Department of Computer Science and Mathematics

Software Engineering - Spring 2022

Samer Saber, Jerome Kabbani and Charbel Hayek
Supervised by: Dr. Danielle Azar

April 13, 2022

Contents

1 Preface **3**

2 Introduction **3**
2.1 The Team . . . 3
2.2 The Client . . . 3
2.3 The Project . . . 3
2.4 Programming Environment . . . 4

3 Glossary **4**

4 Project Management **4**
4.1 Feasibility Study . . . 4
4.2 Gantt Chart . . . 5
4.3 Risk Management . . . 6
4.4 Resource List and Cost Estimate . . . 7

5 Software Requirements **7**
5.1 Functional User Requirements . . . 7
5.2 Non-Functional Requirements . . . 13
5.3 System Requirements . . . 14

6 Scenarios **23**

7 System Models **29**
7.1 Sequence Diagrams . . . 29
7.2 Data-flow Diagrams . . . 38
7.3 Use Cases . . . 42
7.4 ER Diagrams . . . 55
7.5 Stimulus Response Model . . . 55

8 System Architecture **55**
8.1 Subsystems . . . 55
8.2 Components . . . 55

9 Client's Business **55**
9.1 Before . . . 55
9.2 After . . . 55

10 Signatures **55**

11 Index **55**

1 Preface

2 Introduction

2.1 The Team

- **Charbel Hayek** (Team Leader, Full-Stack Developer)
  ID: 201904984
  email: charbel.elhayek@lau.edu
- **Samer Saber** (Full-Stack Developer)
  ID: 201401460
  email: samer.saber@lau.edu
- **Jerome Kabbani** (full-Stack Developer)
  ID: 202004409
  email: jerome.kabbani@lau.edu

2.2 The Client

Business Owner	Jad Michael
Business Name	Candia Cameo
Contact Number	(054) 545-4060
E-mail	admin@candiacameo.com
Address	Street Address: Dubai City: Dubai State / Province: Dubai Postal / Zip Code: 999999
Type of Business	Others, please specify below.
Others	Online Jewelry

2.3 The Project

Candia Cameo is a company that produces Exquisite Moissanite Jewelry in the UAE, and sells them online through social media platforms such as Instagram and Facebook. It also has an up and running website where customers can make their payments online.
To increase their customer reach and provide their clients with the best online jewelry shopping experience, they are requesting another E-commerce web application that meets all the requirements they want for their business.
So, as part of our Software Engineering project, we decided to contact the company and convince them to give us the job.
Since, all 3 of us in the team have taken the database course, we think this project is a perfect fit for us to apply what we've learned and further expand our experience in Database Management Systems.

2.4 Programming Environment

- **Operating Systems**
  Microsoft Windows 10

- **Platforms**
  Visual Studio Code, EditorX

- **Programming Languages**
  HTML, CSS, JavaScript, PhP, SqL

3 Glossary

Candia Cameo: The name of the company that will sell Moissanite Jewelry through the website.
Moissanite: Moissanite is naturally occurring silicon carbide and its various crystalline polymorphs. It has the chemical formula SiC and is a rare mineral, discovered by the French chemist Henri Moissan in 1893. The type of jewelry the website will sell.
Customer: a person who want to buy goods from the website.
Coupon: a voucher with a code, entitling the holder to a discount off any particular jewelry from the website.
EditorX: Editor X is a revolutionary web creation platform made exclusively for designers and agencies.
User: Anyone who accesses the website through the any browser.

4 Project Management

4.1 Feasibility Study

- **Is the System worthwhile?**

  The company claims that their current website is barely attracting attention, and that most of their sales are coming from Instagram users.
  As a Software Engineering team, we believe that the reason for the weak traffic on their website is the lack of an attractive and professional web design, as well as the lack of functionalities that provide the customers with the best online jewelry shopping experience possible.
  Also, for their jewelry company to be taken seriously, they must have a professional web application system that can gain the trust of frequent jewelry buyers, and can compete with many other jewelry businesses available online.
  Therefore, the implementation of an E-commerce web application system that meets a professional and attractive design, and their full business requirements, is worthwhile and a must.

- **Does the system contribute to the objectives of the company?**

The company's objectives are to increase their reach and their sales, to facilitate online purchases, and to provide the customer with a safe and professional jewelry shopping experience that meets the level of other well known jewelry businesses.
Therefore, developing a system with an elegant design and functionalities that meet the standards of a professional online jewelry store, will most definitely contribute to the objectives of the company.

- **Can the system be engineered using current technology and within budget?**

  Since the system is an e-commerce web application, it can most definitely be engineered using 'EditorX', which is a revolutionary web creation platform with approachable CSS and sophisticated web infrastructure made exclusively for designers to produce exceptional digital experiences for any screen imaginable, and to create and store heavy or dynamic content behind the scenes of a site using database collections. We can connect the database to individual site elements like images, and responsive display tools like Repeaters.
  As for the budget, the company has already paid the subscription to start working on it.

- **Can the system be integrated with other systems that are used?**

  Engineering a website on 'EditorX' will make it automatically compatible across different browsers.
  Therefore, the system can be integrated with other systems that are used to browse online websites.

4.2 Gantt Chart

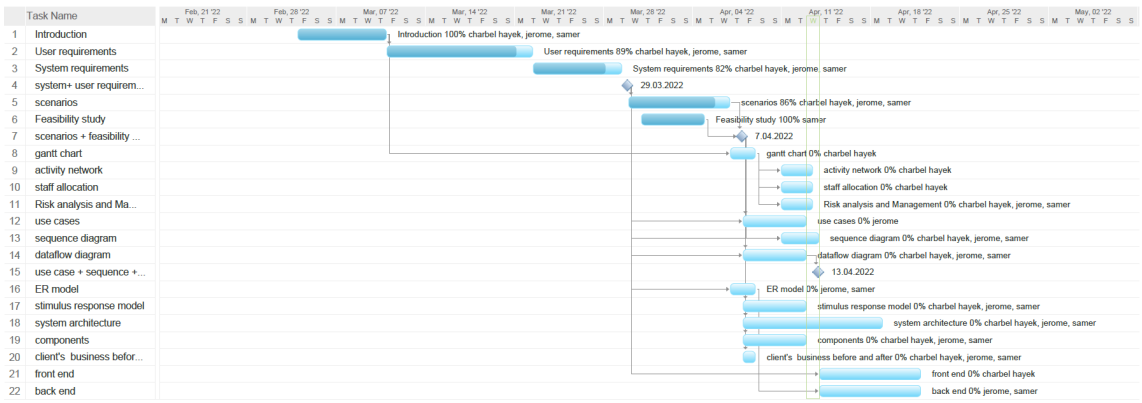


Figure 1: Gantt Chart

4.3 Risk Management

Possible Software Risks					
Risks	**Affects**	**Description**	**Probability**	**Effects**	**Type**
Team member falling ill	Project, Product	Staff might fall ill at critical times, and this will result in their part of the project being assigned to another team member. But this will increase the workload on him, and it might compromise on the quality of the product or the deadline.	Moderate	Tolerable	People
Team member dropping the course	Project, Product	A member of the team may drop out of the course, and this will result in their part of the project being assigned to another team member. But this will increase the workload on him, and it might compromise on the quality of the product or the deadline.	Low	Serious	People
Time management problems with other courses	Project, Product	Some team members taking 12 or more credits, might find it difficult to manage their time, and therefore miss the deadlines.	Moderate	Serious	Estimation
Candia Cameo going bankrupt	Business	The company might go bankrupt, and decide to drop the project without any further contact. This will result in us not being able to keep extracting requirements from the client	Low	Serious	Organization
Disagreements between team members	Project, Product	Team members might come to disagreements when it comes to software design and implementation, which leads to conflict and wasting time.	Moderate	Tolerable	People
EditorX shutting down	Product	The EditorX tool that we're going to use to develop the website, may shut down and no longer be available. This will result in team members not being able to implement the system.	Very Low	Catastrophic	Tools

Contingency Plan	
Risks	**Contingencey Plan**
Team member falling ill	The tasks that the ill team member was supposed to complete, will be divided on all other team members to complete.
Team member dropping the course	The tasks that the team member who dropped the course was supposed to complete, will be divided on all other team members to complete.
Time management problems with other courses	Increase the number of meetings between team members to ensure the tasks are being completed on time.
Candia Cameo going bankrupt	Team members will act as the client to extract requirements.
Disagreements between team members	Team leader should have the final word.
EditorX shutting down	Team members should resort to writing code from scratch.

4.4 Resource List and Cost Estimate

- Draw.io
- MySql Workbench
- Projectmanager.com
- LaTex
- Google docs

Cost estimates: 0$

5 Software Requirements

5.1 Functional User Requirements

5.1.1 Displaying Jewelry Models

The system shall display the company's jewelry worn by models.

Rationale: A display of models wearing the company's jewelry will give the customer a better idea of how it will look on them.

Specification: Software Requirements/System Requirements/5.3.1

Source: Charbel Hayek

5.1.2 Displaying Company's Summary

The system shall display the company's Summary on the home page.

Rationale: A display of a summary will provide the user with a brief comprehensive understanding of what the company does.

Specification: Software Requirements/System Requirements/5.3.2

Source: Charbel Hayek

5.1.3 Displaying Featured Products

The system shall display the company's featured products on the home page.

Rationale: A display of featured products on the home page will attract the user to look further more into other products.

Specification: Software Requirements/System Requirements/5.3.3

Source: Charbel Hayek

5.1.4 Displaying Categories

The system shall display product categories on the home page.

Rationale: A display of product categories on the home page will provide the user with an enhanced view of different jewelry categories available on the system.

Specification: Software Requirements/System Requirements/5.3.4

Source: Charbel Hayek

5.1.5 Displaying Services

The system shall display the company's services on the home page.

Rationale: A display of the company's services will provide the user with an enhanced view of different services provided by the system.

Specification: Software Requirements/System Requirements/5.3.5

Source: Charbel Hayek

5.1.6 Displaying Moissanite Education

The system shall display a moissanite education on the homepage where the user can click on it and get educated on moissanite stones.

Rationale: Educating customers about moissanite will increase their interest in moissanite jewelry.

Specification: Software Requirements/System Requirements/5.3.6

Source: Charbel Hayek

5.1.7 Searching For Products

Users shall be able to search for any product available on the system.

Rationale: A search field where users can type-in keywords in order to search for a specific product on the system is easier and faster.

Specification: Software Requirements/System Requirements/5.3.7

Source: Charbek Hayek

5.1.8 Navigating Through a Section Bar

The system shall contain a navigation bar composed of different product categories.

Rationale: Organizing the system into sections will be much easier for the users to navigate through.

Specification: Software Requirements/System Requirements/5.3.8

Source: Charbel Hayek

5.1.9 Creating an Account

The user shall be able to sign up and set up a profile.

Rationale: Setting up a profile on the system will allow the users to save their favorite products and track their order history.

Specification: Software Requirements/System Requirements/5.3.9

Source: Charbel Hayek

5.1.10 Signing In

The user shall be able to sign in to their existing profile.

Rationale: Signing in to an account will enable a user to access and edit their profile.

Specification: Software Requirements/System Requirements/5.3.10

Source: Charbel Hayek

5.1.11 Booking An Appointment

The user shall be able to book an appointment for consultation.

Rationale: Providing customers with the option to book an appointment for consultation, will help them make a best choice of purchase.

Specification: Software Requirements/System Requirements/5.3.11

Source: Jerome Kabbani

5.1.12 Contacting Us

The user shall be able to contact the company through the system.

Rationale: Providing customers with the ability to contact the company will help them voice their questions and concerns .

Specification: Software Requirements/System Requirements/5.3.12

Source: Jerome Kabbani

5.1.13 Purchasing Products

Any user(registered or not) shall be able to purchase one or several jewelry items.

Rationale: Any user who wants to purchase a product or several must be able to do so through the system.

Specification: Software Requirements/System Requirements/5.3.13

Source: Jerome Kabbani

5.1.14 Adding Products For Purchase

The system shall contain a shopping bag section where products selected by the user for purchase, are stored.

Rationale: Storing selected products in one place will provide the user with a consolidated view of the selected products that they would like to purchase.

Specification: Software Requirements/System Requirements/5.3.14

Source: Jerome Kabbani

5.1.15 Removing Selected Products

The user shall be able to remove previously selected products from the shopping bag.

Rationale: In case the user does not want to purchase a selected product in the shopping bag anymore, they must have the option to remove it.

Specification: Software Requirements/System Requirements/5.3.15

Source: Jerome Kabbani

5.1.16 Paying through Debit/Credit

Users shall make the payment through credit/debit card only.

Rationale: It is safer for the company to receive payments through credit/debit cards only.

Specification: Software Requirements/System Requirements/5.3.16

Source: Jerome Kabbani

5.1.17 Inputting Coupon code

The user shall be able to submit a coupon code upon checkout for discounts.

Rationale: Coupons attract more customers to make more purchases from the company through the system.

Specification: Software Requirements/System Requirements/5.3.17

Source: Jerome Kabbani

5.1.18 Summarizing order

The system shall display a summary of the order before checkout.

Rationale: Displaying a summary of the order before checkout will provide clarity for the customer before committing to any purchase.

Specification: Software Requirements/System Requirements/5.3.18

Source: Jerome Kabbani

5.1.19 Suggesting Products

The system shall display a set of other products as a suggestion before checkout.

Rationale: A display of other products as a suggestion before checkout will increase the likelihood of the user to add more to the bag, hence more sales.

Specification: Software Requirements/System Requirements/5.3.19

Source: Samer Saber

5.1.20 Displaying Package

The system shall display the type of package the product will be packed with before checkout.

Rationale: A display of the package before checkout will provide the customer with a more detailed idea of what the final product will be delivered in.

Specification: Software Requirements/System Requirements/5.3.20

Source: Samer Saber

5.1.21 Providing Assistance Information

The system shall display assistance information before checkout.

Rationale: Displaying information for assistance before checkout will provide the customer with security and reassurance.

Specification: Software Requirements/System Requirements/5.3.21

Source: Samer Saber

5.1.22 Inputting Shipping Address

The user shall be able to input their shipping address for delivery.

Rationale: For the company to be able to ship products to customers they should have their address.

Specification: Software Requirements/System Requirements/5.3.22

Source: Samer Saber

5.1.23 Inputting Billing Address

The user shall be able to input their billing address for payment.

Rationale: For the company to be able to ship products to customers they should receive payment first.

Specification: Software Requirements/System Requirements/5.3.23

Source: Samer Saber

5.1.24 Live Chatting

The user shall be able to start live chatting with customer support with ease.

Rationale: An accessible live chat will give customers a faster way to reach customer support in the exact moment they have questions or problems that they can't solve.

Specification: Software Requirements/System Requirements/5.3.24

Source: Samer Saber

5.2 Non-Functional Requirements

5.2.1 Payment Security

Payment on the system shall be secure.

Rationale: A secure payment on the system will prevent customer complaints, and bad reputation.

Specification: Software Requirements/System Requirements/5.3.25

Source: Jerome Kabbani

5.2.2 User friendly

The system shall be user friendly.

Rationale: A friendly user interface will let customers feel relaxed and shop with ease.

Specification: Software Requirements/System Requirements/5.3.26

Source: Jerome Kabbani

5.2.3 Responsive System

The system shall respond to any screen size.

Rationale: Nowadays, most users are browsing the internet on their mobile phones and tablets, so it has become essential to create websites that are optimized for usage across all platforms.

Specification: Software Requirements/System Requirements/5.3.27

Source: Charbel Hayek

5.2.4 Secured Database

The system's database shall be secured.

Rationale: To prevent hackers from accessing sensitive data, the database should be secure

and prevent any data theft.

Specification: Software Requirements/System Requirements/5.3.28

Source: Samer Saber

5.2.5 System Compatibility

The system shall be compatible on different browsers that are available in the market.

Rationale: If a website does not render perfectly on all browser-device combinations, it will alienate potential users, leading to loss of traffic and potential revenue.

Specification: Software Requirements/System Requirements/5.3.29

Source: Charbel Hayek

5.3 System Requirements

Software Requirements/Functional User Requirements/5.1.1
Function: displaying jewelry models: models **Description:** a simple video/pictures of models wearing the jewelry **Inputs:** picture or video **Source:** online database **Outputs:** a video/photo of the model **Destination:** home page **Action:** a query will be called to get the photo or video in the main table called models **Requires:** the website to be open **Pre-condition:** internet connection + online database up and running **Post-condition:** the info on the screen **Side-effects:** none

Software Requirements/Functional User Requirements/5.1.2
Function: displaying company's summary: summary **Description:** a summary about the company **Inputs:** a text message (paragraph) **Source:** online database **Outputs:** home page **Destination:** home page **Action:** a query will be executed that will fetch the summary about the company from the table in the database. **Requires:** the website to be open **Pre-condition:** internet connection **Post-condition:** the info on the screen **Side-effects:** none

Software Requirements/Functional User Requirements/5.1.3

Function: displaying featured products: featured
Description: a couple of products will be featured
Inputs: product names
Source: online database
Outputs: product info and pictures
Destination: home page
Action: A query shall be executed that searches the table called products and gets the information (rows) or instances of the products that are the most purchased or the most viewed, and then the products shall be displayed on the home page under "featured products", and then a div section called " featured-products" will be issued and then we will insert the products that are extracted from the database into the div and it will be shown under the featured section (h2) header. The featured section is under the summary-model section
Requires: Database up and running + internet connection
Pre-condition: availability of the products
Post-condition: products info on the screen
Side-effects: none

Software Requirements/Functional User Requirements/5.1.4

Function: displaying jewelry categories: jewelry-categories
Description: 4 categories will be displayed
Inputs: names, sub section names
Source: online database
Outputs: categories with subsections
Destination: home page
Action: a query will be executed that will fetch the section and subsections and then it will be displayed as a jewelry category section (div) called "jewelry-category" will be displayed as a div under "featured", it will display sub-sections as pictures with the same sub sections as the nav bar. The subsections are: Rings - Bands - Bracelets - Necklaces - Earrings - Cufflinks – Charms.
Requires: Database up and running + internet connection
Pre-condition: no bugs
Post-condition: categories on the screen
Side-effects: none

Software Requirements/Functional User Requirements/5.1.5

Function: displaying services: The Candia Cameo experience
Description: the company's services will be displayed
Inputs: company's services info
Source: html document
Outputs: services
Destination: home page
Action: a html div called "services" that contains a title (h1 header) called " the-candia-cameo-experience" will be made under the "jewelry-category" section. This section will contain 4 different sections (div) the first one is: complementary shipping and returns, second one candia cameo at your service, the third one book an appointment , the last one : the candia box. All the name of the sub divisions (div) classes have the same name as the services. And each service will contain an a tag that contains a link redirecting the customer to another page that contains more info about the clicked section
Requires: Database up and running + internet connection
Pre-condition: no bugs
Post-condition: displaying services on the screen
Side-effects: none

Software Requirements/Functional User Requirements/5.1.6

Function: displaying moissanite education: education
Description: the moissanite information will be displayed for clarifications
Inputs: information about moissanite
Source: online database
Outputs: information section
Destination: home page
Action: a query will be executed that will fetch the education about the moissanite, and will be displayed as an html div called "education" that will contain all the information about moissanite. This section will be under "jewelry-category" section but above "the-candia-cameo-experience"
Requires: Database up and running + internet connection
Pre-condition: no bugs
Post-condition: moissanite education section
Side-effects: none

Software Requirements/Functional User Requirements/5.1.7

Function: Searching Products: Search
Description: a search icon will be displayed on the website to help users search in a fast way for what they want
Inputs: product name
Source: online database
Outputs: the similar products
Destination: screen
Action: a query shall be executed that searches for the products table and retrieves the info of some of the products that are similar in name, colour, or any identifier that can be written. And the following products shall be displayed on the user's screen.
Requires: Database up and running + internet connection
Pre-condition: user typing something in the search bar
Post-condition: displaying products
Side-effects: none

Software Requirements/Functional User Requirements/5.1.8

Function: displaying navigation bar
Description: a navigation bar will help people who don't know what they want to discover the products
Inputs: title of each category
Source: html document
Outputs: categories and sub categories
Destination: screen
Action: when the user open the website they shall see 4 categories on top of the website as a navigation bar (nav-bar) header , Category 1 (Jewelry) contains Sub-categories (Rings - Bands - Bracelets - Necklaces - Earrings - Cufflinks - Charms), Category 2 (Collections) contains Sub-Categories (winter Collection - Summer Collection - Wedding and Engagement), Category 3 (Bespoke) contains: no sub-category, Category 4 (Collectibles) contains: no sub-category. Each category will take you to the wanted products or collections, this category division is made by using the div in html.
Requires: internet connection
Pre-condition: none
Post-condition: displaying categories
Side-effects: none

Software Requirements/Functional User Requirements/5.1.9

Function: creating an account: sign up
Description: a sign up section for new users would be good to save their changes
Inputs: firstName, familyName, userName, password, email address(email), address (address), phone number (phoneNumber)
Source: keyboard , sign up form
Outputs: creating an account
Destination: online database
Action: The user shall be able to sign up as an option that will appear inside the cart ot in the profile section. a query shall be executed that takes the input from the user and put it accordingly in the users table, in the database, the query will take the password and hash it in the password column in the table for security reasons.
Requires: Database up and running + internet connection
Pre-condition: user filling the sign up form correctly
Post-condition: creating an account for the user
Side-effects: none

Software Requirements/Functional User Requirements/5.1.10

Function: signing in
Description: a user that already has an account can sign in again
Inputs: username and password
Source: keyboard , sign in form, online database
Outputs: accessing the profile
Destination: online database , front screen
Action: the user shall be able to sign in also from the shopping bag section or the profile section. A query shall be executed that takes the username and password to the online database specifically to the table called users, and then it will compare the username and password. So if the user is found in the database , then they will be directed to their profile and all the changes will appear, if the user was not found a message will appear on the sign in form "incorrect username or password please try again".
Requires: Database up and running + internet connection
Pre-condition: user filling the sign in form correctly
Post-condition: accessing the user's account
Side-effects: none

Software Requirements/Functional User Requirements/5.1.11

Function: Displaying Appointments: Booking an Appointment
Description: The user shall be able to book an appointment for consultations.
Inputs: The user chooses a free timeslot
Source: A timeslot in the calendar
Outputs: A message with the date and time of appointment
Destination: Front screen
Action: If time slot is not available, please select another time slot. Else, select this time slot.
Requires: The calendar to always be updated with the availability of the expert and the timeslots that are taken
Pre-condition: Free timeslots available
Post-condition: Mark the timeslot selected as taken after the user book it
Side-effects: Too many requests of appointments

Software Requirements/Functional User Requirements/5.1.12

Function: Contacting Us
Description: The user shall be able to contact the company through the system.
Inputs: A box with all phone numbers and emails
Source:
Outputs: Display the different ways of contacting the company
Destination: Display the different ways of contacting the company
Action: The user will be able to contact the company
Requires: The website to be opened on the menu page
Pre-condition: Null
Post-condition: Contacting during opening hours
Side-effects: Many people calling at the same time

Software Requirements/Functional User Requirements/5.1.13

Function: Purchasing Products
Description: Any user (registered or not) shall be able to purchase one or several jewelry items.
Inputs: Pic an item
Source:
Outputs: The item with its certificate
Destination: Cart page
Action: The user presses on the purchase button and everything contained in the bag will be purchased
Requires: Some items to be selected in the bag
Pre-condition: The payment must be completed
Post-condition: Null
Side-effects: Not knowing who did the purchase

Software Requirements/Functional User Requirements/5.1.14

Function: Adding Products for Purchase
Description: The system shall contain a shopping bag section where products selected by the user for purchase, are stored.
Inputs: An item
Source: Input Text Here
Outputs: The item will be added to the bag
Destination: From catalogue to bag
Action: The user will press on an item he/she desire to buy from the catalogue, this item will then be added to the bag
Requires: At least one item should be selected
Pre-condition: Null
Post-condition: Null
Side-effects: Null

Software Requirements/Functional User Requirements/5.1.15

Function: Removing Selected Products
Description: The user shall be able to remove previously selected products from the shopping bag.
Inputs: The item desired to be deleted
Source:
Outputs: The bag without the item removed
Destination: Bag
Action: The user will select one or more item they desire to remove from the cart by selecting them, after pressing on the remove button they will then be removed from the bag
Requires: A bag that is not empty
Pre-condition: At least one item should be selected
Post-condition: Null
Side-effects: Null

Software Requirements/Functional User Requirements/5.1.16

Function: Paying through Debit/Credit
Description: Users shall make the payment through credit/debit card only
Inputs: Enter the card information
Source:
Outputs: Receipt, with thank you message
Destination: Payment page
Action: The user will enter his/her card information then press on the purchase button to confirm the purchase
Requires: Having items in the bag
Pre-condition: Having a working debit/credit card and having enough money in the account
Post-condition: Null
Side-effects: Null

Software Requirements/Functional User Requirements/5.1.17

Function: Inputting Coupon Code
Description: The user shall be able to submit a coupon code upon checkout for discounts.
Inputs: Null
Source:
Outputs: Offers and bundles for customers
Destination: Bag
Action: By putting the password given in the coupons the user redeems the offer
Requires: Null
Pre-condition: Null
Post-condition: Null
Side-effects: Null

Software Requirements/Functional User Requirements/5.1.18

Function: Summarizing Order
Description: The system shall display a summary of the order before checkout.
Inputs: A bag with items in it
Source:
Outputs: All the items that are inside the bag
Destination: An order summary page after the payment page to confirm the purchase
Action: After pressing on the checkout button that will take the user from the payment page to the order summary page
Requires: A bag with items in it
Pre-condition: Null
Post-condition: Null
Side-effects: Null

Software Requirements/Functional User Requirements/5.1.19

Function: Suggesting Products
Description: Before checking out some products shall be displayed as suggestion to add in the bag.
Inputs: Pictures, String texts
Source: Online database
Outputs: Product pictures and information
Destination: Screen
Action: A dragdrop option will automatically execute a query to select specific product pictures and information from the online database to display in the shopping bag page.
Requires: Connection to the internet
Pre-condition: Available product pictures and information in the Database
Post-condition: The display of products on shopping bag page before checkout
Side-effects: none

Software Requirements/Functional User Requirements/5.1.20

Function: Displaying Package
Description: Before checking out some the type of packaging shall be displayed as an information to the customer.
Inputs: Pictures, String texts
Source: Online Database
Outputs: Packaging picture and information
Destination: Screen
Action: A dragdrop option will automatically execute a query to select a specific packaging picture and information from the online database to display in the shopping bag page before checkout.
Requires: Internet Connection
Pre-condition: Available package pictures and information in the Database
Post-condition: The display of packaging information on shopping bag page before checkout
Side-effects: none

Software Requirements/Functional User Requirements/5.1.21

Function: Displaying Assistance Information
Description: Before checking out, assistance information shall be displayed on the shopping bag page before checkout for the customer in case they needed assistance.
Inputs: String text
Source: Online Database
Outputs: String text
Destination: Screen
Action: A dragdrop option will automatically execute a query to select specific assistance information from the online database to display in the shopping bag page before checkout
Requires: Internet Connection
Pre-condition: Available assistance text in the database
Post-condition: The display of assistance information on shopping bag page before checkout
Side-effects: none

Software Requirements/Functional User Requirements/5.1.22

Function: Inputting Shipping Address
Description: This function will take Address input from the customer to store in an online database, so the company can ship the products to the specific address
Inputs: String text
Source: Keyboard
Outputs: String text
Destination: Screen, online Database
Action: A dragdrop option will automatically generate POST APIs to update online database with address information, and display input fields for the user to input text.
Requires: Internet Connection
Pre-condition: At least one character input on each field
Post-condition: Storing Address text in Online Database
Side-effects: none

Software Requirements/Functional User Requirements/5.1.23

Function: Inputting Billing Address
Description: This function will take Billing Address input from the customer to store in an online database, so the company can receive payment before delivering products.
Inputs: String text
Source: Keyboard
Outputs: String text
Destination: Screen, online Database
Action: A dragdrop option will automatically generate POST APIs to update online database with billing address information, and display input fields for the user to input text.
Requires: Internet Connection
Pre-condition: At least one character input on each field
Post-condition: Storing Billing Address text in Online Database
Side-effects: none

Software Requirements/Functional User Requirements/5.1.24
Function: Live Chatting **Description:** A live chatting box where customers can type-in text to contact customer care instantly. **Inputs:** String text **Source:** Keyboard **Outputs:** String text **Destination:** Screen **Action:** A dragdrop option will automatically generate a sending/receiving function so the user can send and receive message texts. **Requires:** Internet Connection **Pre-condition:** At least one character input **Post-condition:** Chatting text output on both ends (Customer's screen and company's screen) **Side-effects:** none

6 Scenarios

Initial Assumption: the user is trying to search for products. **Normal:** : the user pressed on search, they are prompted to type what they want. **What can go wrong:** the user did not type correctly. In this case, the system should filter what the user wrote to be as close as possible to the expected answer. The search engines did not find what the user is looking for, Therefore the user is asked to type what they want again. **Other activities:** the user searched for many products at the same time. **System State on completion:** : the searched product was found and was shown to the user

Initial Assumption: the user is trying to search through in the navigation bar. **Normal:** : : the user clicks on the any section in the navigation bar then the subsection appears where they can click and go to the page accordingly. **What can go wrong:** : nothing happens, it stays on the same page, probably a glitch happened the user is asked to refresh. The user clicks on a specific section and it is empty, this means that the section is empty till now. **Other activities:** the user clicked on multiple sections multiple times. **System State on completion:** : the user is redirected to a subsection and after that to the product.

Initial Assumption: the user is trying to sign in.
Normal: the user enters his/her username and password and then they are asked to authenticate. Then he/she is prompted to write a code that was sent to their email or phone number to check if it is really them. then when the user writes everything and is accepted, he/she are redirected to the page as their profile user.
What can go wrong: : the user writes a wrong username or password , and therefore they are prompted to type it again a certain number of times. The user might have forgot their password , so they can restore their password through a process containing their email and phone number and some information that they wrote while signing up
Other activities: the user writes an invalid username or password.
System State on completion: the user signs up, and enters the website on his profile, where everything that he did on the website is saved there.

Initial Assumption: the user doesn't have an account and is trying to sign up.
Normal: the user enter his/her information in the form, chooses a username and a Password and then they are prompted to sign in.
What can go wrong: : the user enters a username that already exists, therefore They are prompted to use another one. The user enters an invalid email, therefore they are also asked to Write a valid one, and will check by sending an email to that Account. The user failed to fill one of the information that were important Therefore they are redirected to the sign up page to continue the Process and the system will mark the empty information form. The user failed to verify the password therefore they are prompted to rewrite the password.
Other activities: the user fails to submit the form.
System State on completion: a profile will be created for the user and pushed into the database.

Initial Assumption: displaying models on the home screen.
Normal: a video or a photo will be fetched from the database to the home page and It will display the wanted picture.
What can go wrong: : an error while retrieving the information from the database so maybe the wrong photo will appear, The admin might make a mistake while inserting the model's picture or video therefore the wrong info is again shown on the screen.
Other activities: nothing is shown.
System State on completion: the picture will be uploaded and shown in the database.

Initial Assumption: displaying the company's summary.
Normal: a simple summary paragraph will be displayed on the home screen.
What can go wrong: an error while fetching the information from the database The admin might have inserted the wrong summary about the company.
Other activities: nothing appears on the screen.
System State on completion: the summary will be shown on the home screen.

Initial Assumption: displaying featured products.
Normal: some of the featured products will be displayed on the front screen.
What can go wrong: an error while doing the calculations to get the featured products The admin might have inserted the wrong products to be featured.
Other activities: nothing is shown on the screen.
System State on completion: the products will be shown on the screen.

Initial Assumption: displaying jewelry categories on the screen .
Normal: some pictures of the products of the jewelry categories will be displayed on the home screen.
What can go wrong: error while fetching the information from the database The admin might have inserted some wrong categories in the database.
Other activities: : jewelry displayed.
System State on completion: the jewelry categories pictures and information will be displayed on the home screen.

Initial Assumption: displaying services.
Normal: the services section will be shown with 4 services The user can click on any services and get the according service.
What can go wrong: error while retrieving services from the database Wrong link for each service and therefore the suer might not be able to benefit from the services Some services are not available therefore a message will be shown to the user in order to let him/her know the status of the service.
Other activities: : wrong info is shown.
System State on completion: the proper 4 services will be displayed.

Initial Assumption: displaying moissanite education.
Normal: a simple paragraph will be shown on the home screen that displays the information about moissanite.
What can go wrong: the admin might have inserted the wrong information About moissanite in the database therefore the wrong info Will be shown. An error while retrieving the information from the database.
Other activities: : nothing is shown on the screen.
System State on completion: the information will be shown on the home screen.

Initial Assumption: : contact us.
Normal:the user will be able to contact the enterprise .
What can go wrong: if the user does not receive a response – if there is no service.
Other activities: None.
System State on completion: the user would have made contact with the company.

Initial Assumption: : purchasing products.
Normal:any user registered or not is able to buy one or more item by selecting products and adding them to the bag .
What can go wrong: if the user .
Other activities: None.
System State on completion: the items will be bought by the user.

Initial Assumption: : adding products for purchase.
Normal:the user will be able to select items and add them to a bag to purchase them by pressing on the "add to bag" button.
What can go wrong: if the user adds an item to the bag by selecting it by mistake – a bug occurs and the item added to the bag was wrongly priced .
Other activities: None.
System State on completion: the items will be bought by the user.

Initial Assumption: : removing selected products.
Normal:the user will be able to remove products from his/her bag if they do not desire to buy them anymore by selecting the item and pressing on the "remove from bag" button.
What can go wrong: none .
Other activities: None.
System State on completion: the item selected will be removed from the bag .

Initial Assumption: : debit/credit payment.
Normal:the user should either pay by credit card or by debit card only by pressing on one of the two buttons "debit card" or "credit card".
What can go wrong: no money in the account .
Other activities: None.
System State on completion: the payment would be completed and a thank you message with a receipt will appear .

Initial Assumption: : coupons.
Normal:the user will be able to benefit from discounts and offers by redeeming gifts from coupons offered by the company.
What can go wrong: : if a bug occurs and too many coupons are offered the company might go bankrupted.
Other activities: None.
System State on completion: the user will benefit from the bundles offered and have discount on products while buying items .

Initial Assumption: : order summary.
Normal:the user will receive a summary of everything he ordered in order for him to make sure of every item present in the bag .
What can go wrong: : a miscopy of the data from the bag to the order summary .
Other activities: None.
System State on completion: the list of items in the bag will appear on a new page with everything the user selected.

Initial Assumption: : booking an appointment.
Normal:the user spots the booking option and books an appointment by selecting the empty option.
What can go wrong: no free timeslots to book an appointment.
Other activities: None.
System State on completion: an appointment will be booked.

Initial Assumption: The user has already selected products for purchase and is on the shopping bag page at the section titled "You May Also Like".
Normal: Products are displayed as a suggestion for the customer. If the user clicks on "Add to Bag", the system will automatically add the product to the shopping bag list.
What can go wrong: Data in the online database might not be existent, so the pictures and texts will not appear on the system.
Other activities: None.
System State on completion: After adding the suggested product/s to the bag, the shopping bag page is still the same but extra products are added to the list/shopping bag.

Initial Assumption: The user has already selected products for purchase and is on the shopping bag page, scrolled all the way down.
Normal: the package is displayed as an information to the customer so he can see what type of packaging his products will be packed in.
What can go wrong: Nothing
Other activities: None.
System State on completion: NA

Initial Assumption: The user is on the shopping bag page, and ready to click on 'Checkout' button.
Normal: Under the 'Checkout' button, a section titled 'Need Help' is displayed with all assistance information the user might want to use. If the user clicks on one of the information displayed, a drop down with detailed explanation will appear.
What can go wrong: Nothing
Other activities: None
System State on completion: Page still the same.

Initial Assumption: The user has clicked on the 'Checkout' button, and is on the checkout page.
Normal: User sees input fields to input their address information or the address where they would like the products to be delivered. If user inputs address information on all fields, they'll be ready to click on 'continue'.
What can go wrong: User inputs gibberish text not related to the address.
Other activities: User does not input information on all fields, so the page will reload with a red string text indicating that the user should input information.
System State on completion: Page moves to another page titled 'Billing Address'.

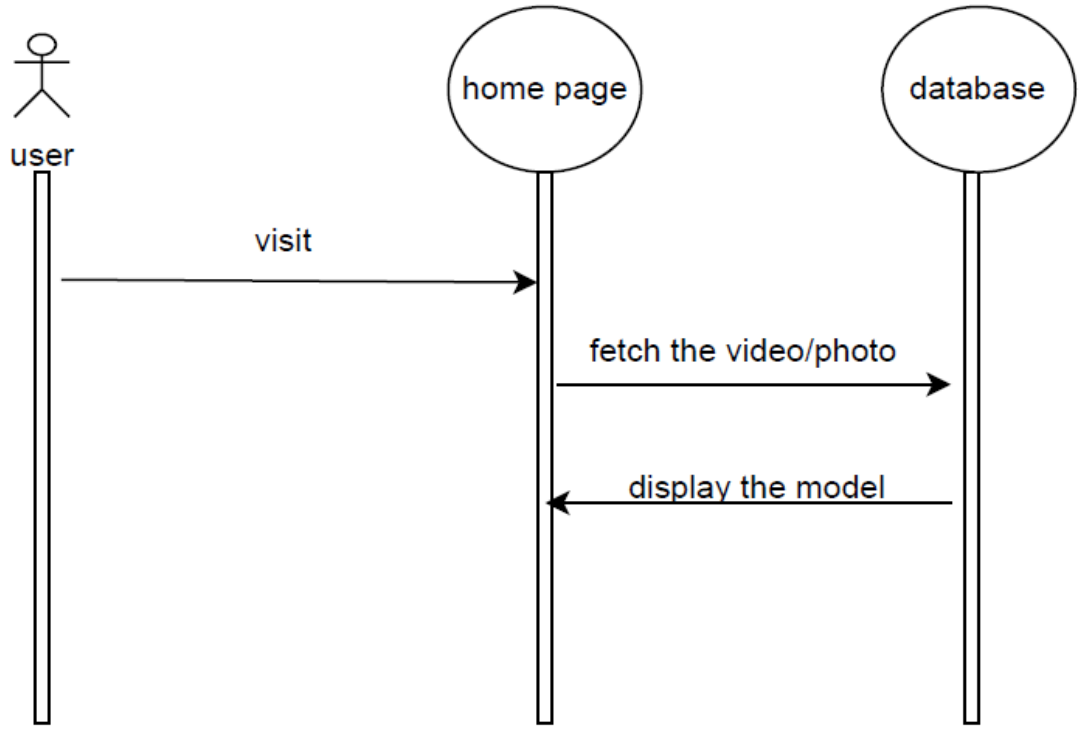
Initial Assumption: The user has clicked on 'Continue' button after inputting address information, and is on the page titled 'Billing Address'.
Normal: User sees input fields to input their billing address information. If user inputs billing address information on all fields, they'll be ready to click on 'Complete purchase'.
What can go wrong: User inputs billing address information of an empty bank account.
Other activities: User does not input information on all fields, so the page will reload with a red string text indicating that the user should input information.
System State on completion: System displays 'purchase completed' and goes back to home page.

Initial Assumption: The user is on any page on the website, and would like to contact customer care and get an instant reply.
Normal: User finds a bubble for live chatting on the bottom right corner of the screen. If they click on it, then a chat system appears with an automated first message from the system, then if user enters a message, an employee from the company will reply as soon as they receive the message from the user.
What can go wrong: No one is available to reply to the user that started the chat.
Other activities: None
System State on completion: The conversation is over and the page is still the same, but the chatting system is closed.

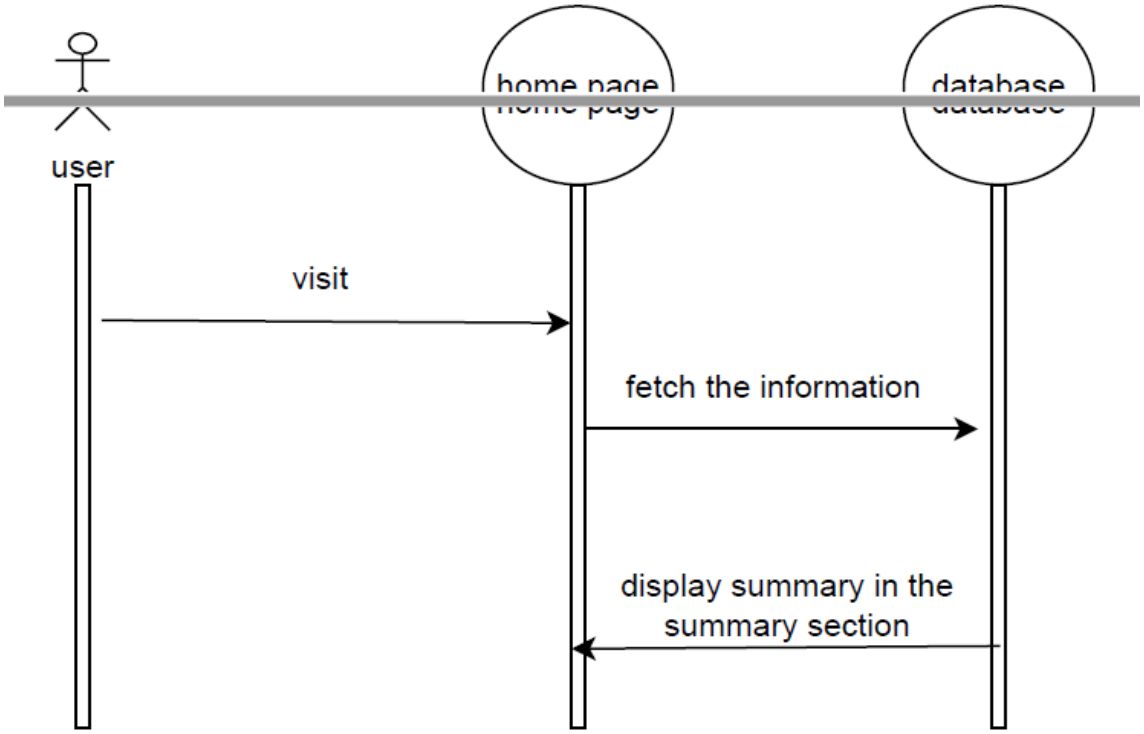
7 System Models

7.1 Sequence Diagrams

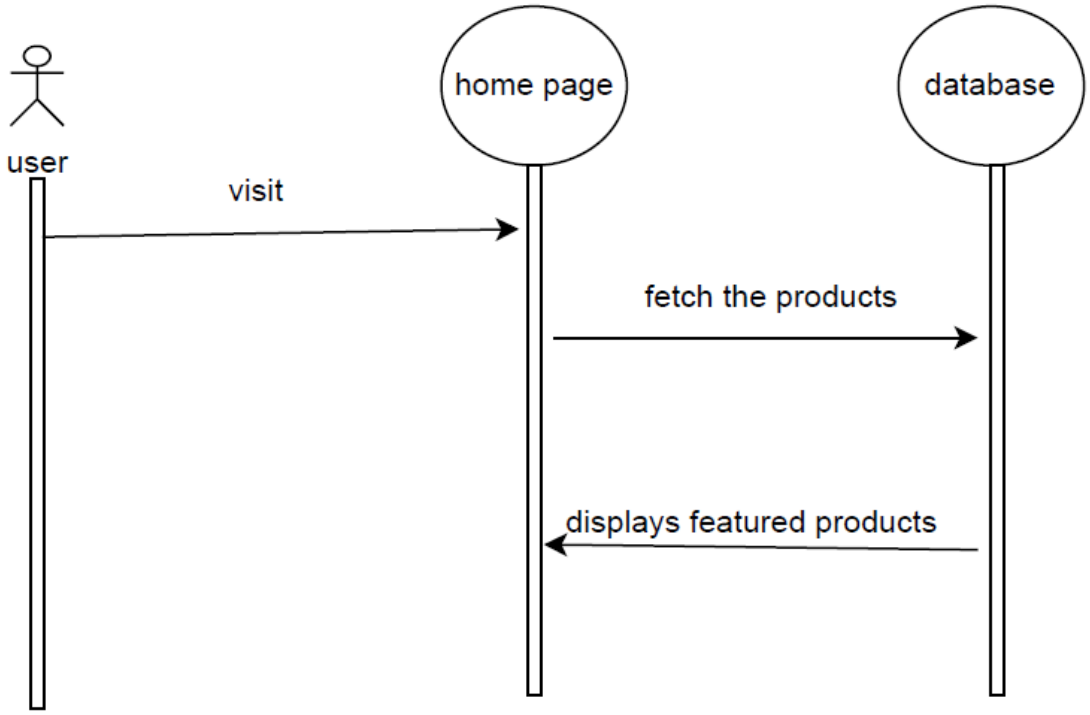
7.1.1 Displaying Jewelry Models



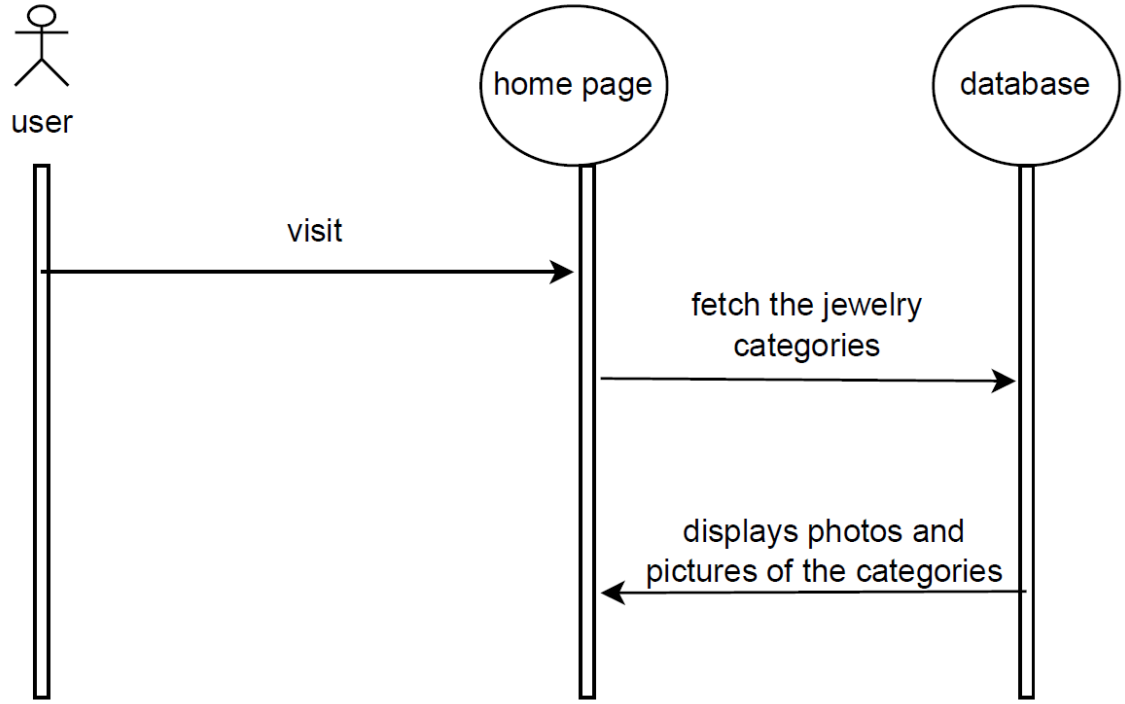
7.1.2 Displaying Company's Summary



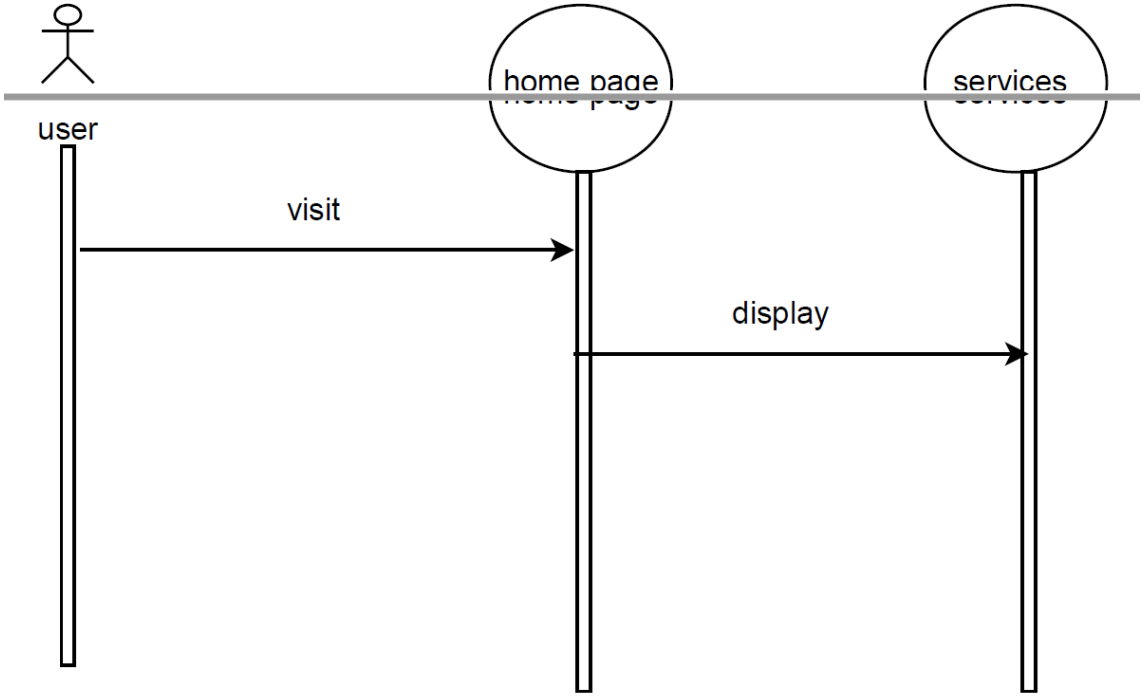
7.1.3 Displaying Featured Products



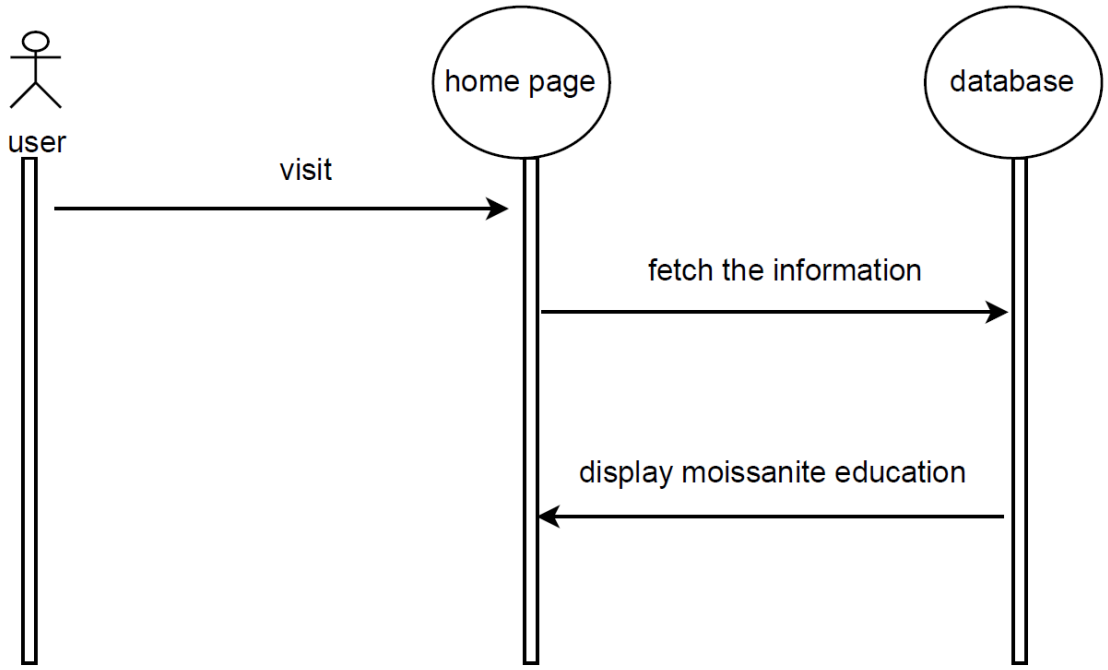
7.1.4 Displaying Categories



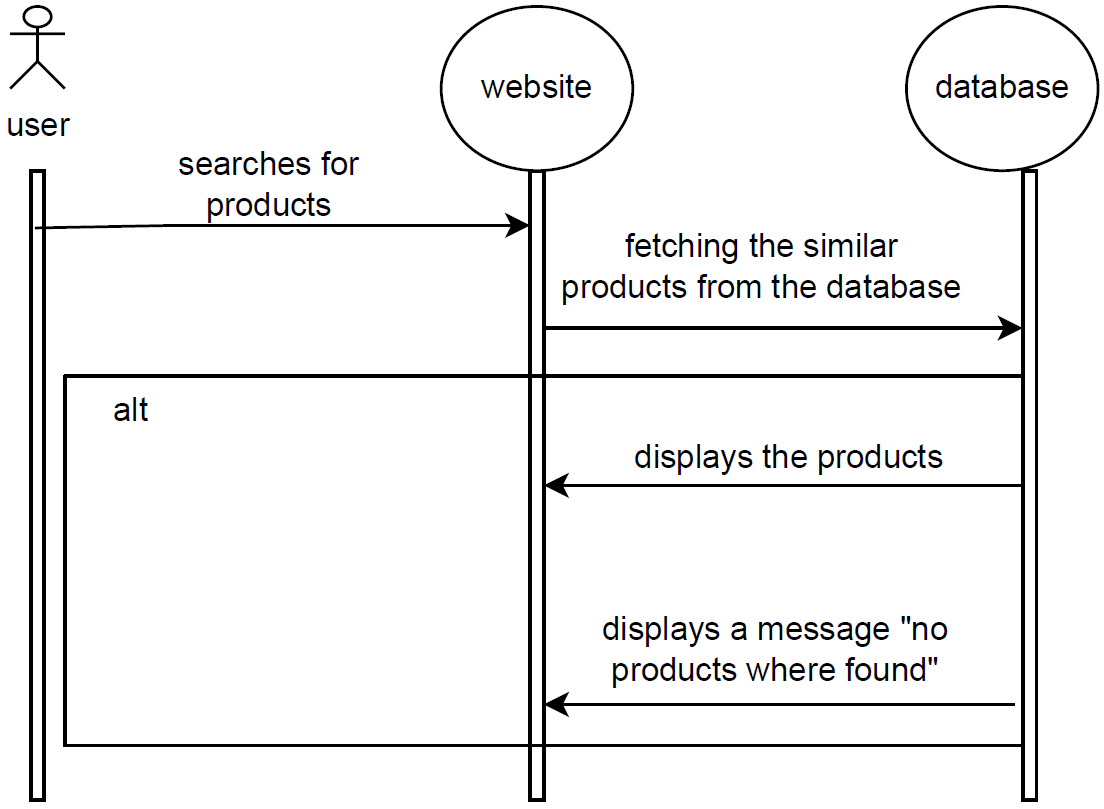
7.1.5 Displaying Services



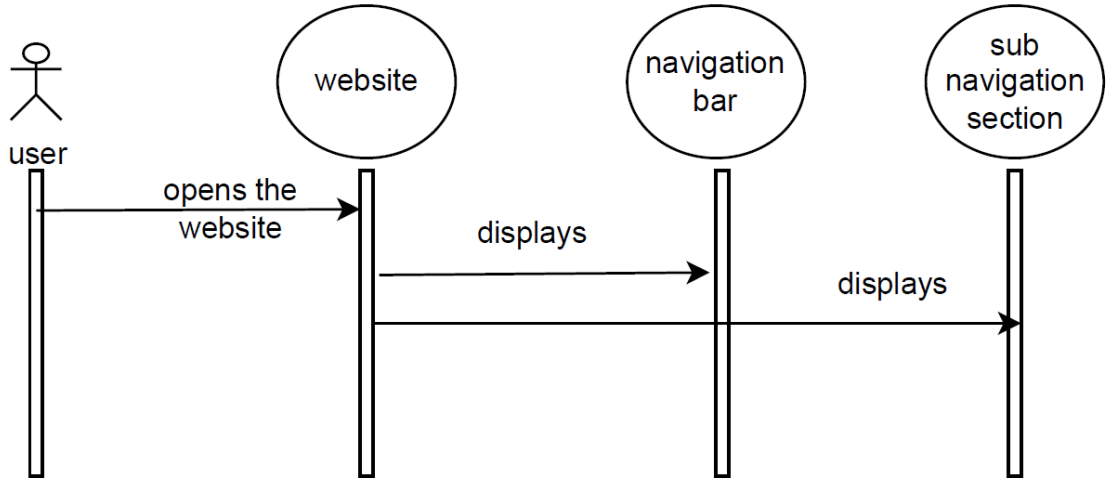
7.1.6 Displaying Moissanite Education



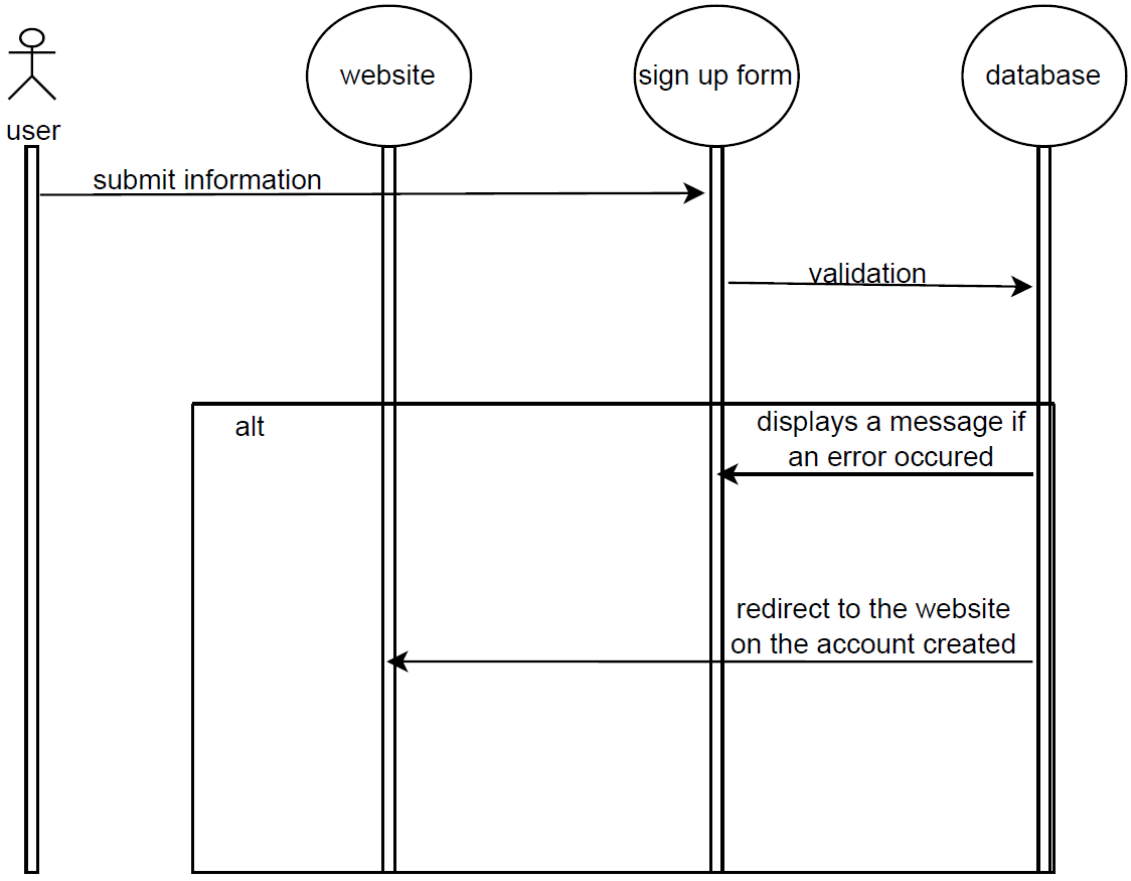
7.1.7 Searching For Products



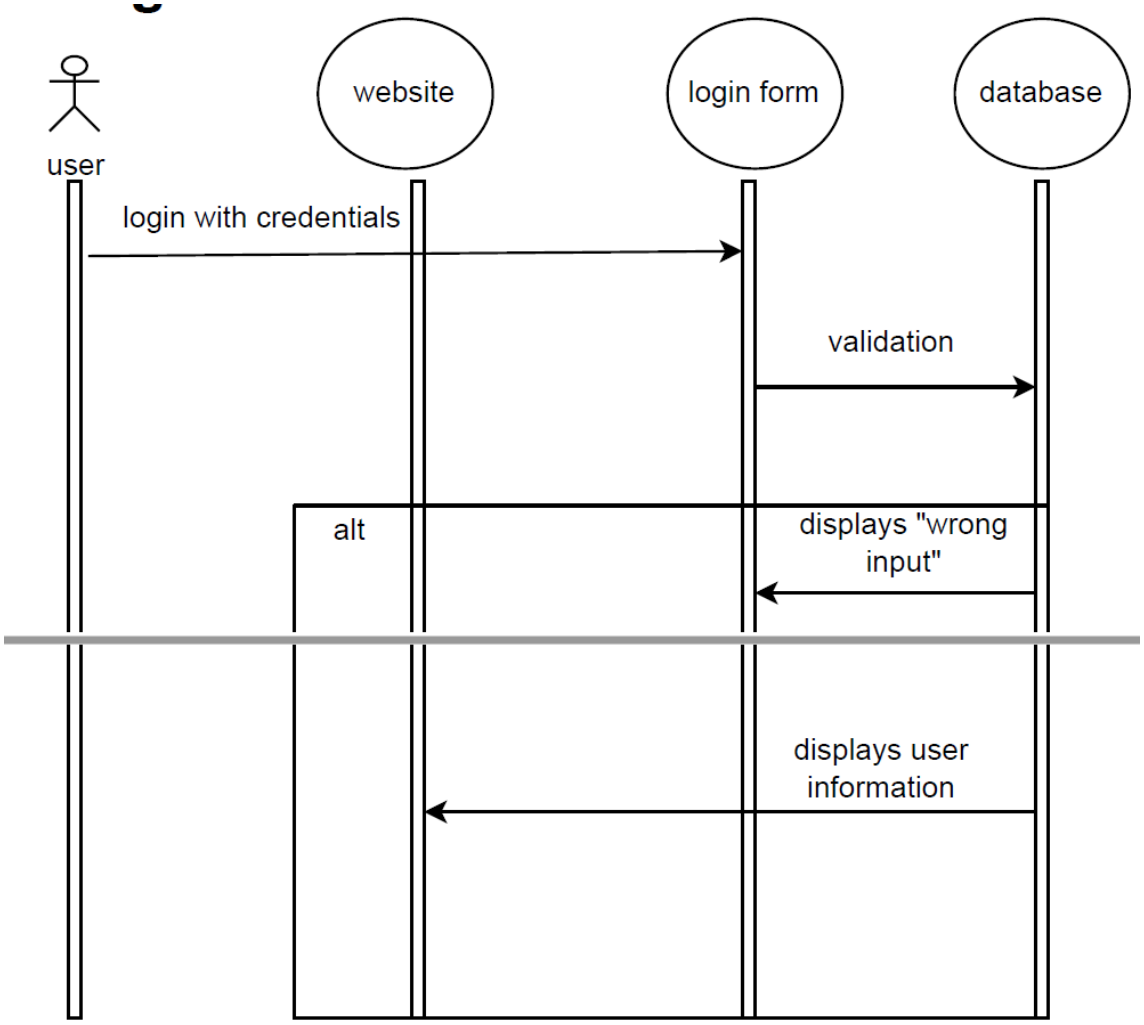
7.1.8 Navigating Through a Section Bar



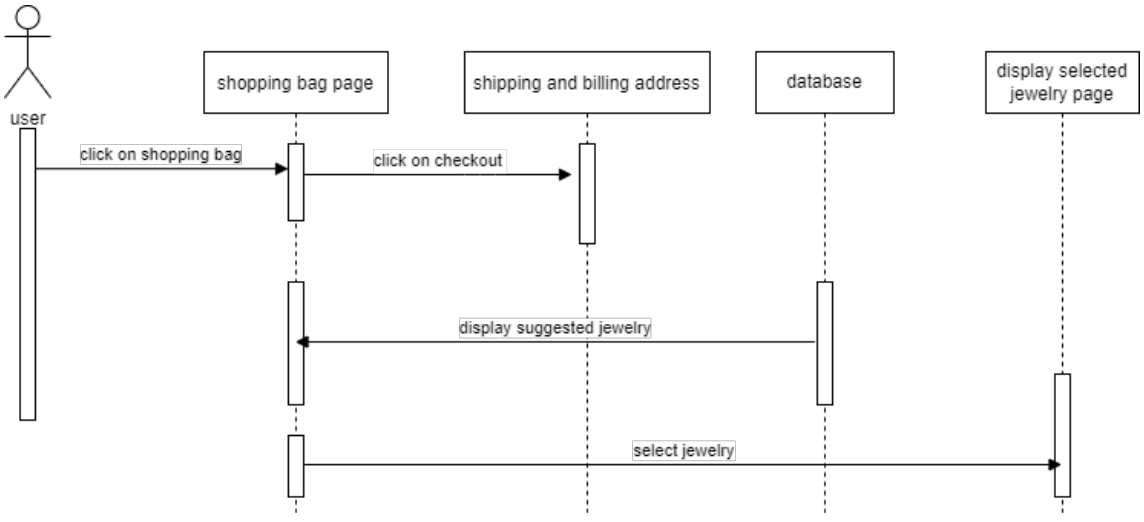
7.1.9 Creating an Account



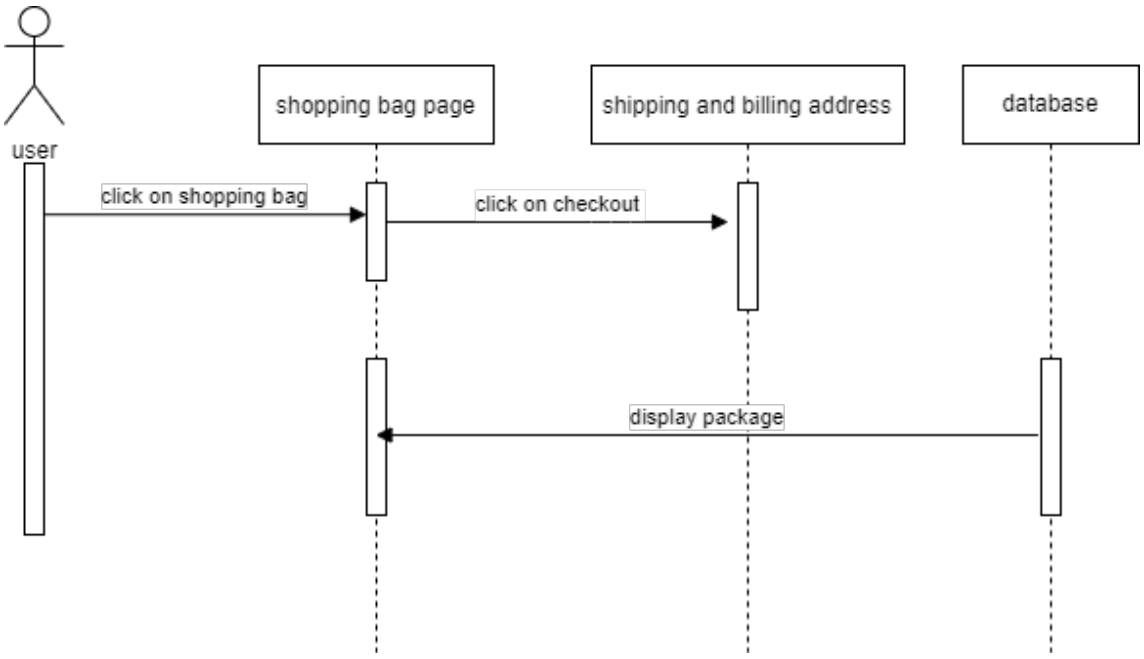
7.1.10 Creating an Account



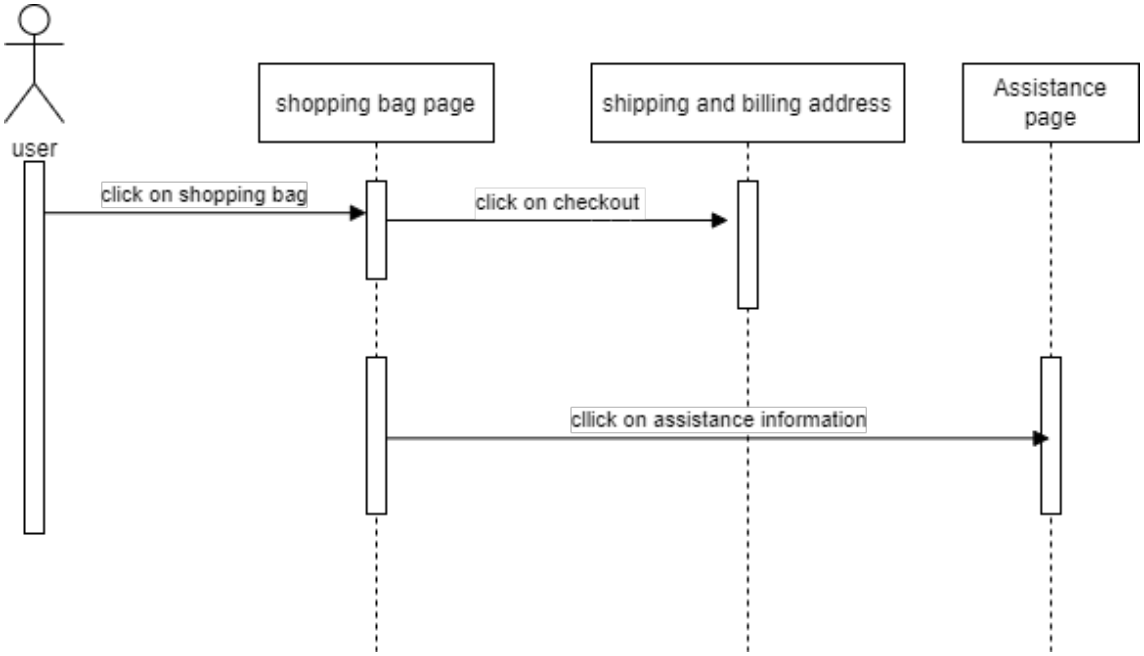
7.1.11 Suggesting Products



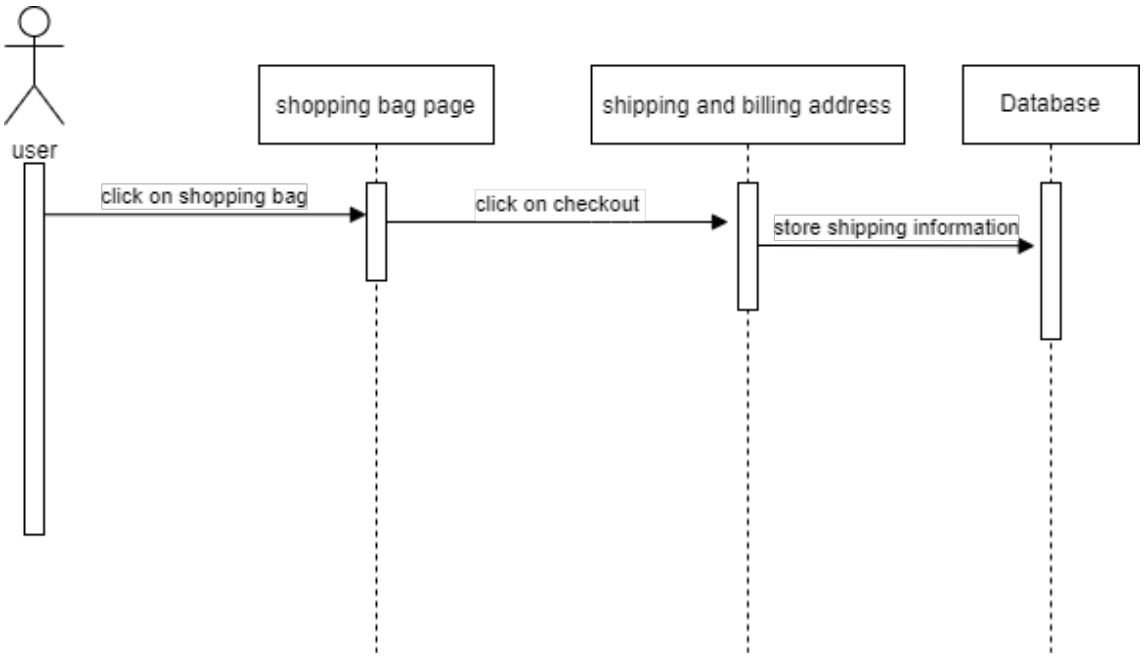
7.1.12 Displaying Package



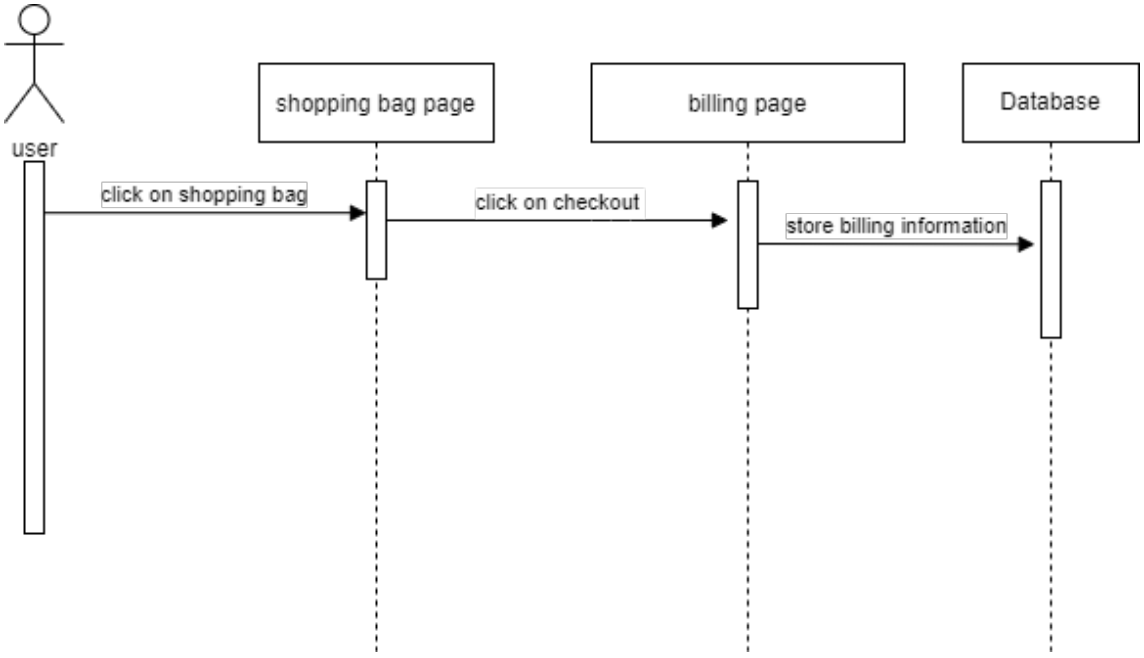
7.1.13 Providing Assistance Information



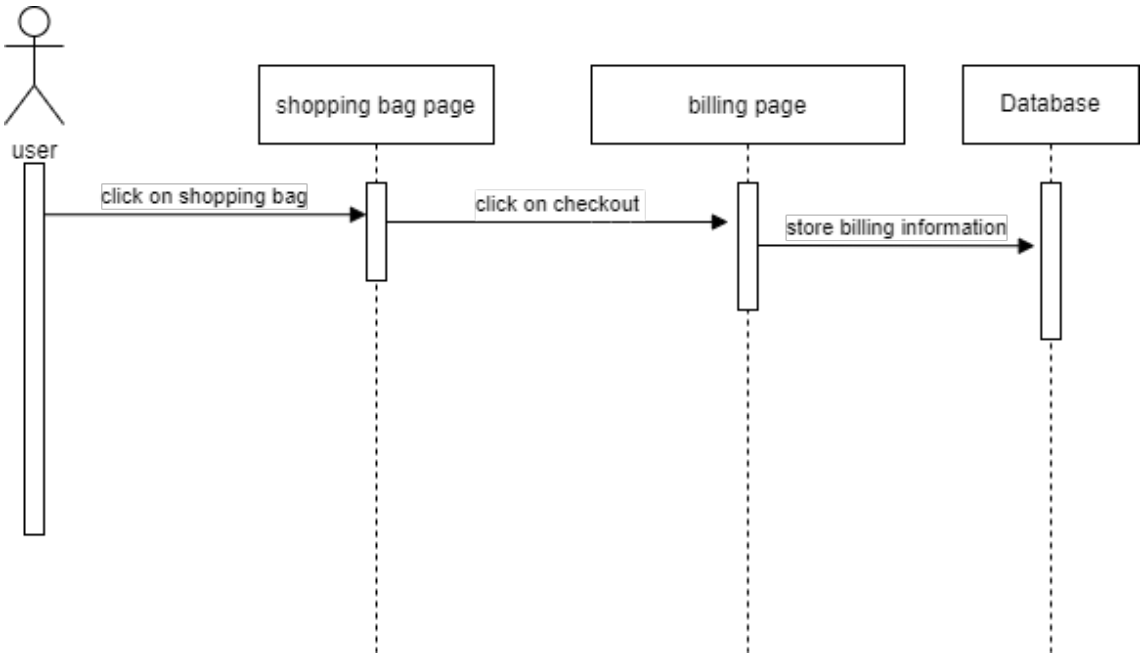
7.1.14 Inputting Shipping Address



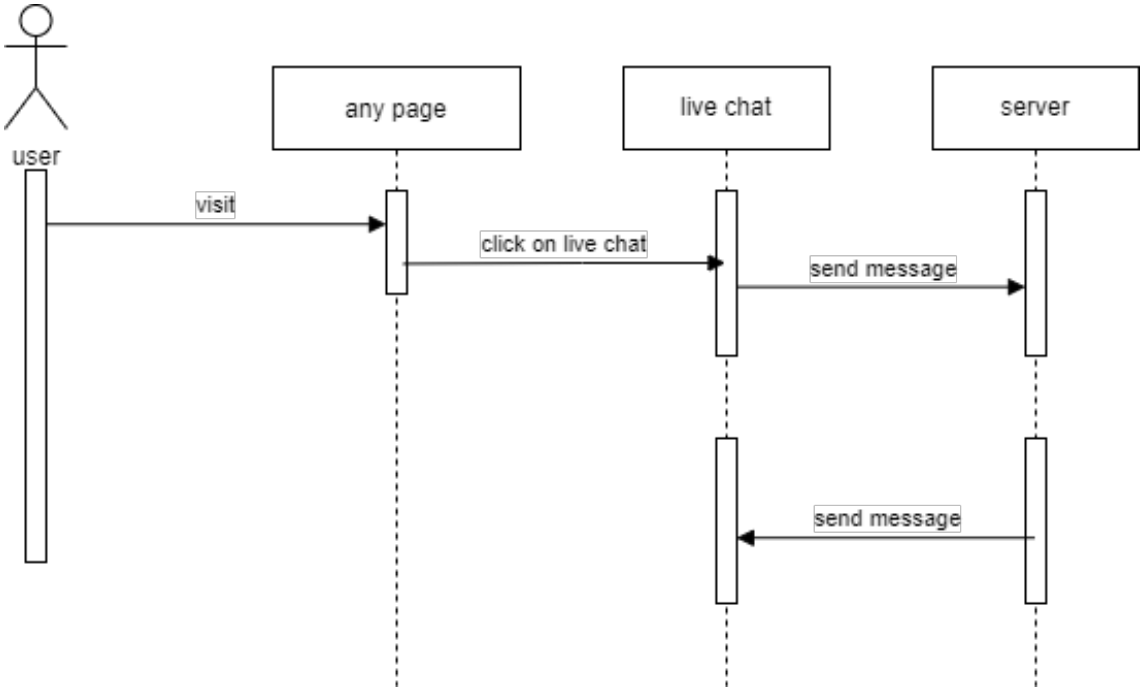
7.1.15 Inputting Billing Address



7.1.16 Inputting Billing Address

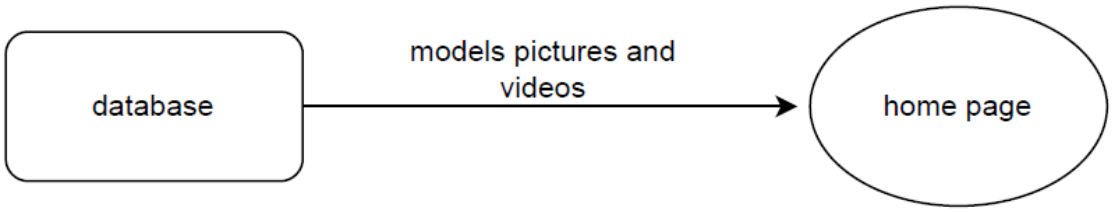


7.1.17 Live Chatting

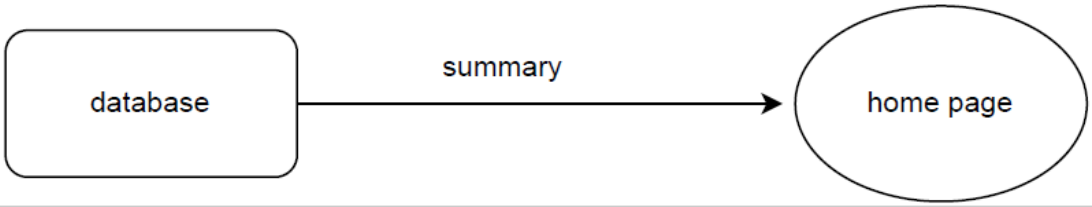


7.2 Data-flow Diagrams

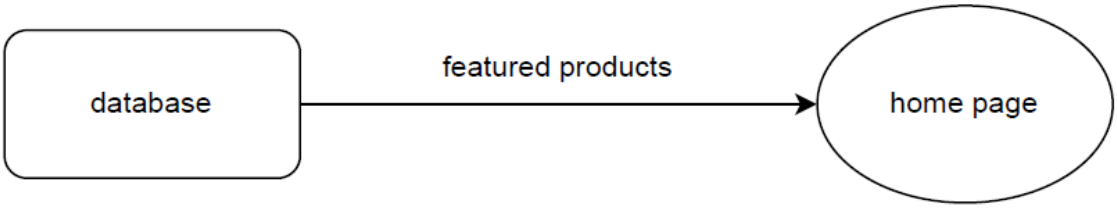
7.2.1 Displaying Jewelry Models



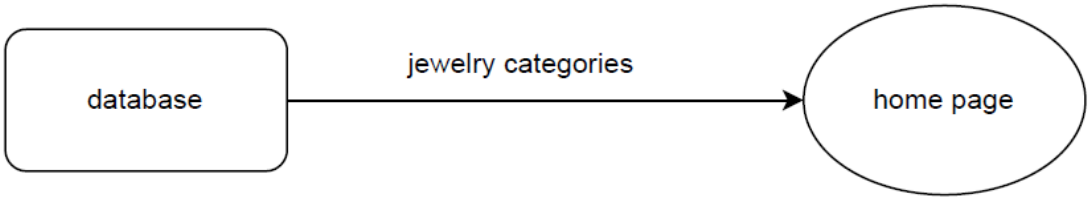
7.2.2 Displaying Company's Summary



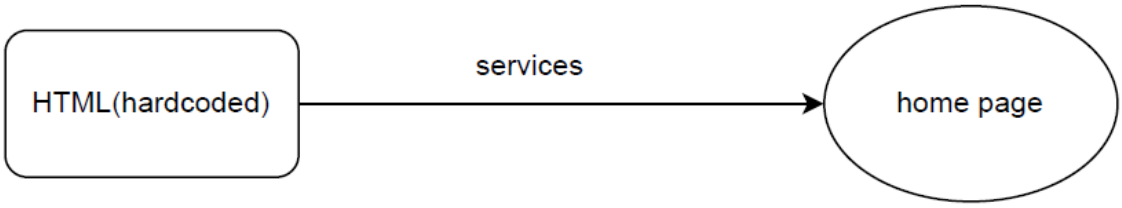
7.2.3 Displaying Featured Products



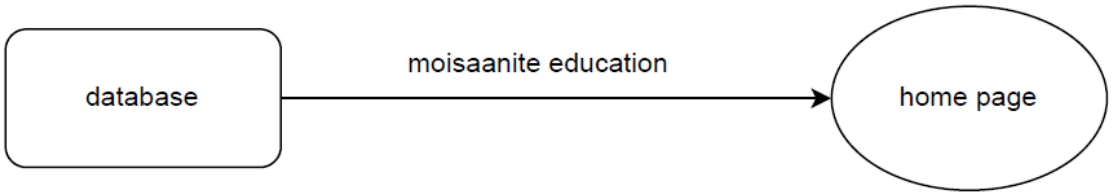
7.2.4 Displaying Categories



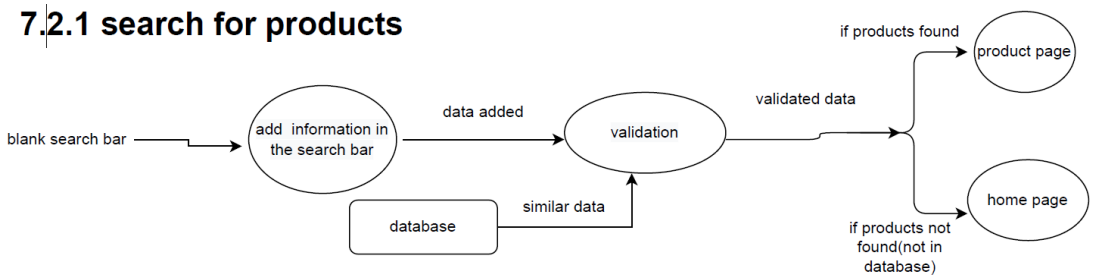
7.2.5 Displaying Services



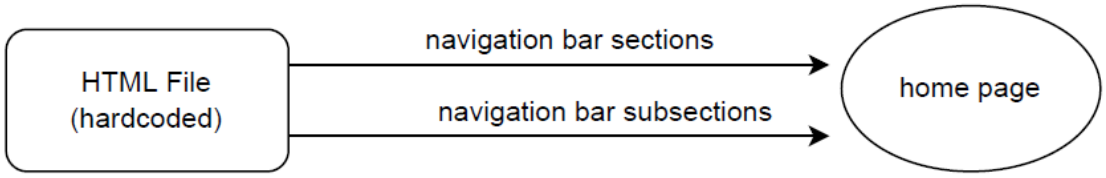
7.2.6 Displaying Moissanite Education



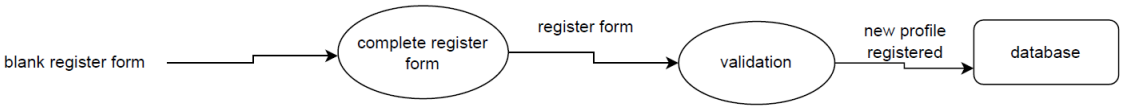
7.2.7 Searching For Products



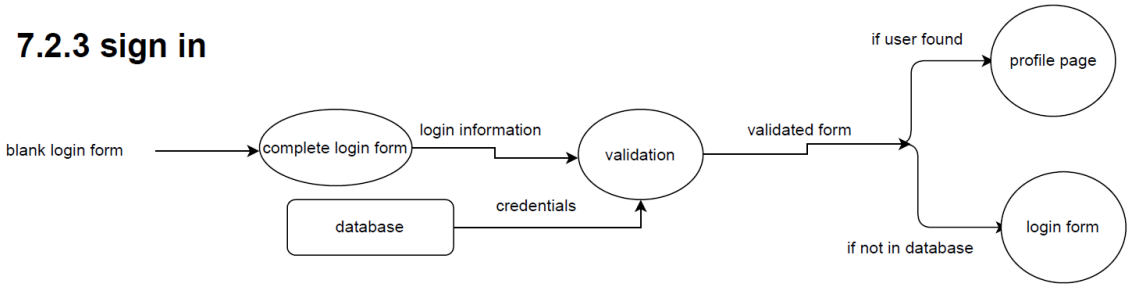
7.2.8 Navigating Through a Section Bar



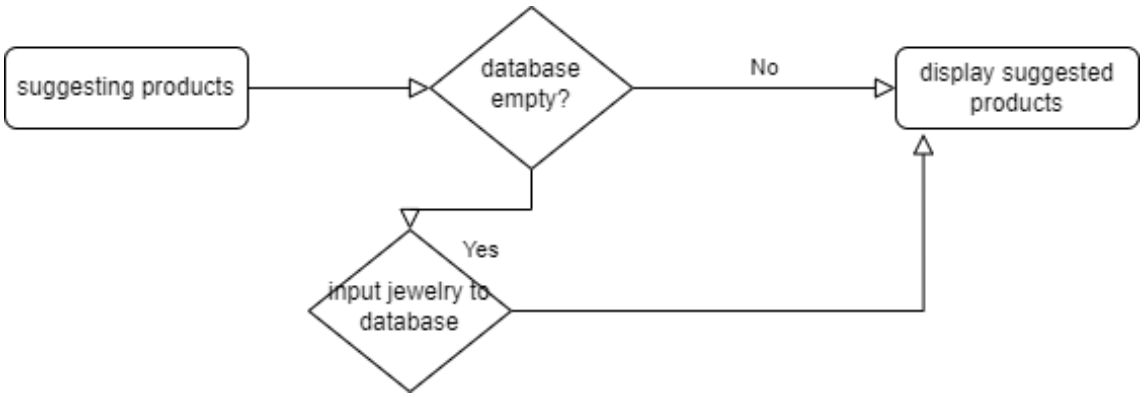
7.2.9 Creating an Account



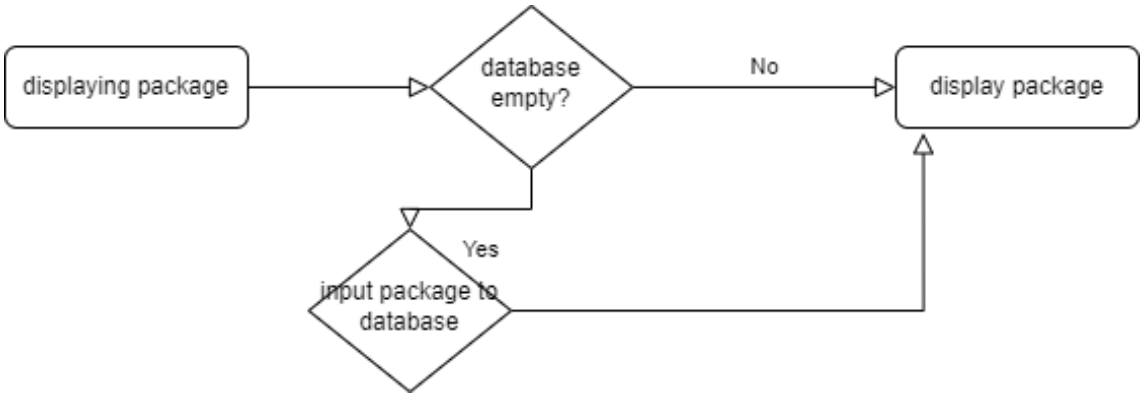
7.2.10 Signing in



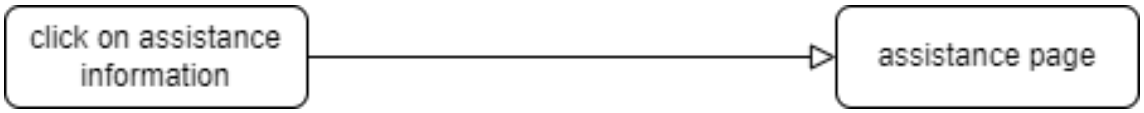
7.2.11 Suggesting Products



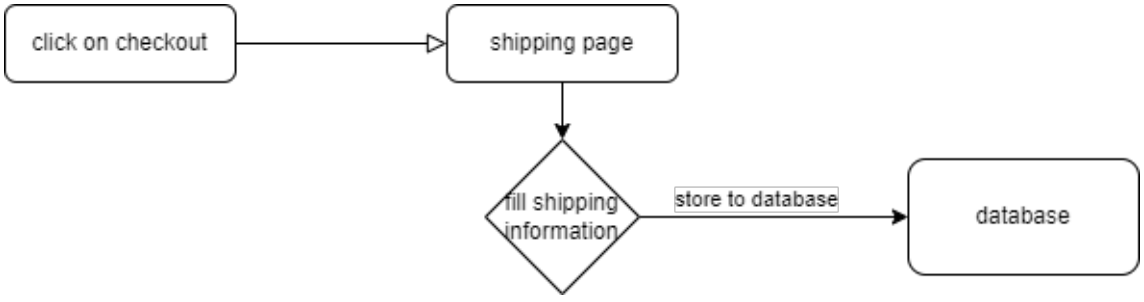
7.2.12 Displaying Package



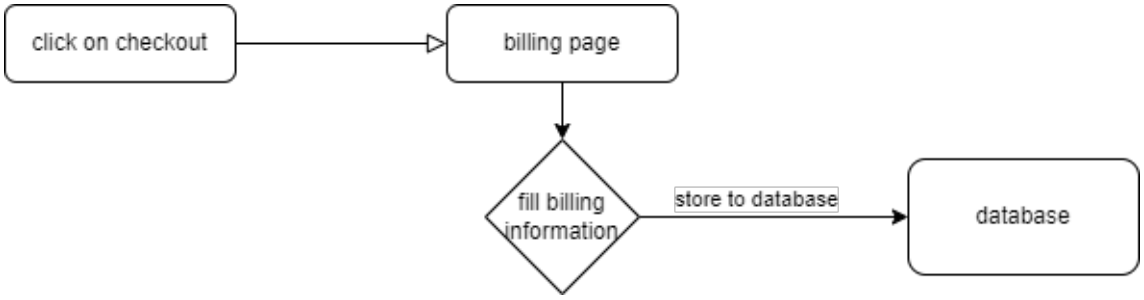
7.2.13 Providing Assistance Information



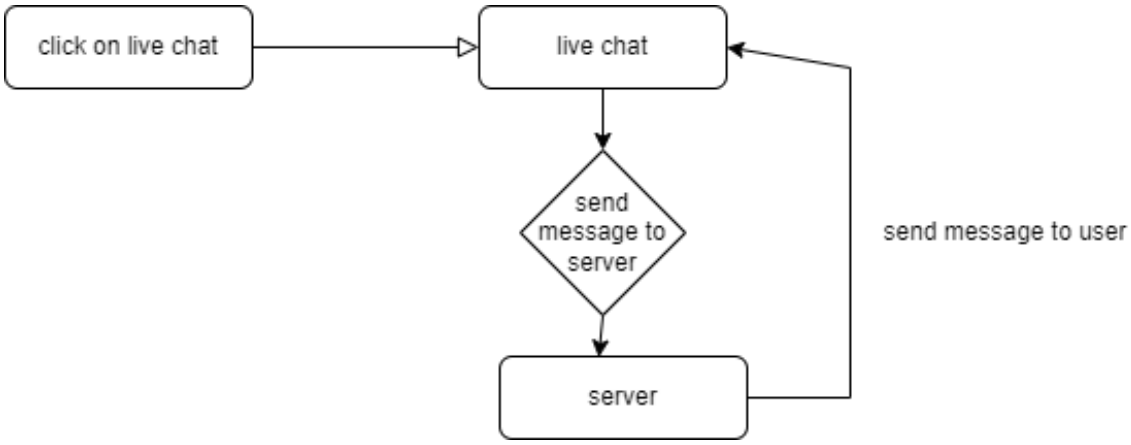
7.2.14 Inputting Shipping Address



7.2.15 Inputting Billing Address

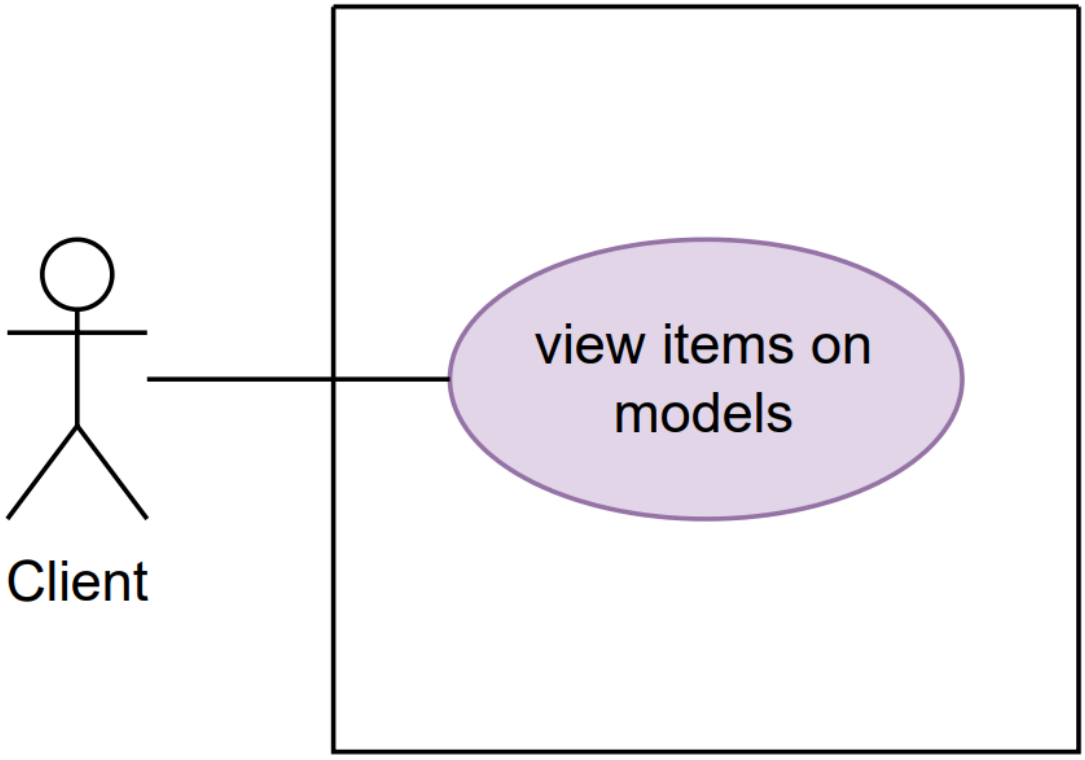


7.2.16 Live Chatting

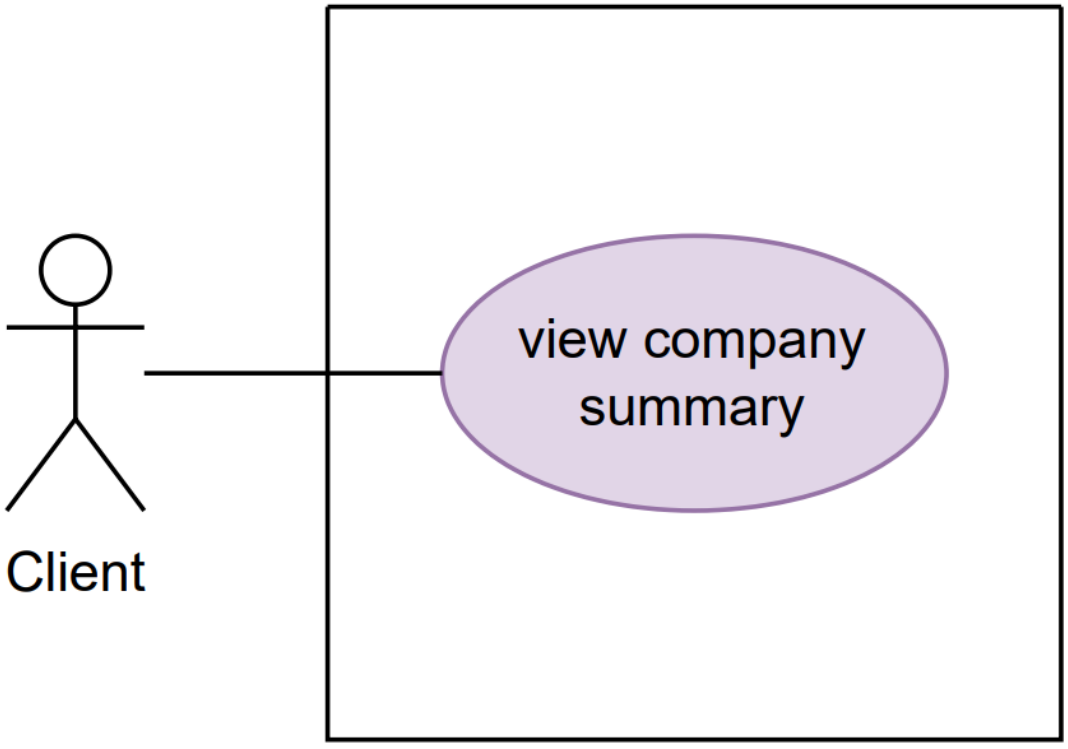


7.3 Use Cases

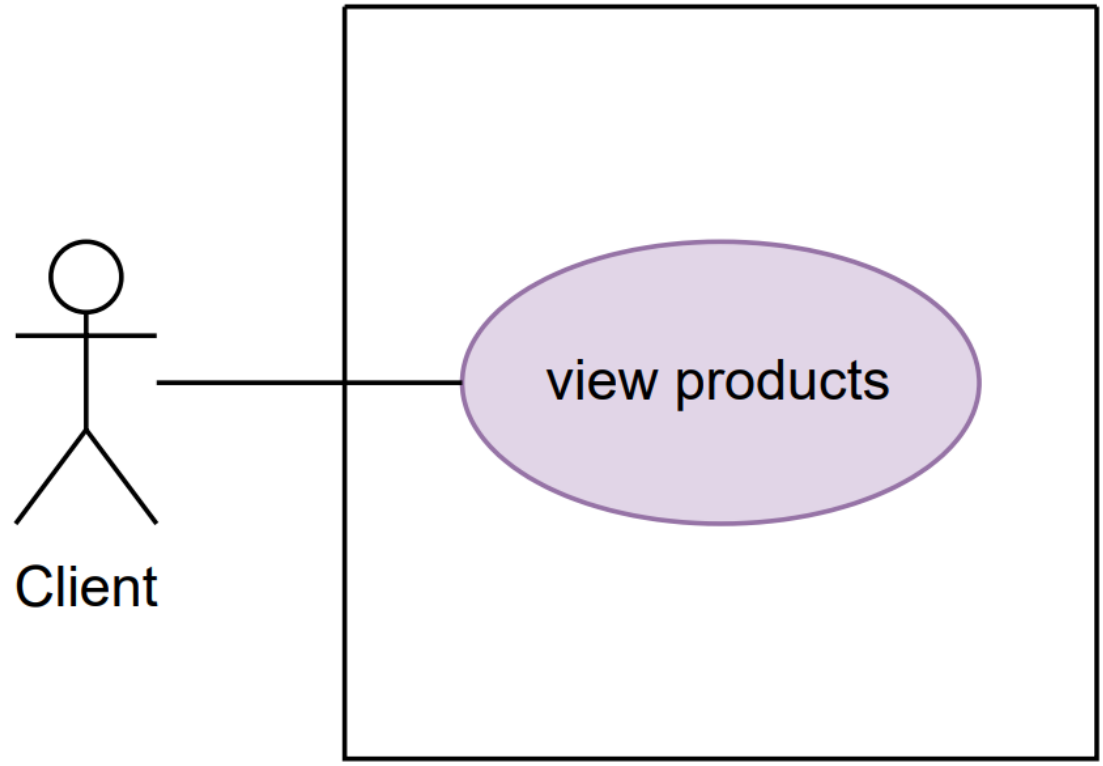
7.3.1 Displaying Jewelry Models



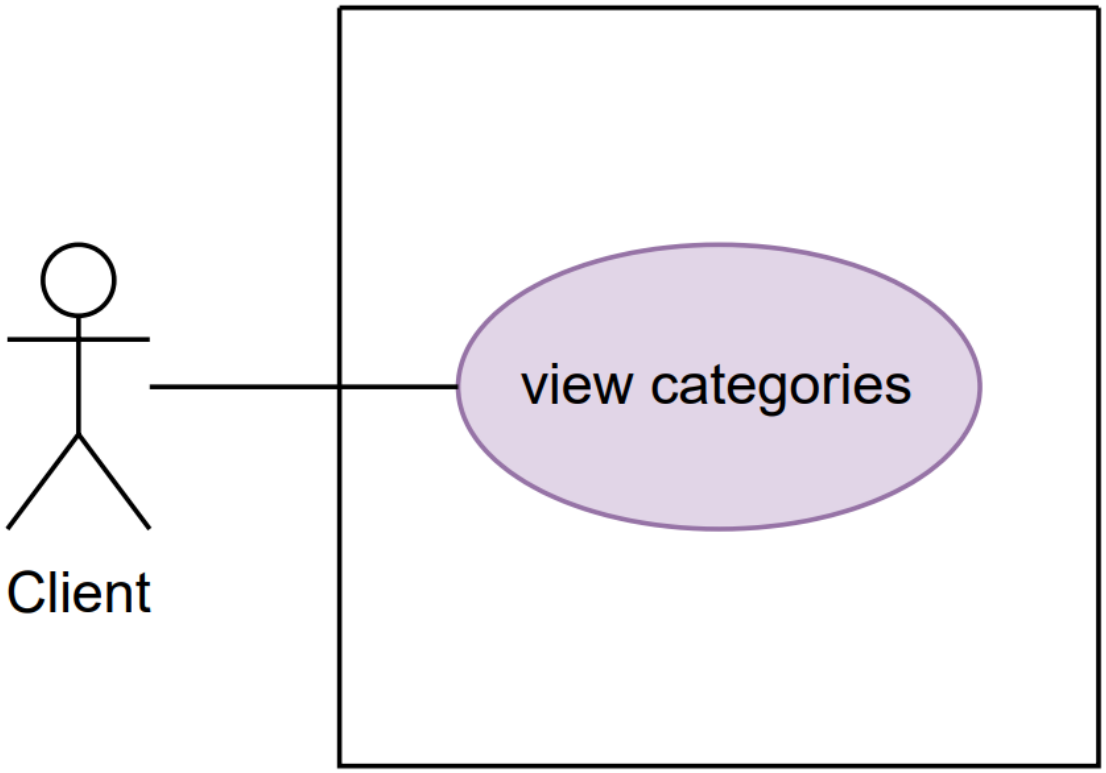
7.3.2 Displaying Company's Summary



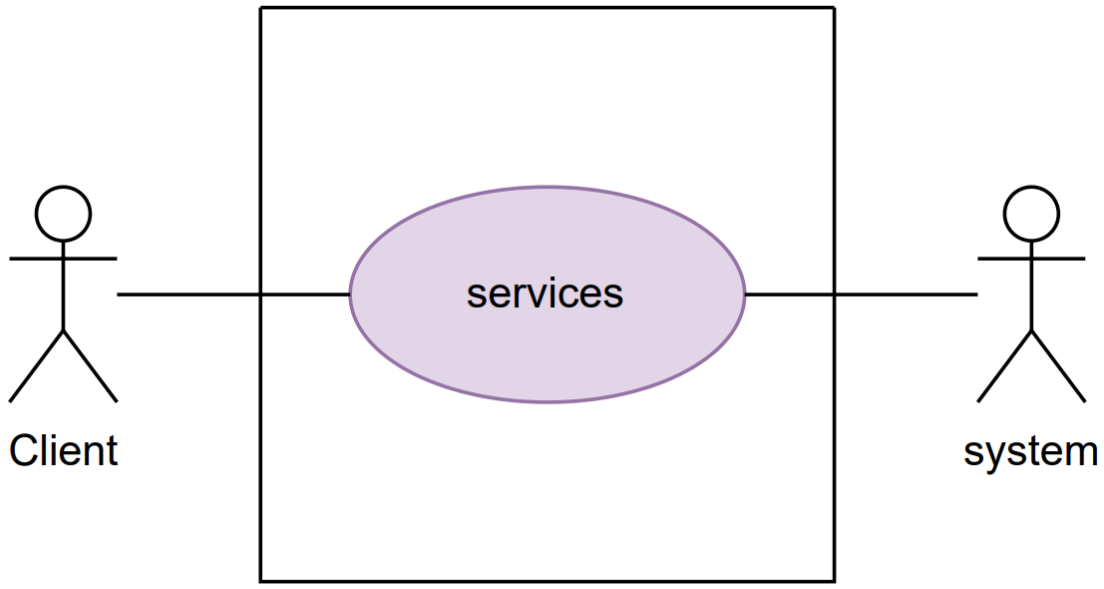
7.3.3 Displaying Featured Products



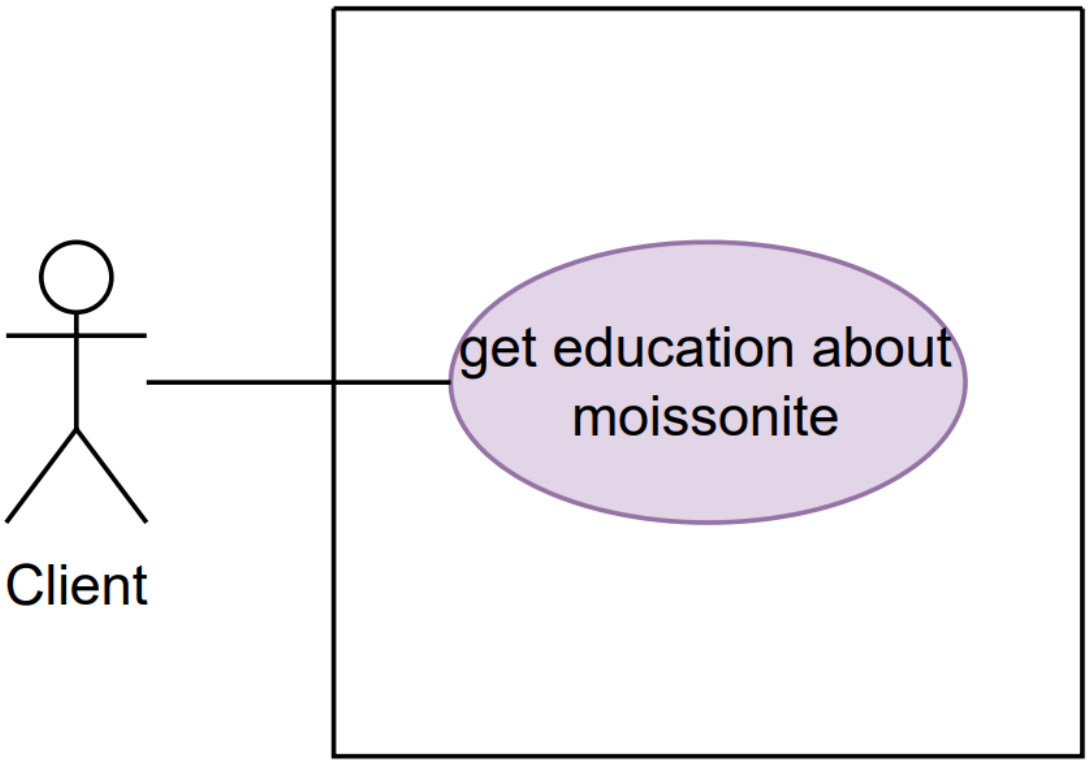
7.3.4 Displaying Categories



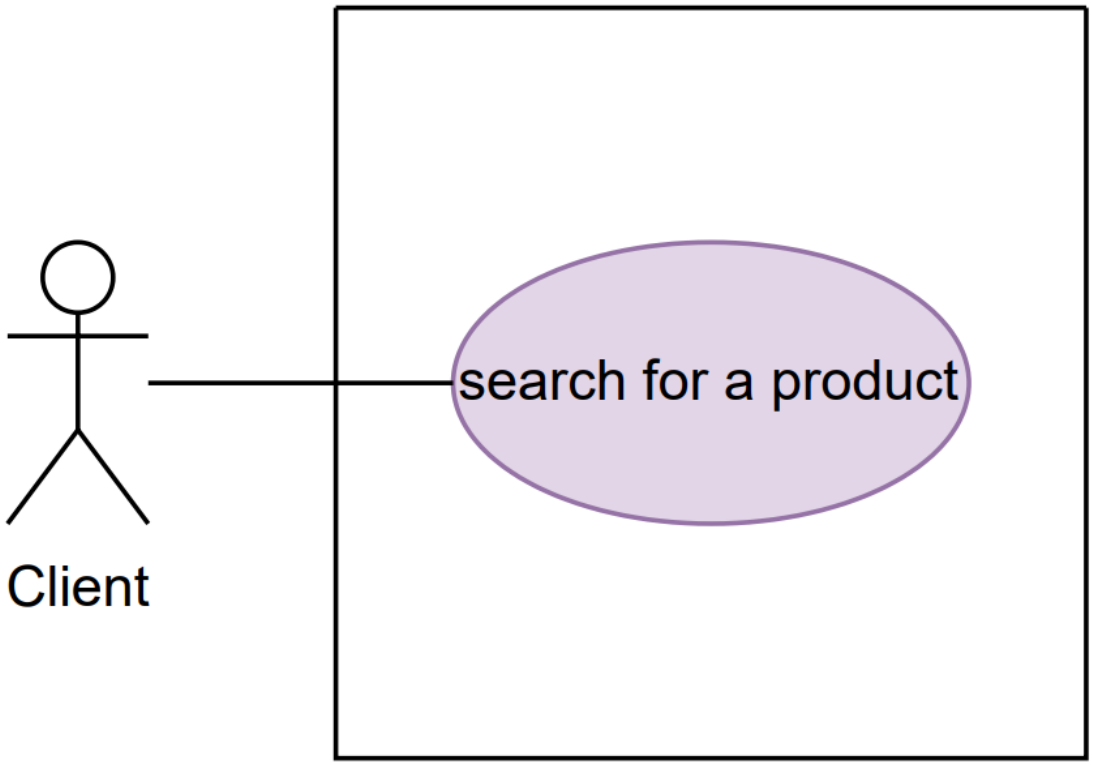
7.3.5 Displaying Services



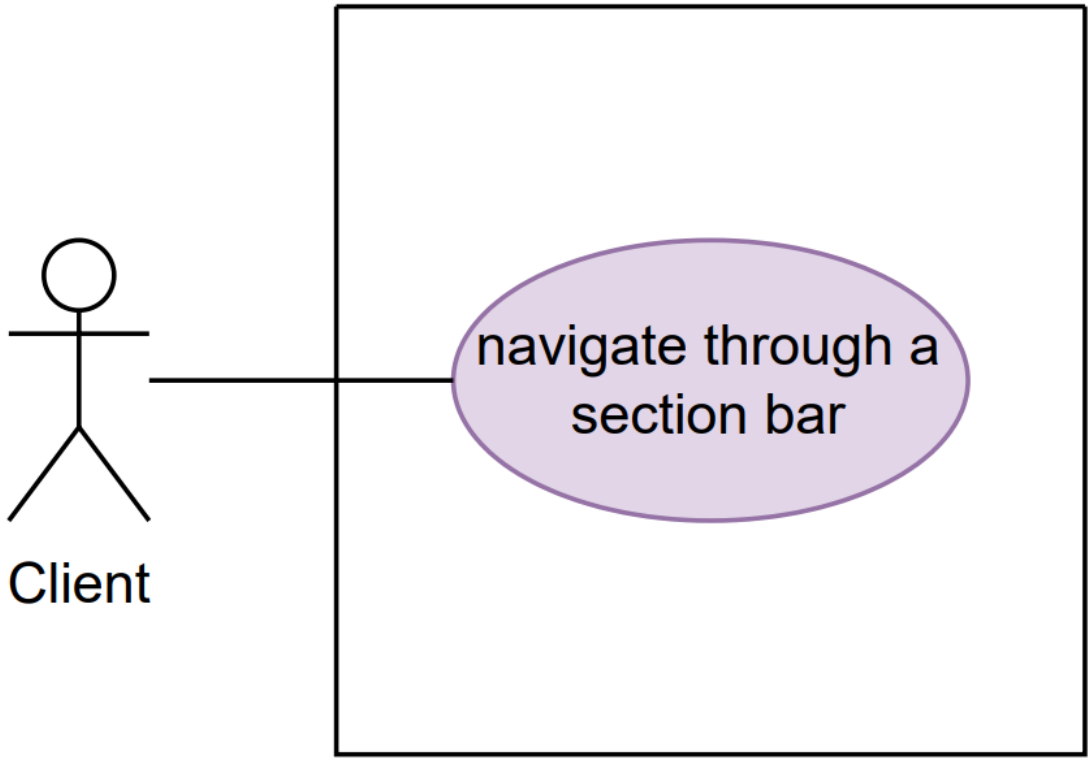
7.3.6 Displaying Moissanite Education



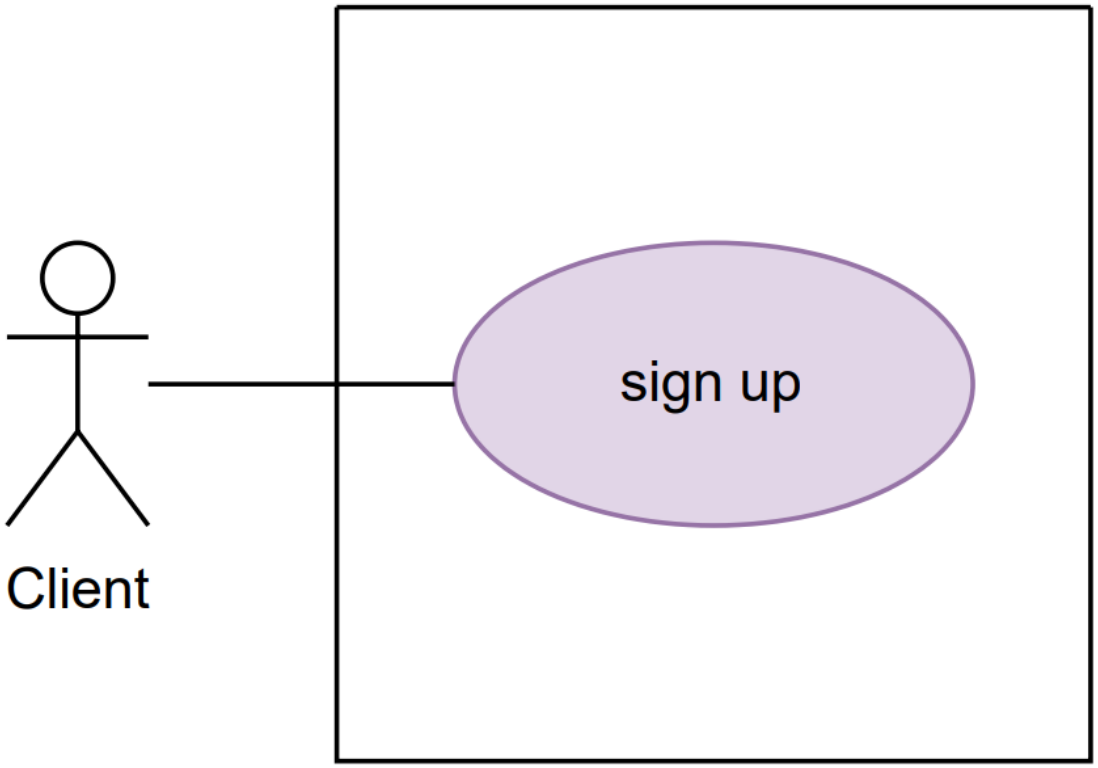
7.3.7 Searching For Products



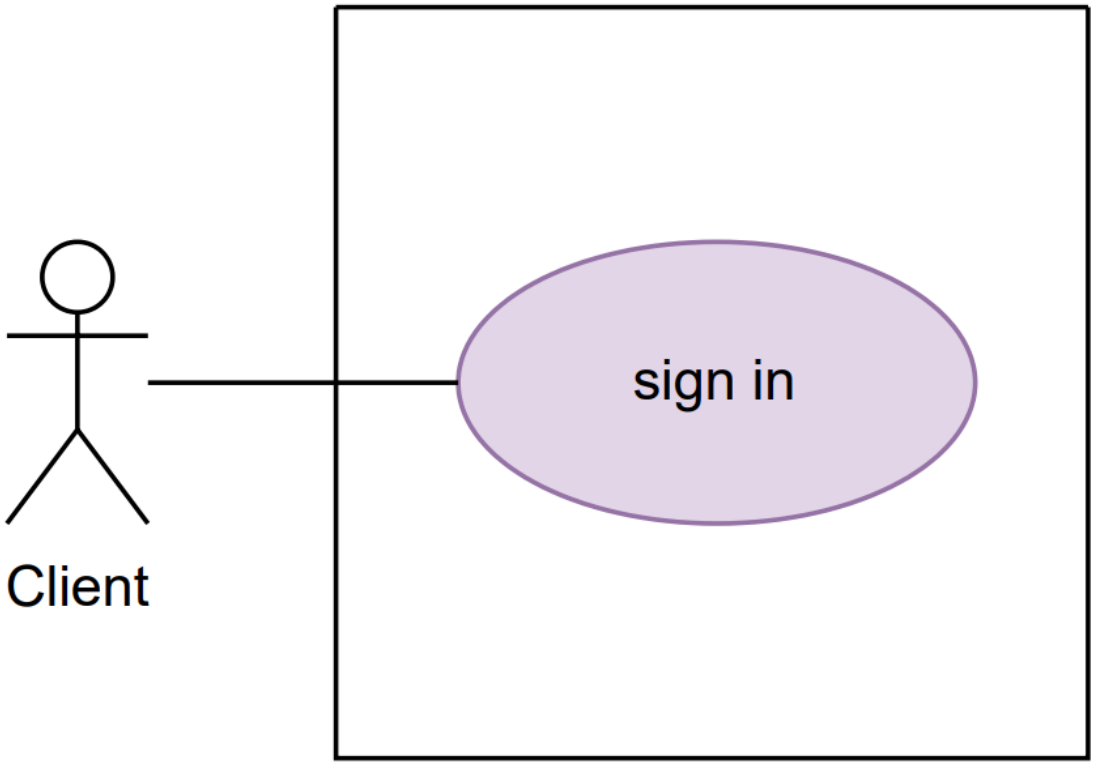
7.3.8 Navigating Through a Section Bar



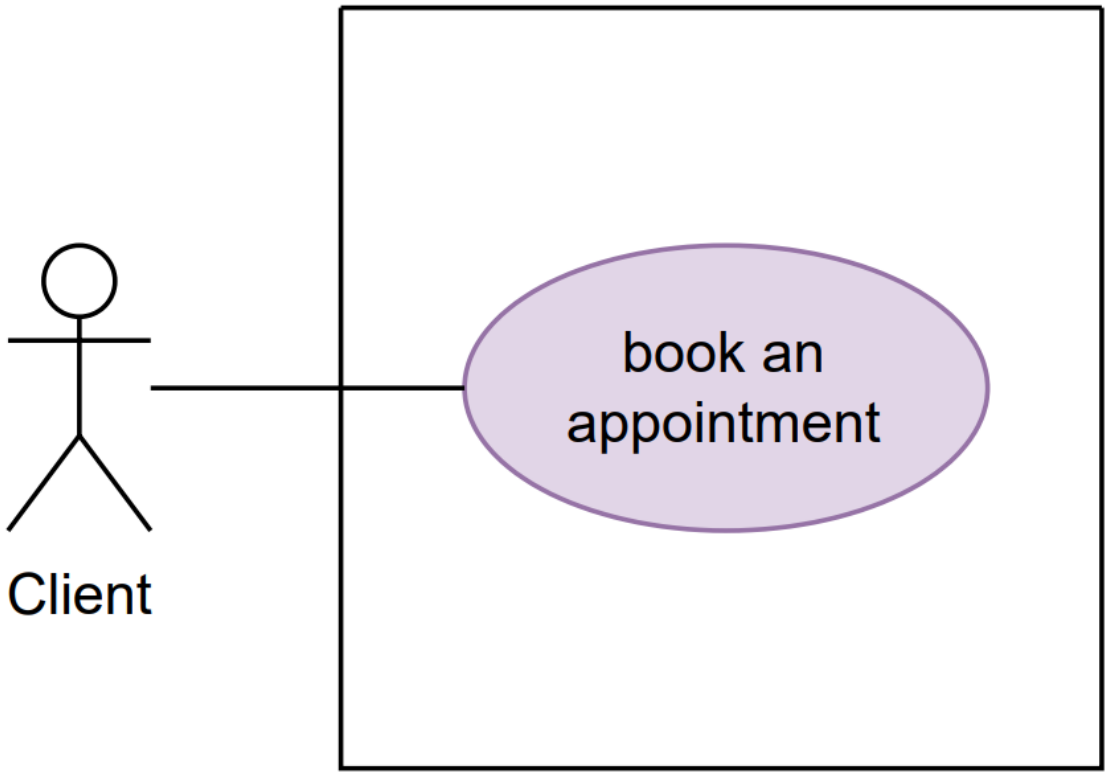
7.3.9 Creating an Account



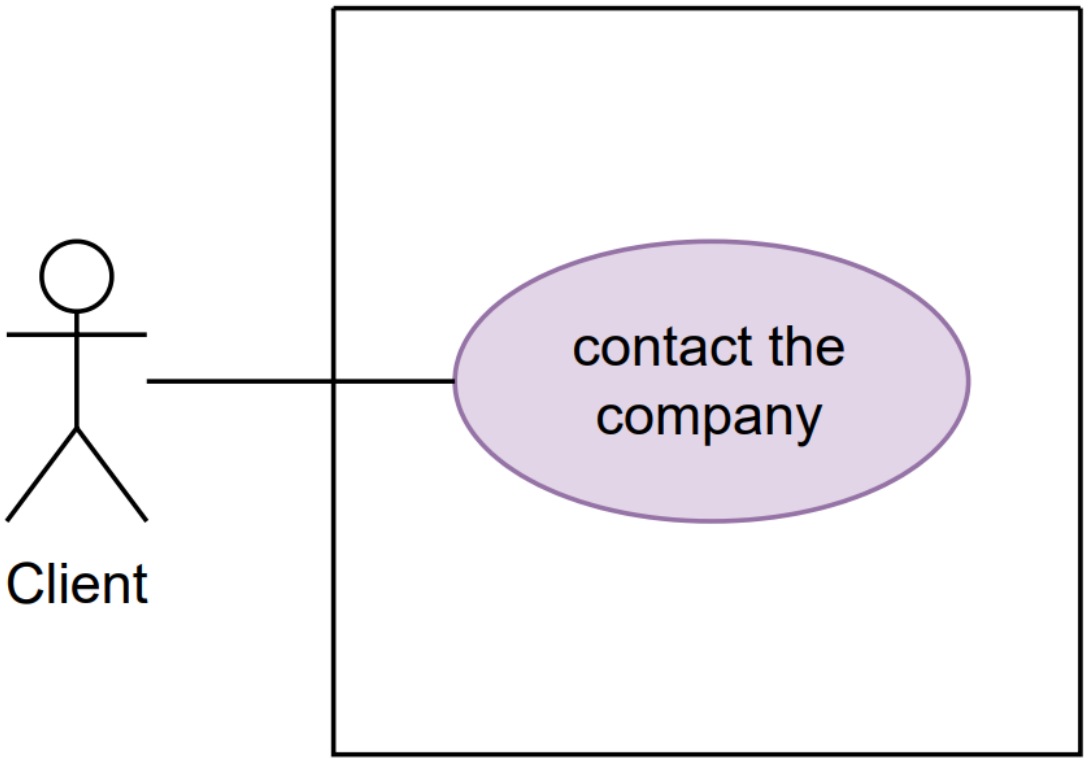
7.3.10 Signing In



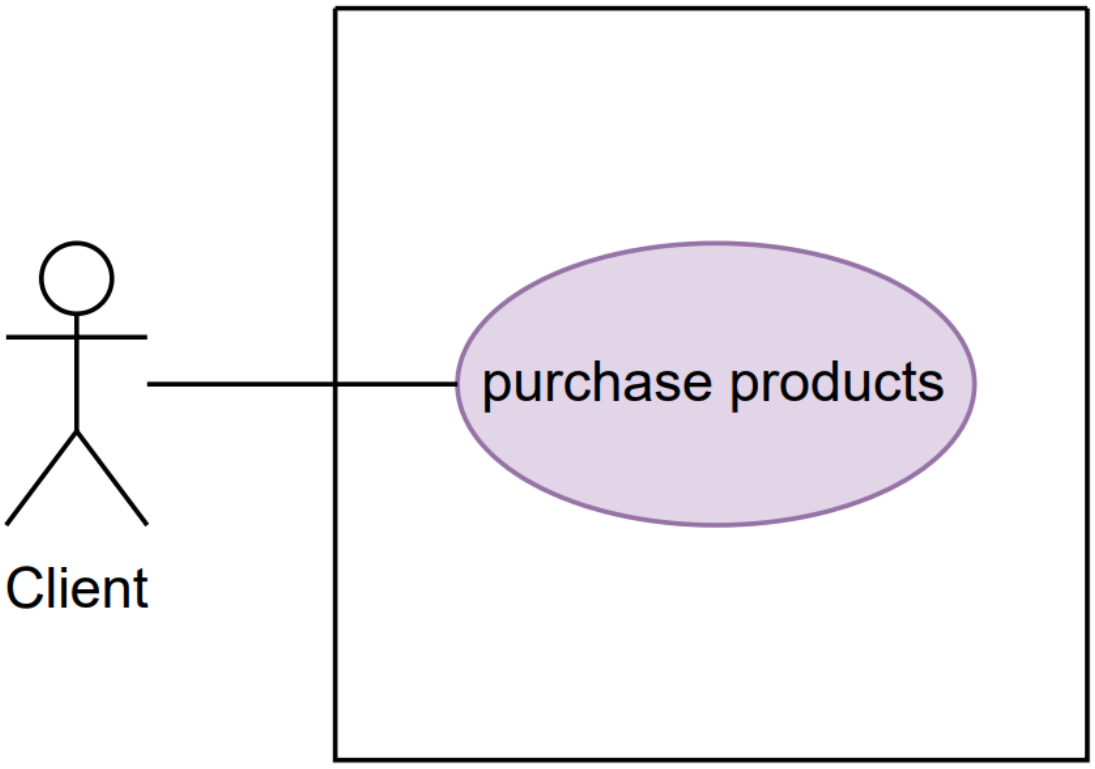
7.3.11 Booking An Appointment



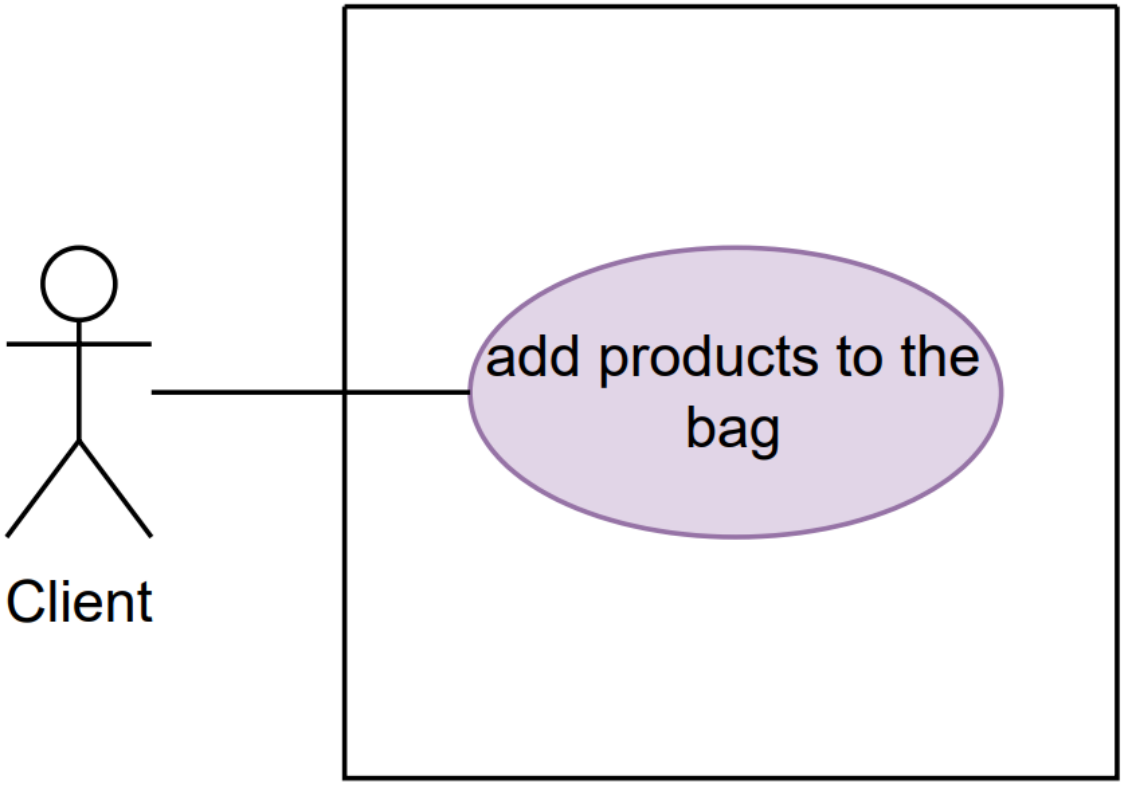
7.3.12 Contacting Us



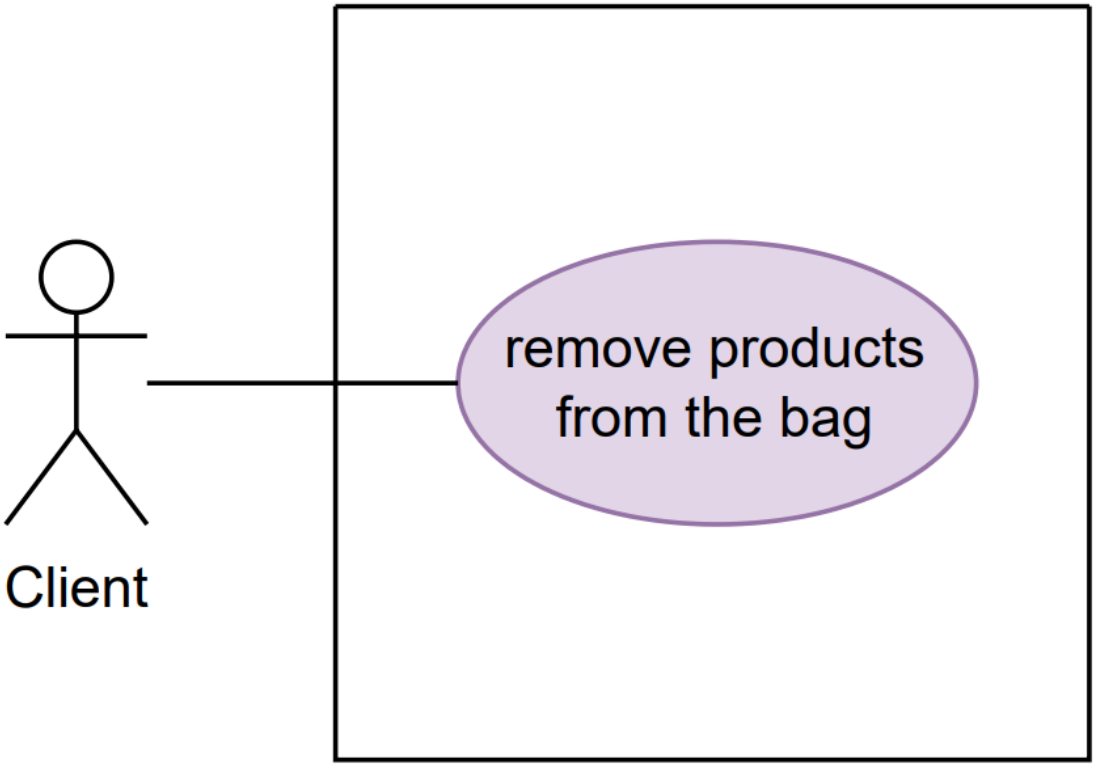
7.3.13 Purchasing Products



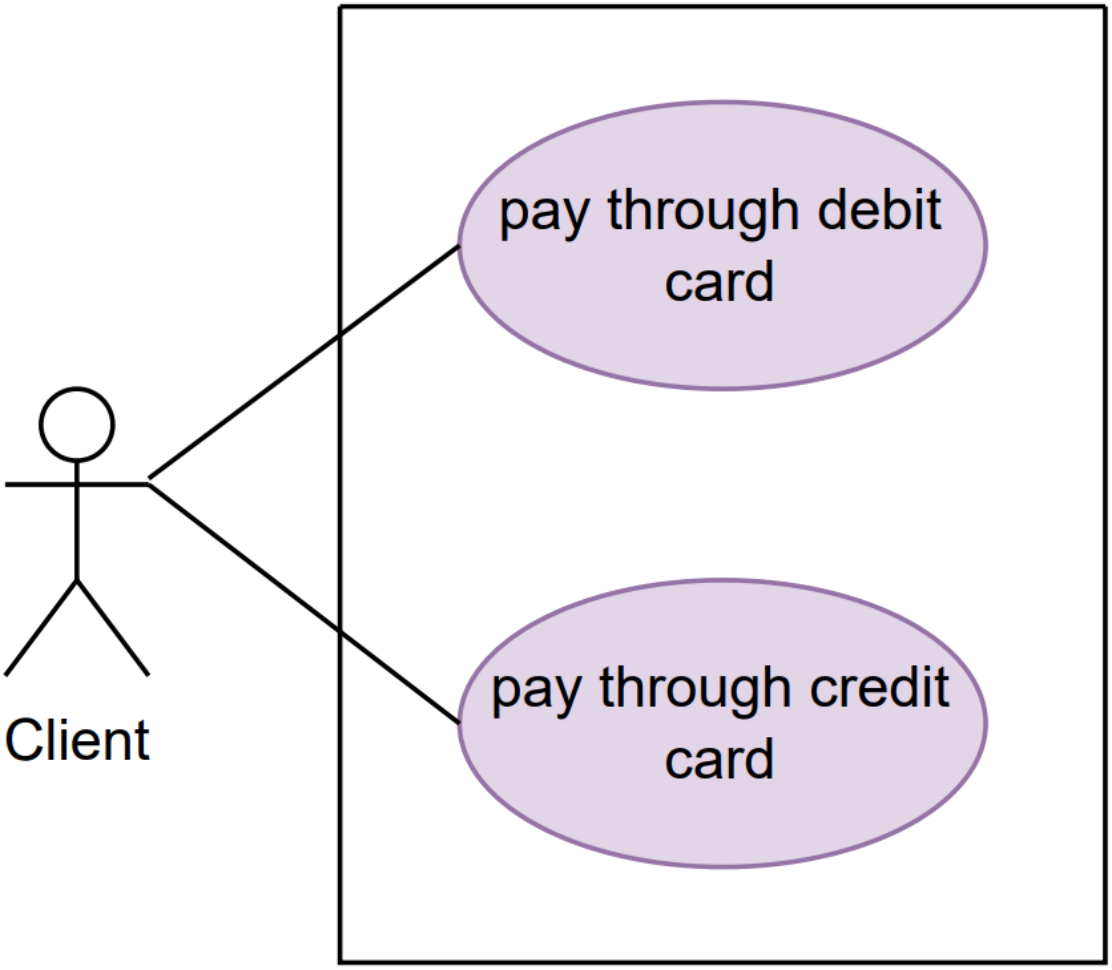
7.3.14 Adding Products For Purchase



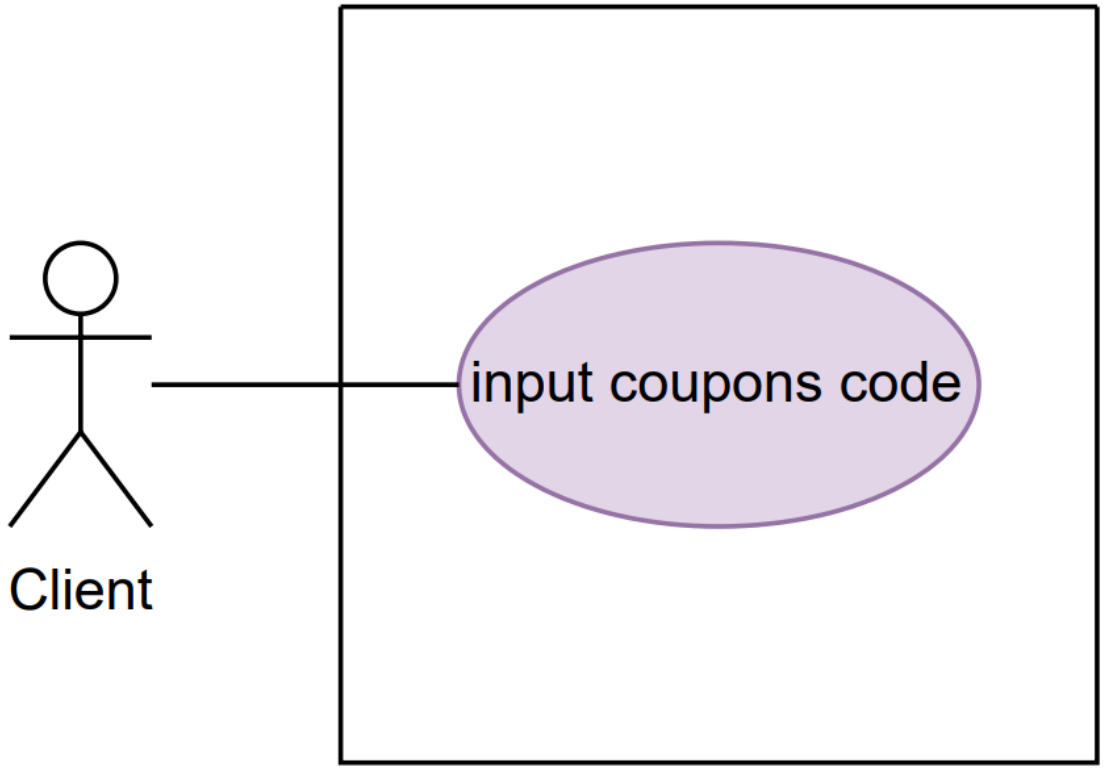
7.3.15 Removing Selected Products



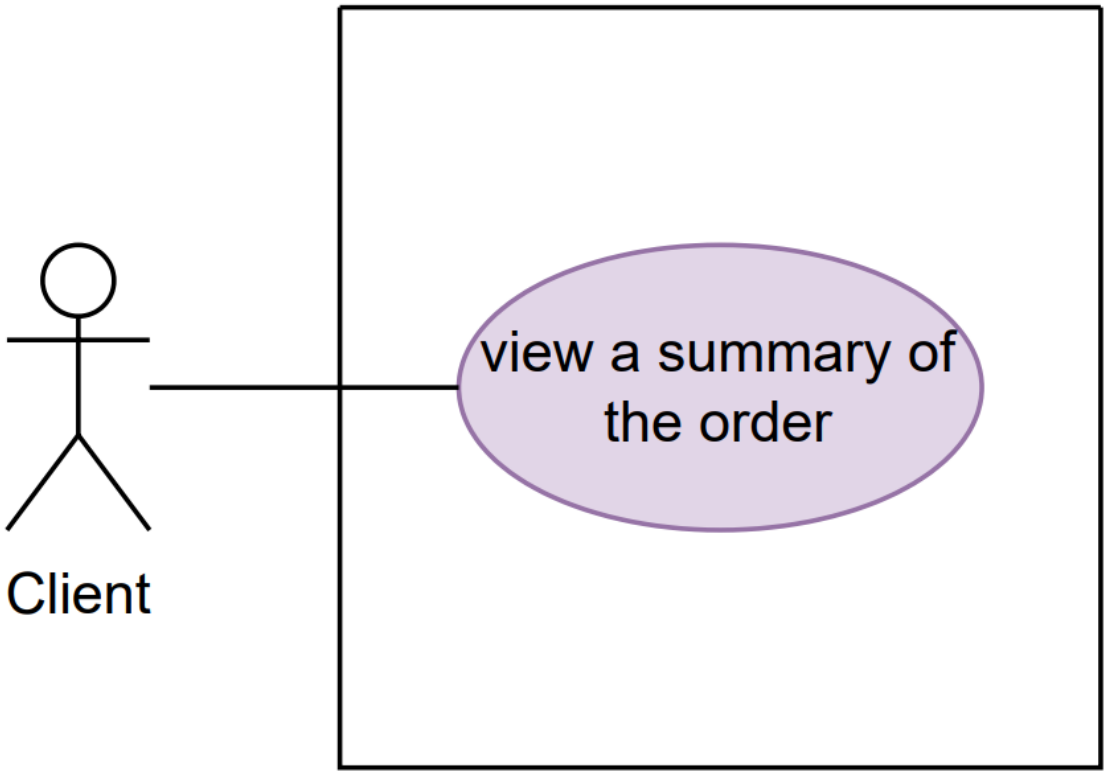
7.3.16 Paying through Debit/Credit



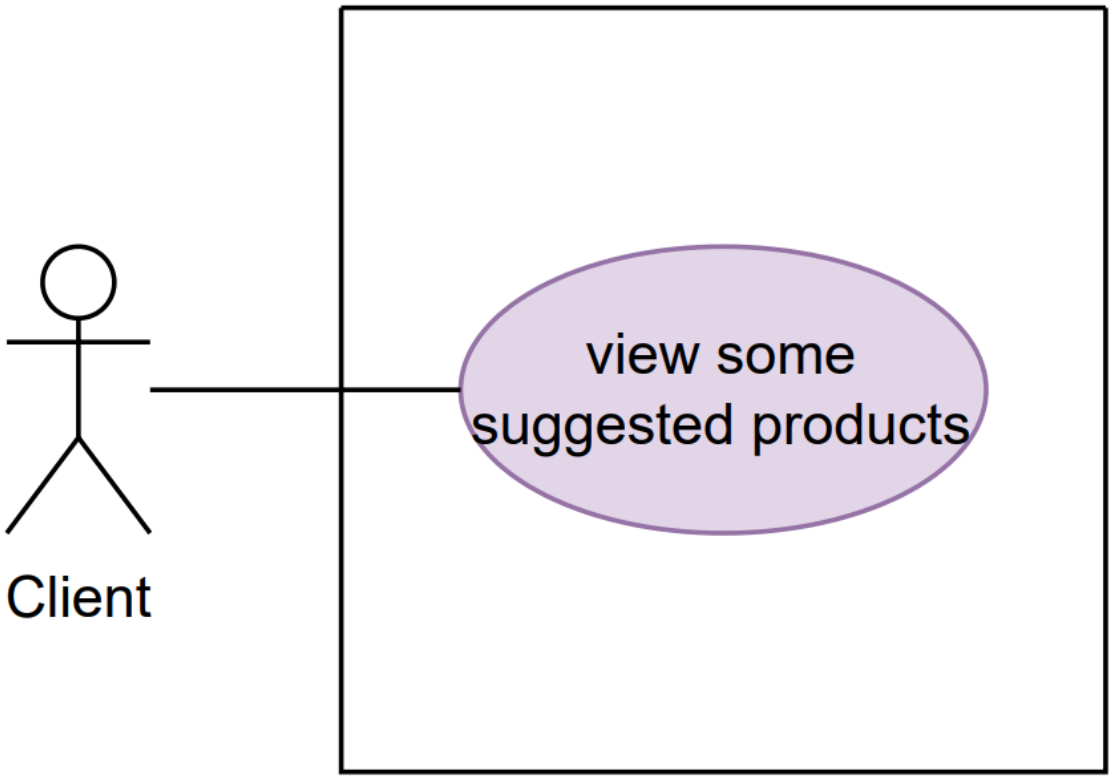
7.3.17 Inputting Coupon code



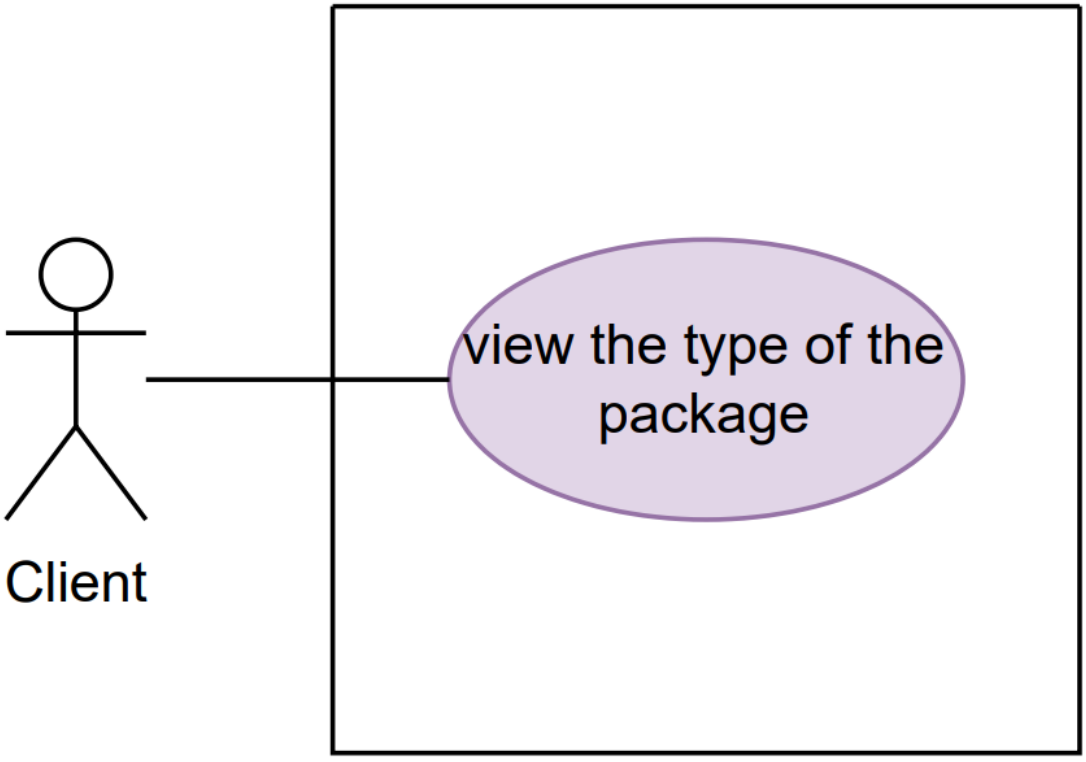
7.3.18 Summarizing order



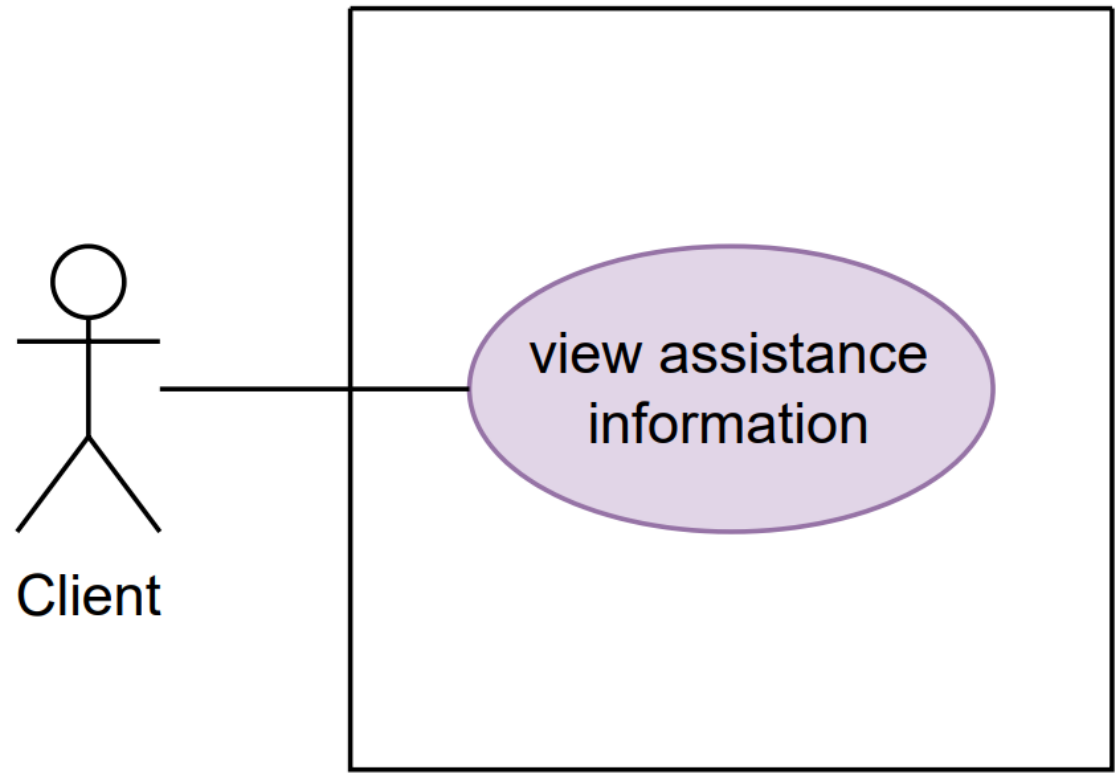
7.3.19 Suggesting Products



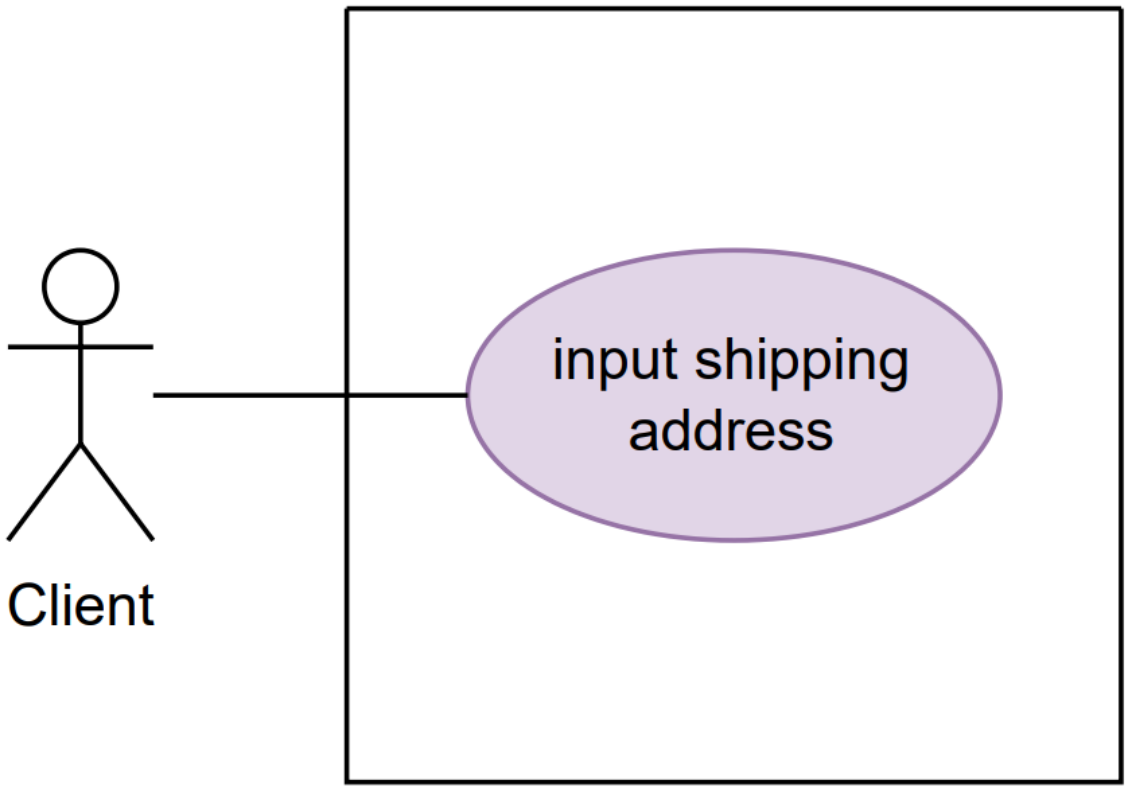
7.3.20 Displaying Package



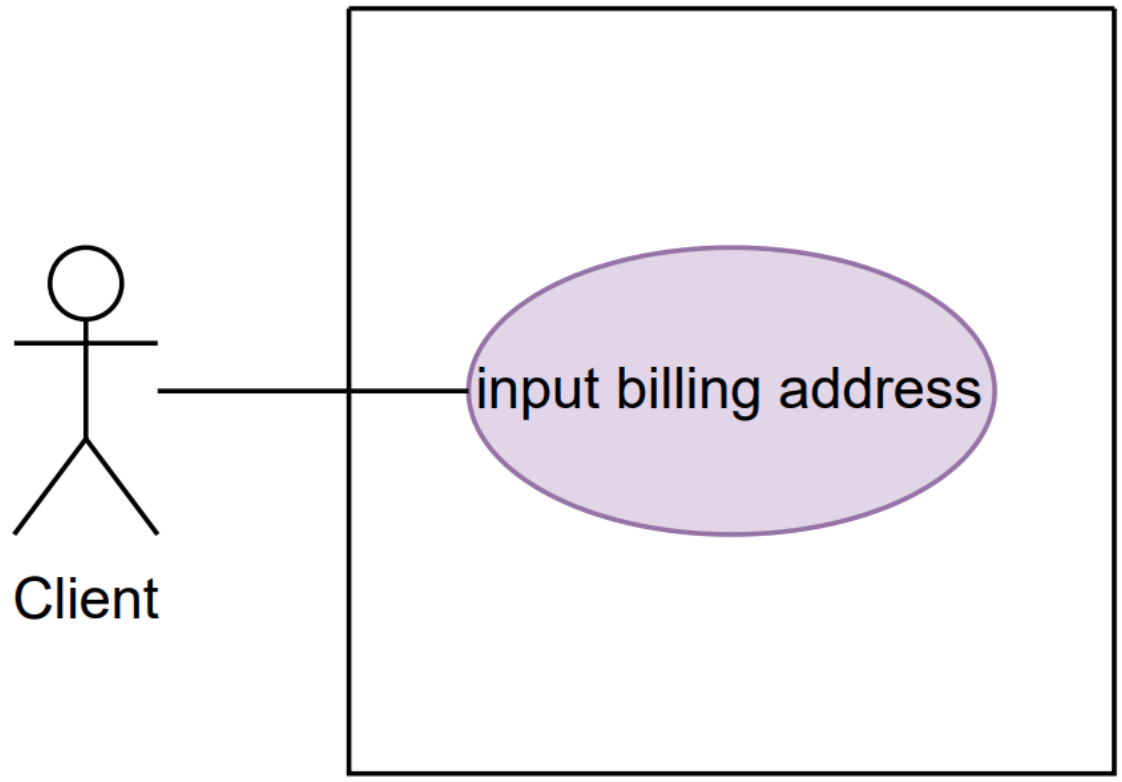
7.3.21 Providing Assistance Information



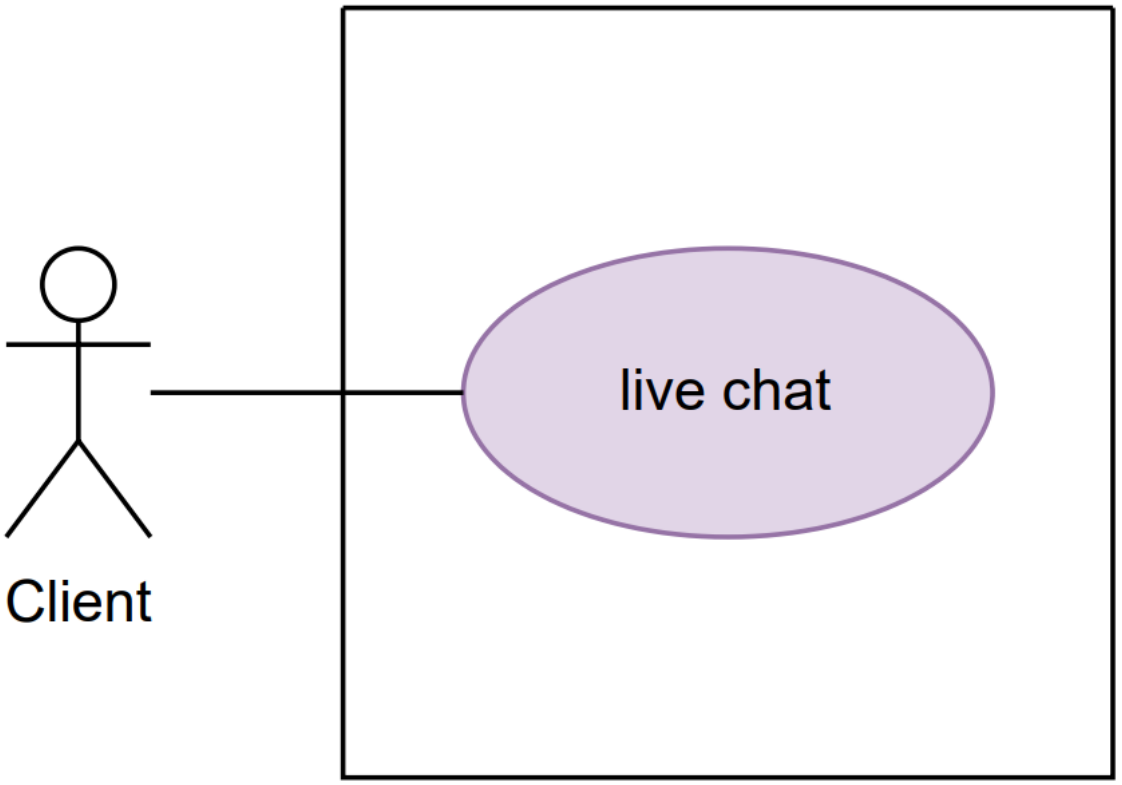
7.3.22 Inputting Shipping Address



7.3.23 Inputting Billing Address



7.3.24 Live Chatting



7.4 ER Diagrams

7.5 Stimulus Response Model

8 System Architecture

8.1 Subsystems

8.2 Components

9 Client's Business

9.1 Before

9.2 After

10 Signatures

11 Index