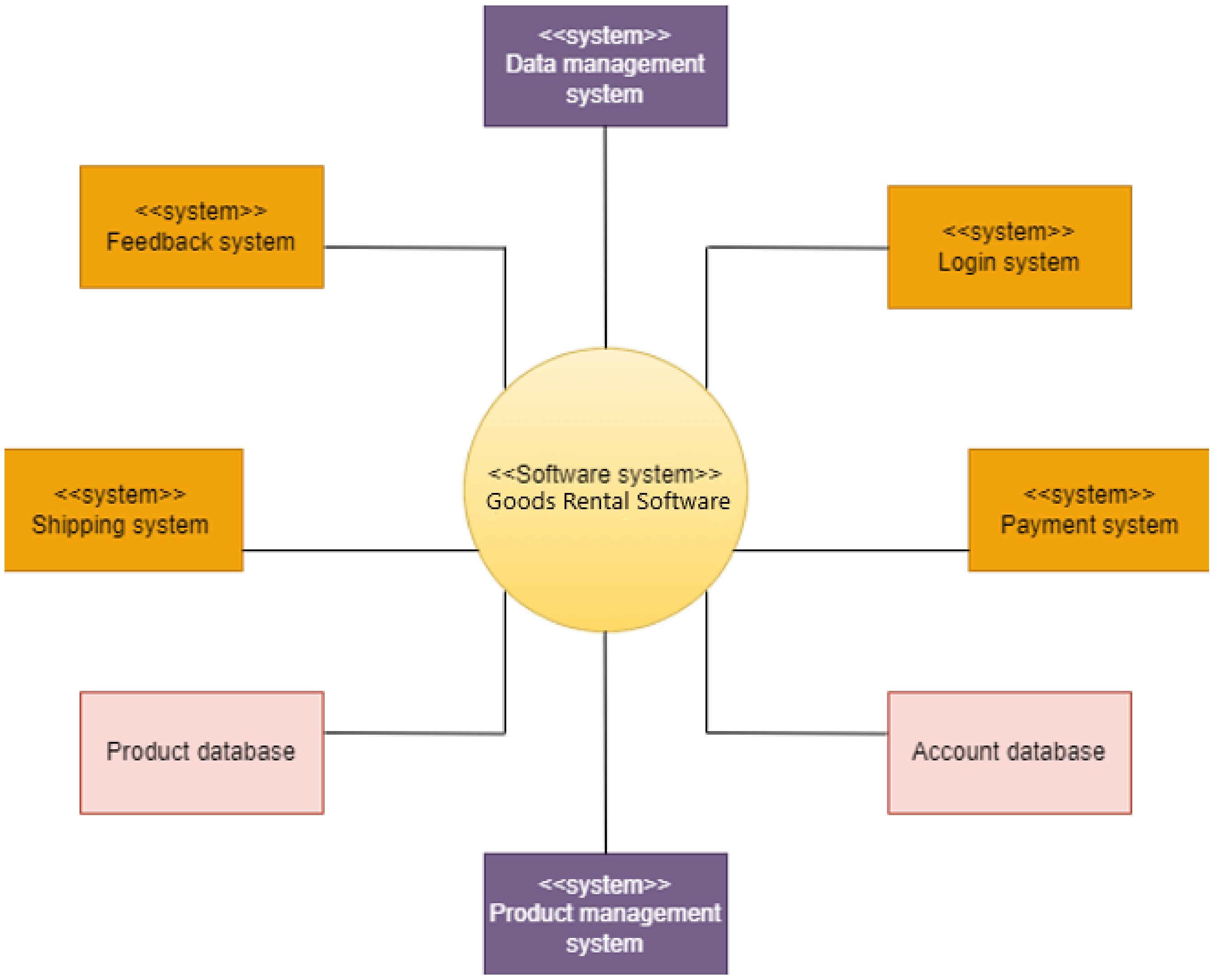
Descriptions:

PHP and MySQL will be used to create a database for the storing and verification of user sign-in, sign-up information. The database will also be used to store data of the items on the server and update item data.

The computer device will be used to program and design the application as well as send progress to other working teams.

9. Modeling

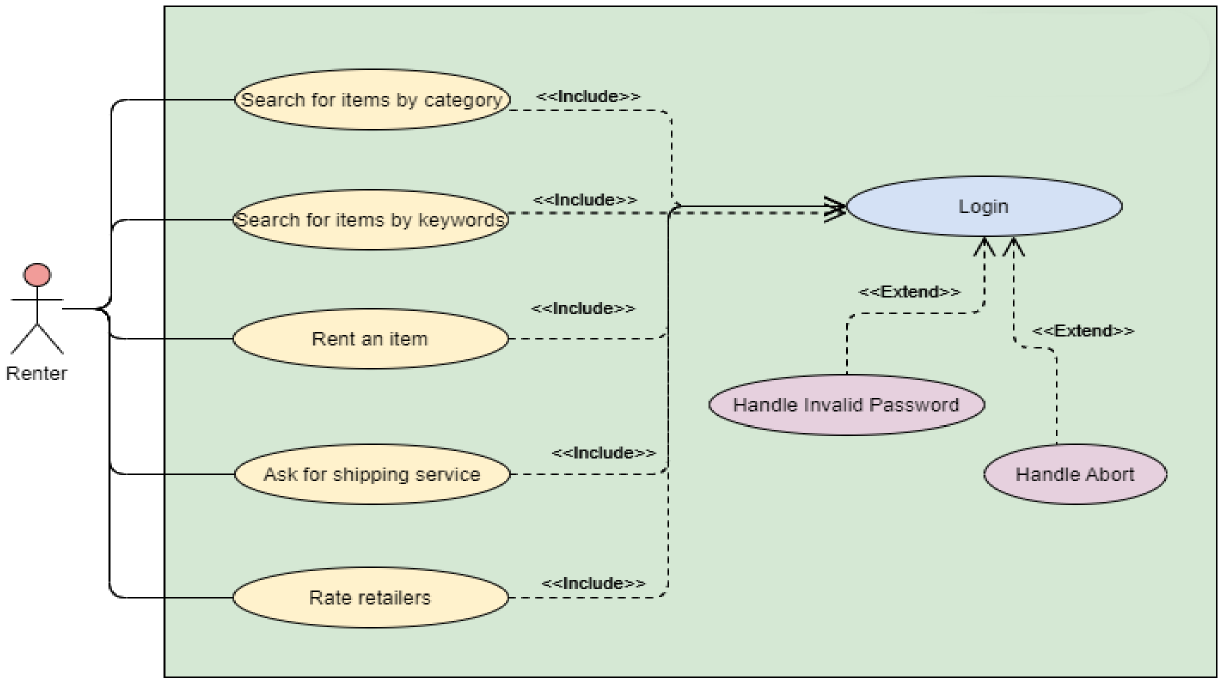
9.1. Context Model



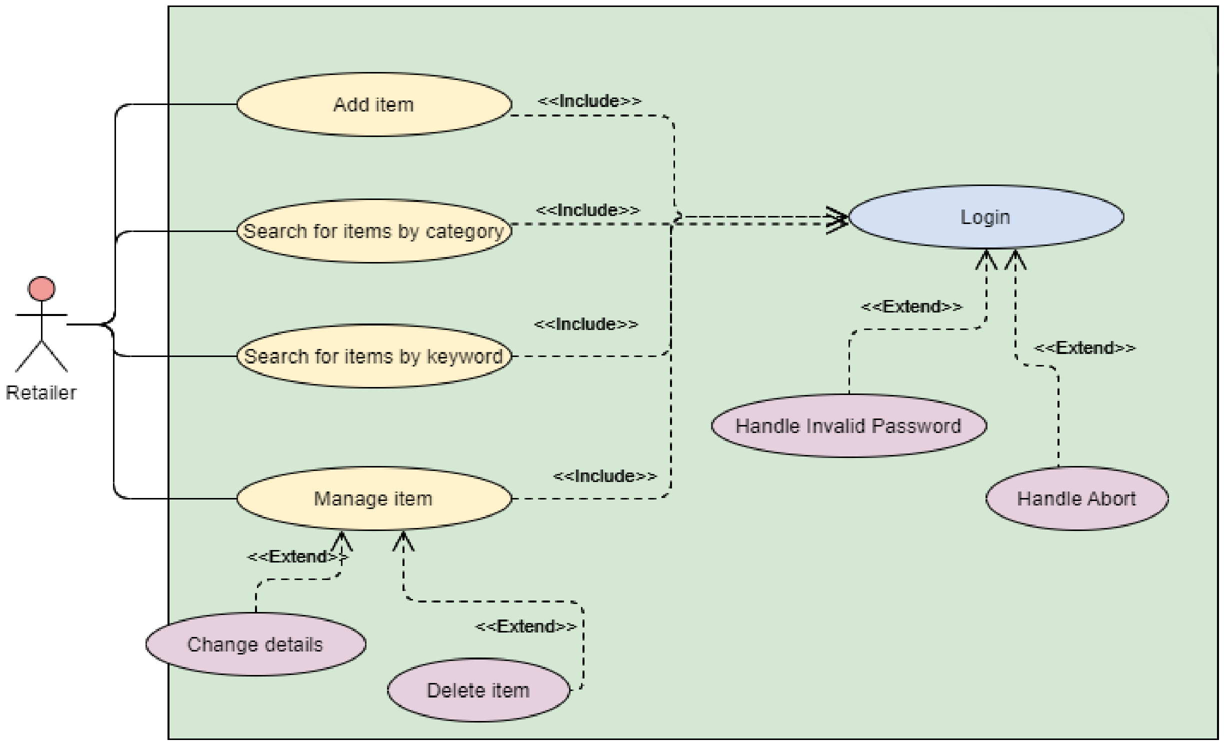
9.2. Interaction Models

9.2.1. Use Cases

1. Use case for Renters:

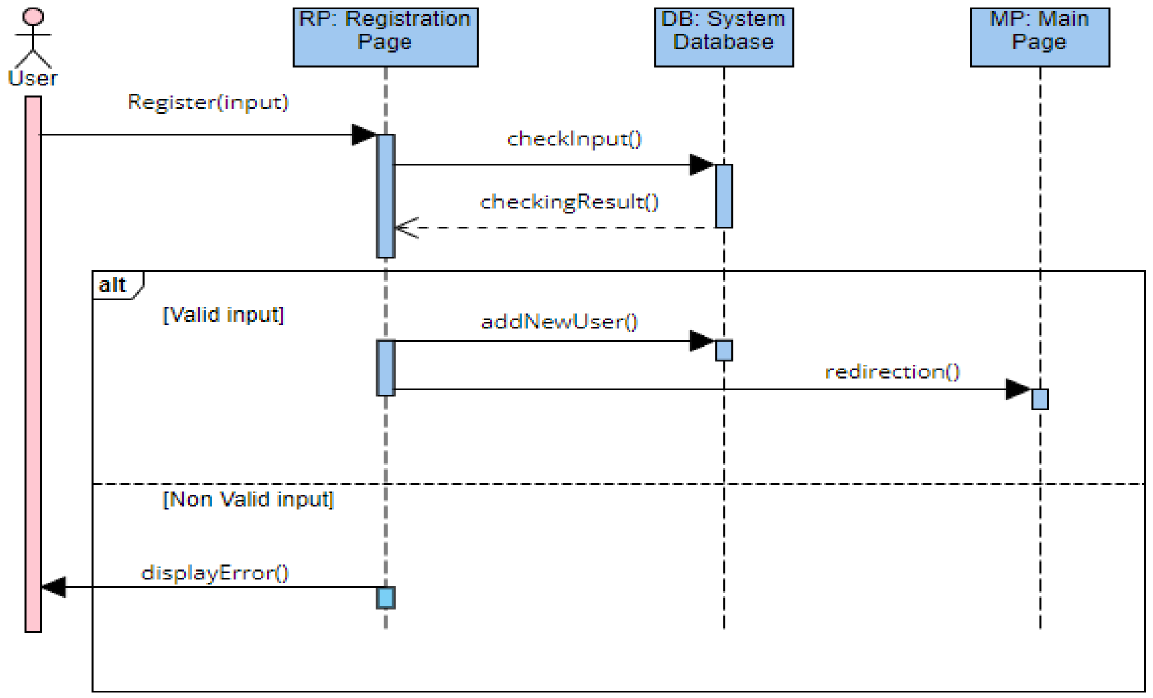


2. Use case for Retailers:

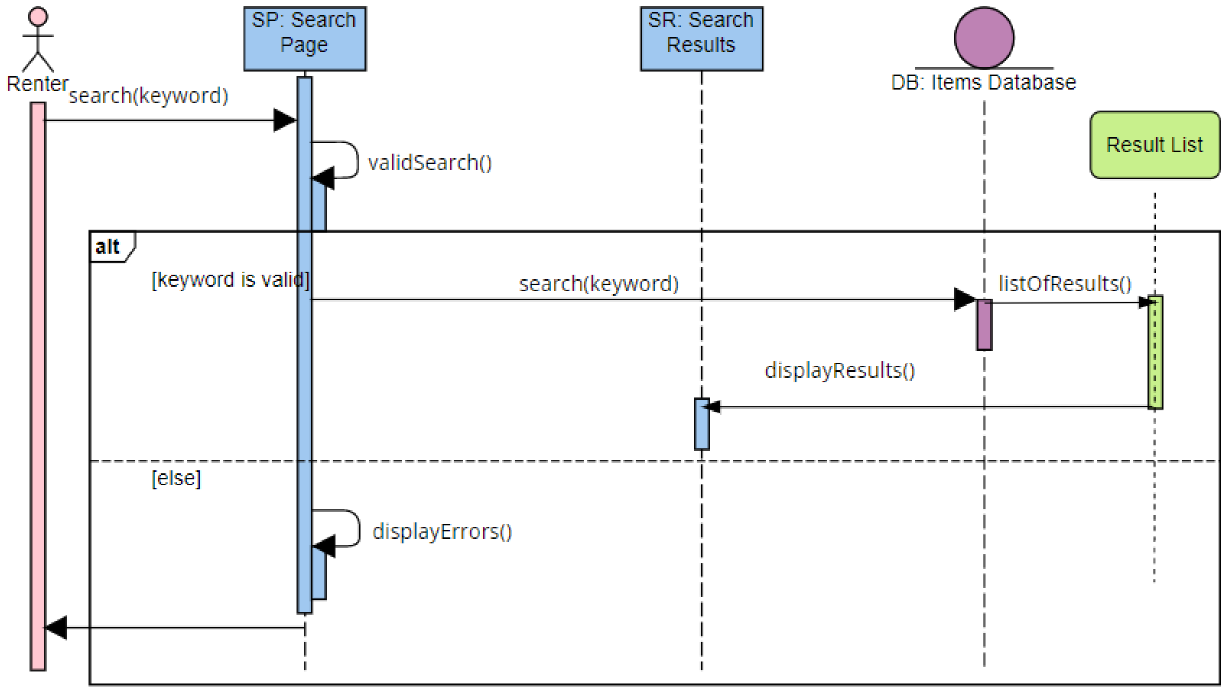


9.2.2. Sequence Diagrams

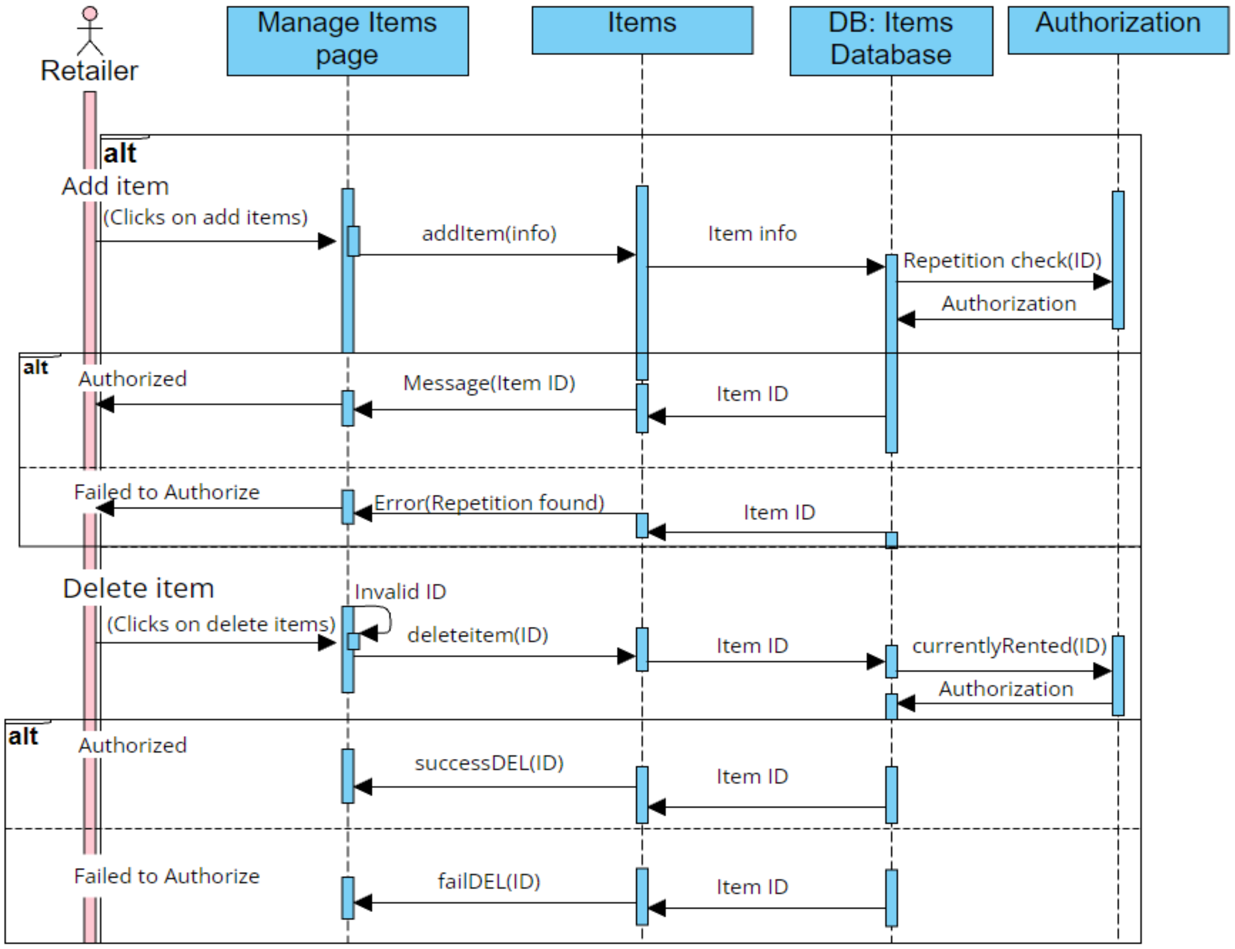
1. Sequence Diagram for Registration:



2. Sequence Diagram for Search process:



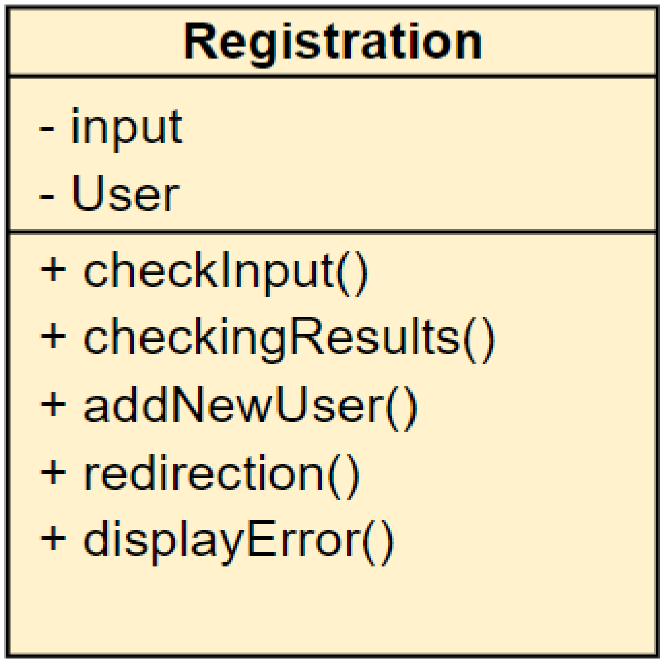
3. Sequence Diagram for Managing Items:



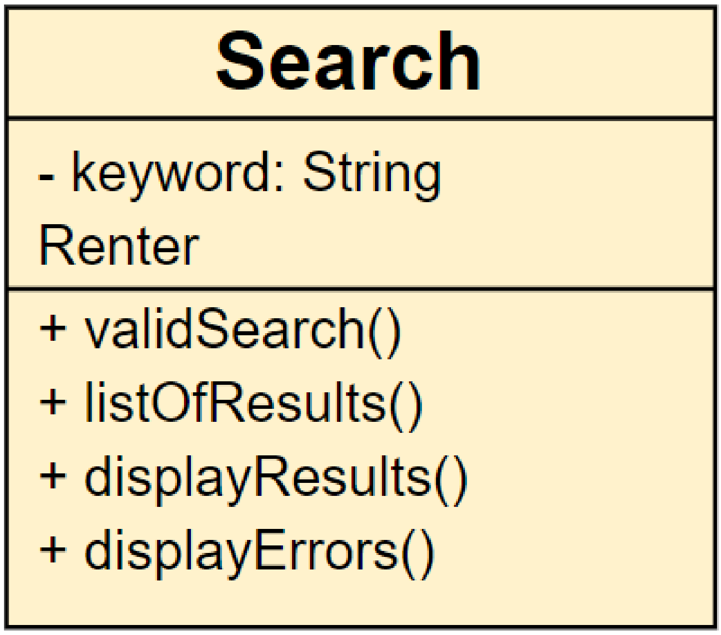
9.3. Structural Models

9.3.1. Class Diagrams

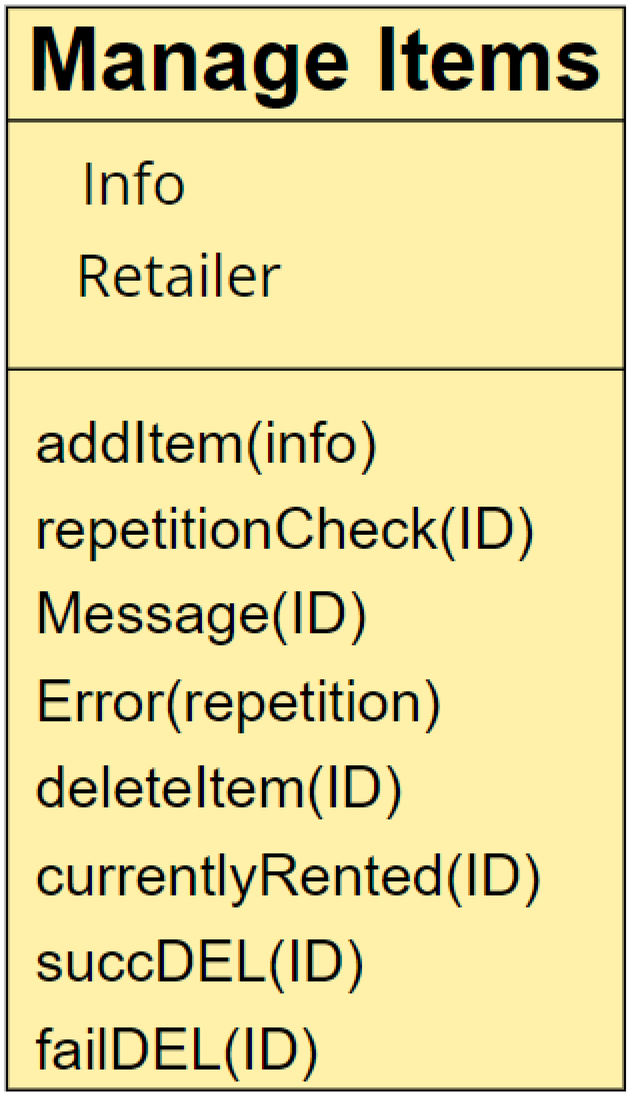
1. Class Diagram for Registration:



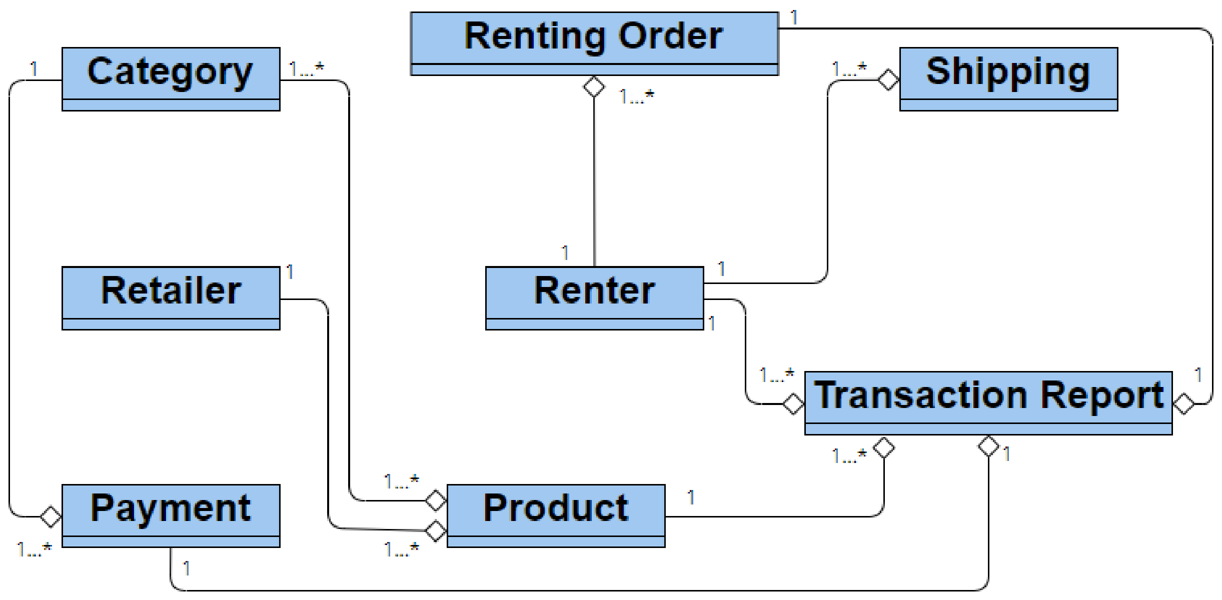
2. Class Diagram for Search process:



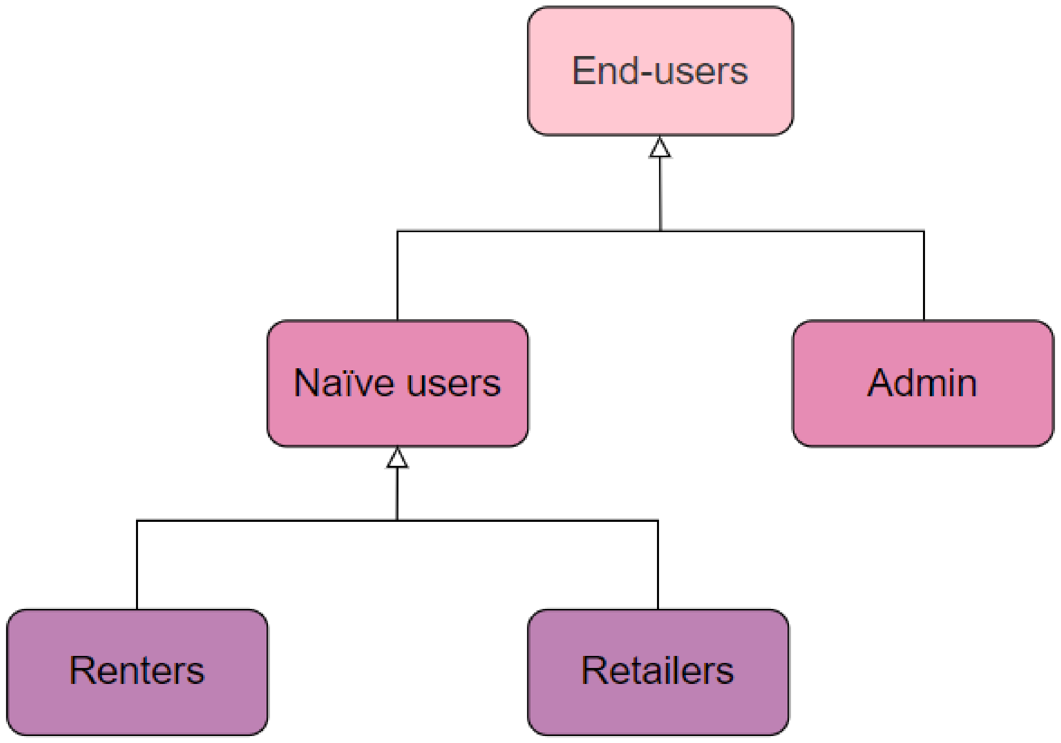
3. Class Diagram for Managing Items:



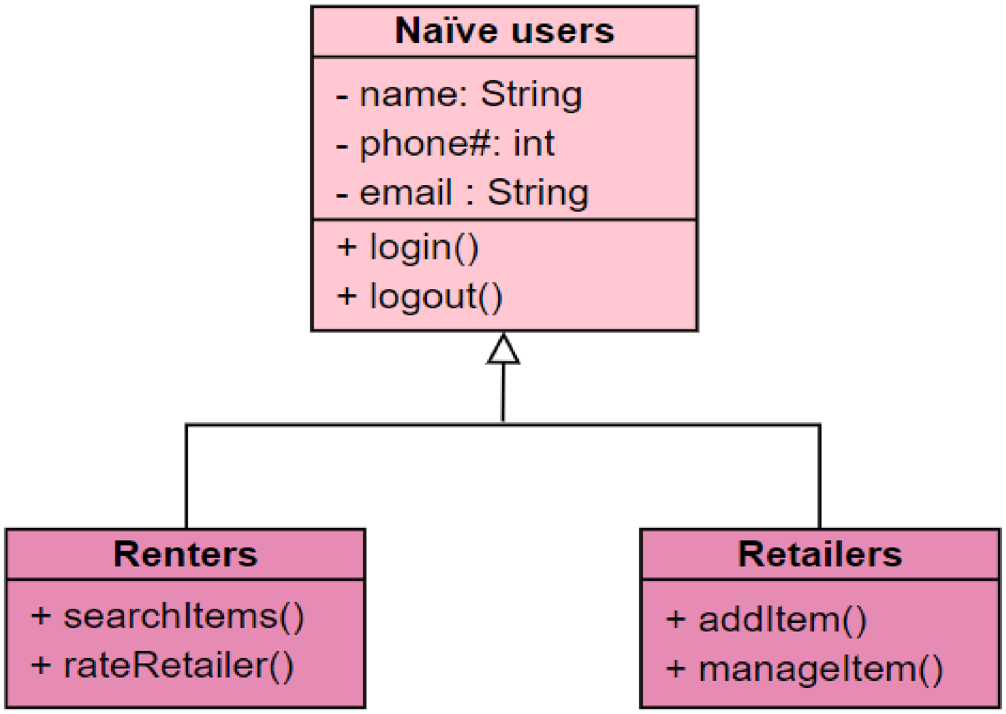
9.3.2. Class Association



9.3.3. Generalization

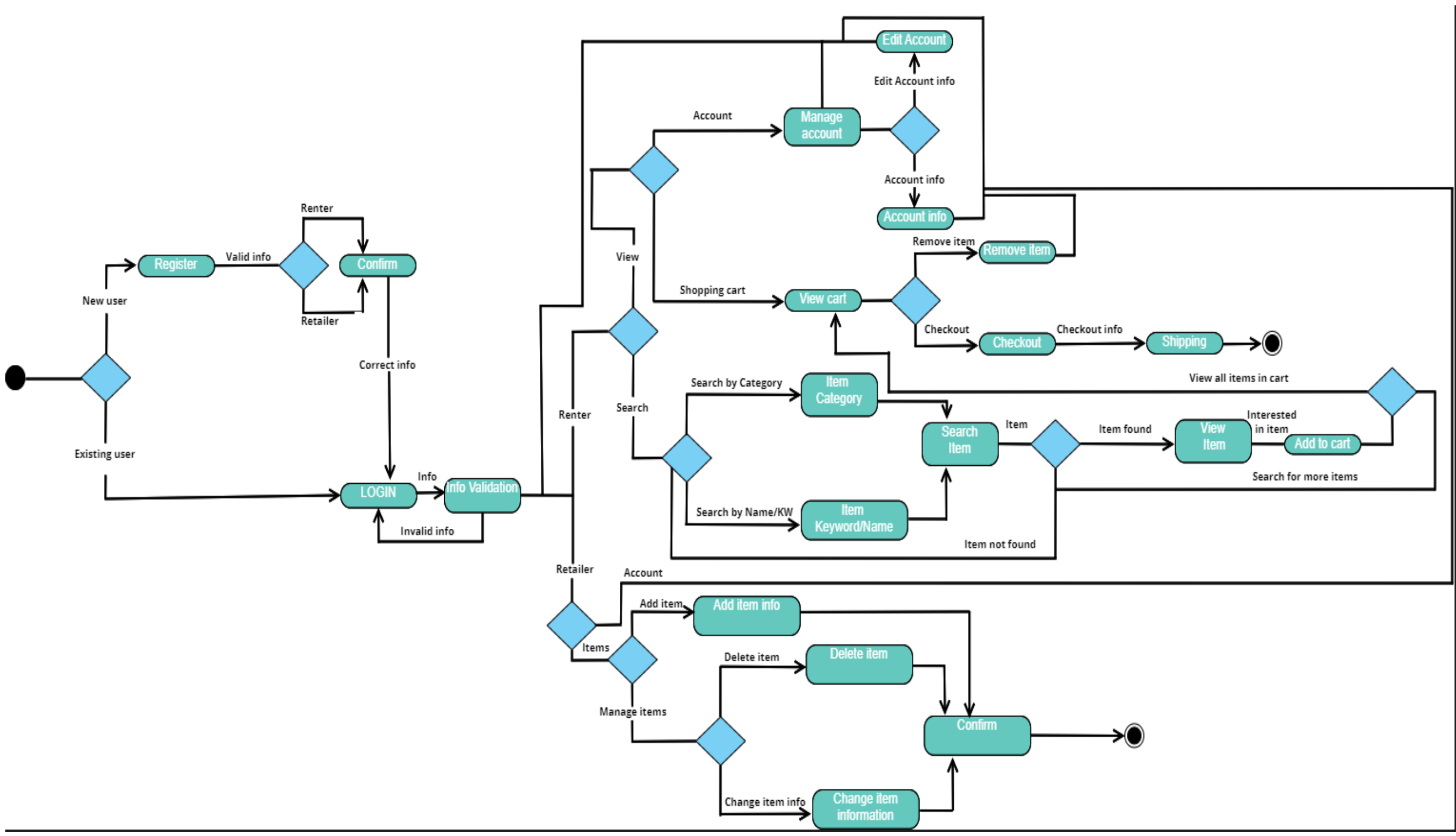


Generalization with more details:



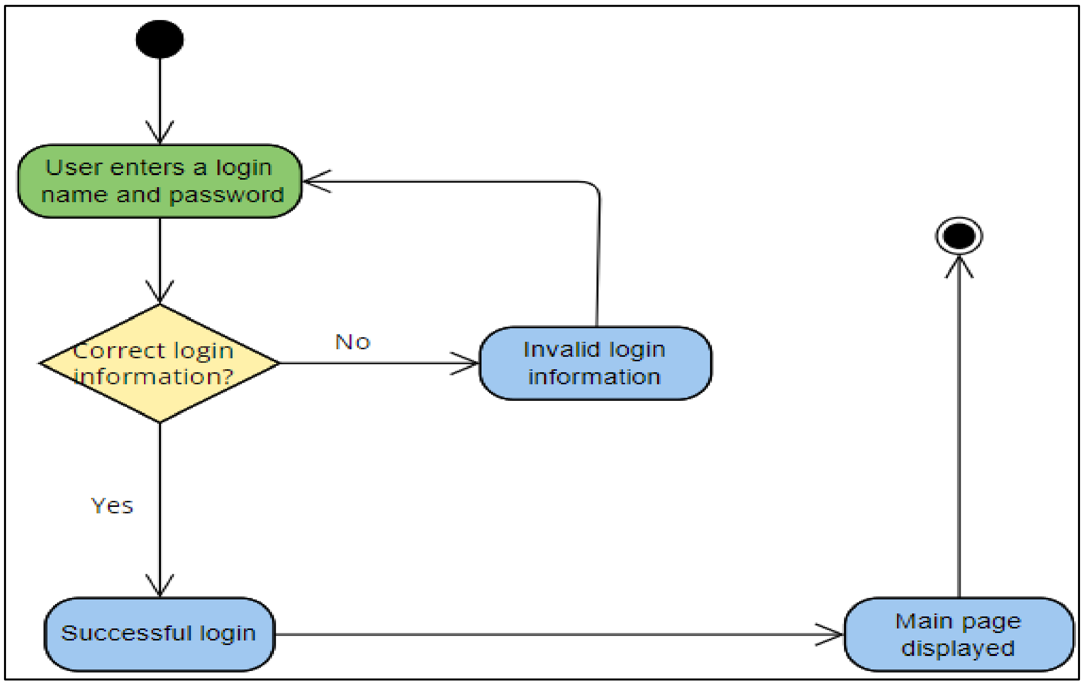
9.4. Behavioral Models

9.4.1. General Behavioral Model



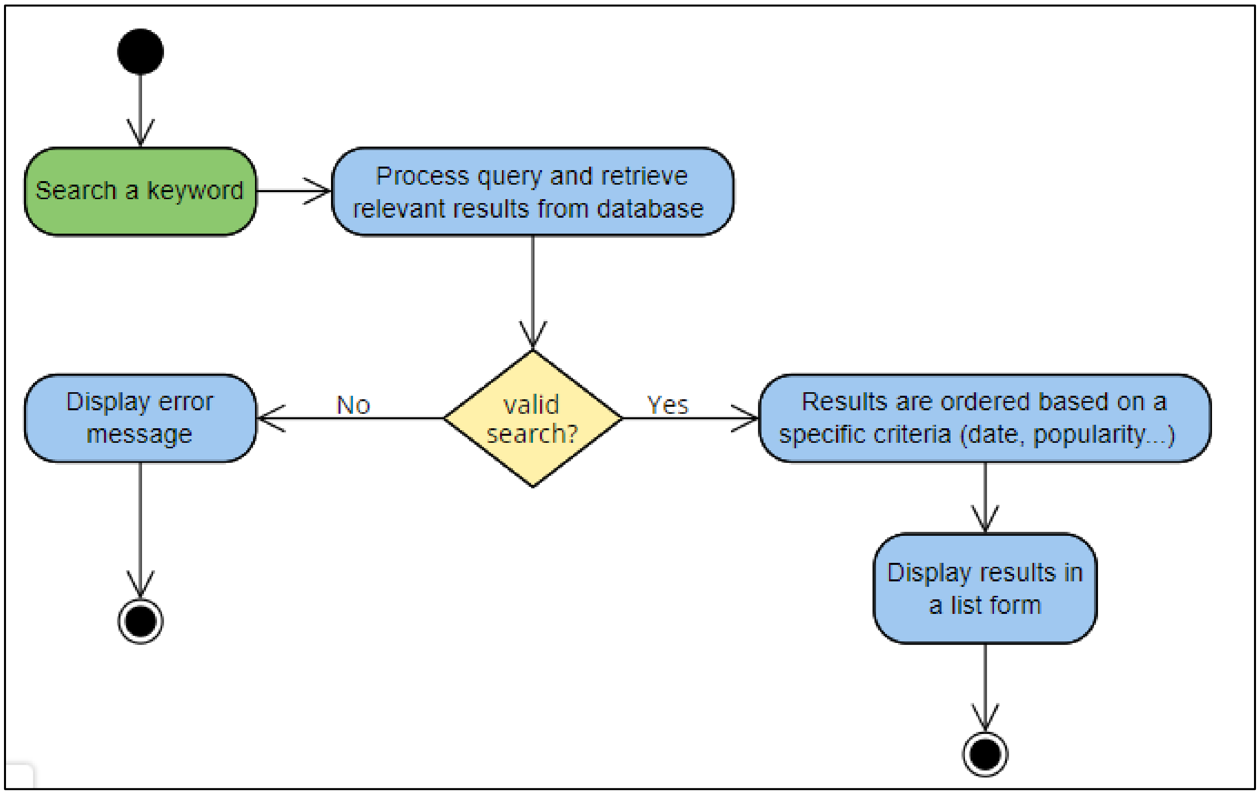
9.4.2. Activity Models

1. Login Activity model:



User enters a login name and password
Correct login information?
No
Invalid login information
Yes
Successful login
Main page displayed

2. Search process Activity model:



Search a keyword
Process query and retrieve relevant results from database
valid search?
No
Yes
Display error message
Results are ordered based on a specific criteria (date, popularity...)
Display results in a list form

10. Index