**Requirements Specifications**

**P02:MinarMarket**

**<team member names & ids>**

| **Student ID** | **Name** |
| --- | --- |
| **25100016** | **Abdul ahad bin ali** |
| **25100017** | **M. Umer Jamil** |
| **25100211** | **Hasan Malik** |
| **25100181** | **Saad Ilyas** |
|  | **Aniqa Aqeel** |

**Table of Contents**

[1.](#_gjdgxs) Introduction 3

[2.](#_30j0zll) System Actors 4

[3.](#_1fob9te) Use Cases 5

[3.1 Use Case Diagrams 5](#_3znysh7)

[3.2 Description of Use Cases 5](#_2et92p0)

[3.2.1 Withdraw cash 5](#_tyjcwt)

[3.2.2 Transfer funds 6](#_3dy6vkm)

[4.](#_1t3h5sf) Class Diagram 7

[4.1 Diagram 7](#_4d34og8)

[4.2 Description 7](#_2s8eyo1)

[5.](#_17dp8vu) Sequence Diagrams 8

[5.1 Use case Name e.g., Withdraw cash 8](#_3rdcrjn)

[5.2 Use case Name e.g., Transfer funds 8](#_26in1rg)

[6.](#_lnxbz9) State Diagrams 9

[6.1 Diagram details 9](#_35nkun2)

[6.2 Diagram 9](#_1ksv4uv)

[7.](#_44sinio) Non-functional Requirements / Quality Attributes 10

[8.](#_2jxsxqh) Who Did What? 11

[9.](#_z337ya) Review checklist 11

# Introduction

In the rapidly evolving world of e-commerce, the relationship between buyers and sellers is continuously being redefined. Traditional online marketplaces tend to operate in a seller-centric manner, where sellers list their products, and buyers browse through these listings to make their purchases. While this model has proven effective in many scenarios, it often leaves buyers with limited options when they have specific needs that don’t align perfectly with the available listings. As consumers increasingly demand personalization and convenience, there is a growing need for a marketplace that addresses this limitation and fosters a more collaborative relationship between buyers and sellers.

Our project introduces an innovative approach to online marketplaces by creating a platform that not only allows sellers to list their products but also gives buyers the power to post their specific requirements. This dual functionality transforms the marketplace into a more dynamic and interactive ecosystem where buyers actively express their needs, and sellers can respond by offering products that meet those exact requirements. This model reduces the gap between supply and demand, enabling sellers to more effectively target interested buyers and ensuring buyers find the products that truly fit their preferences.

The core objective of this project is to design and implement a marketplace that enhances the traditional e-commerce experience. Buyers will no longer be confined to searching through predefined listings but can instead list the products or services they are looking for. Sellers, in turn, will have visibility into these buyer requests and can engage directly with potential customers by offering relevant products or negotiating terms that meet the buyer’s expectations. This two-way interaction fosters a marketplace that is more responsive, transparent, and efficient.

A key focus of our project is to simplify and streamline communication between buyers and sellers. The platform will feature an intuitive user interface that allows both parties to post, search, and communicate with ease. Buyers will be able to track the offers they receive in response to their requests, compare different sellers, and make informed purchasing decisions based on personalized recommendations. Sellers will benefit from real-time notifications of buyer requests that match their inventory, allowing them to act quickly to meet demand. This buyer-driven interaction introduces a new level of personalization and convenience, benefiting both parties involved in the transaction.

Our target audience for this platform includes individual consumers, small businesses, and larger enterprises. Individual consumers will appreciate the ability to request highly specific products, while businesses can leverage the platform to source bulk orders or specialized items. Additionally, the platform can serve niche markets where product availability may be limited, empowering buyers with greater choice and sellers with direct access to a motivated customer base.

Furthermore, our marketplace is designed with scalability and flexibility in mind. As the platform grows, we plan to incorporate advanced features such as AI-driven product matching, where algorithms analyze buyer requests and suggest potential matches from a seller’s inventory. This feature will streamline the offer-making process for sellers and make it easier for buyers to receive relevant product suggestions. The system will also allow for future integrations with payment gateways, shipment tracking, and customer review systems to create a comprehensive and seamless e-commerce experience.

Ultimately, this project aims to revolutionize the traditional marketplace model by making it more buyer-driven, interactive, and efficient. By bridging the gap between buyer needs and seller offerings, our platform will create a more engaging and fulfilling experience for all users. This approach will not only improve transaction success rates but also foster stronger relationships between buyers and sellers, setting a new standard for online commerce in the modern digital era.

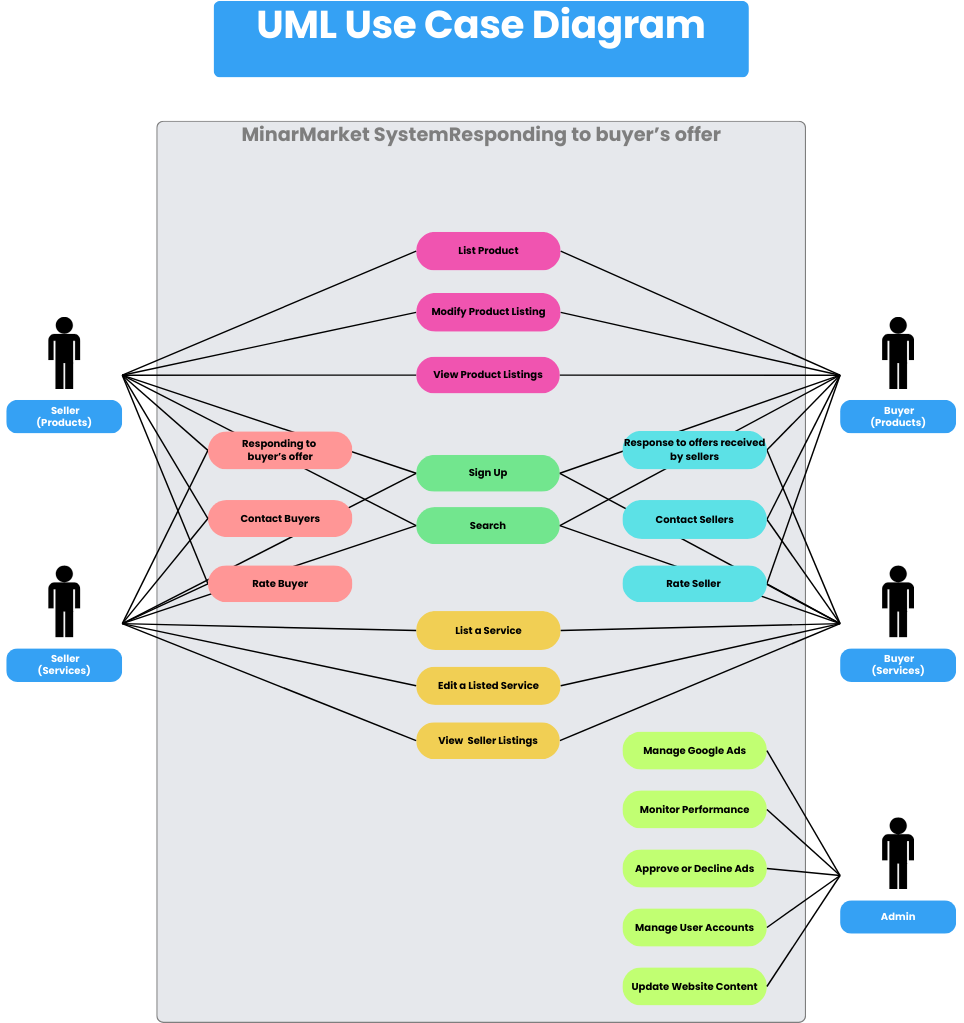
# System Actors

| **Actor Name** | **Description** |
| --- | --- |
| Seller (Services) | This user is one of the two types of sellers on this platform. This user will sell his/her services at a per-hour rate. This user can also consider a better offer, either on the lower or higher ends. |
| Seller (Products) | This is the second type of seller. This user will sell something tangible (i.e., a product) for a fixed price. This user can also consider some flexibility on the price. |
| Customer (Services) | This user is one of the two types of customers on this platform. In addition to buying the services from the seller, this user can also list the services needed independently. This type of user will provide a range of their budget, i.e., a low and a high-end. The sellers can approach these customers directly with a better offer. |
| Customer (Goods) | This is the second type of customer. This type of user can either buy from the listings posted by sellers or list for specifications and a budget for the product they are looking for. This type of user can provide a budget range or a fixed amount. Sellers can approach these customers if they wish to sell at the customer’s price. |
| System Administrator | The System Administrator will maintain the system, which includes managing user accounts and ensuring smooth operational running of the website. Moreover, the system administrator will also work as a moderator, overlooking all the listings being posted and will verify the authenticity of the listing. Look out for any glitches or unexpected errors, and update the website. |

# 

# Use Cases

## Use Case Diagrams



## Description of Use Cases

### 

### Buyers can search for the product or service either by name or add id.

| **Identifier** | | UC-001 |
| --- | --- | --- |
| **Purpose** | | Buyers can search for products or services listed by multiple sellers on the platform. |
| **Pre-conditions** | | The buyer is logged into the platform.  There are products or services listed by various sellers on the platform. |
| **Post-conditions** | | The buyer is able to view a list of products matching the search criteria.  The products displayed belong to multiple sellers. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer navigates to the search bar, typically located at the top of the page. | |
| **2** | The buyer enters a search keyword or add id to find specific products or services. | |
| **3** | The system suggests search terms or auto-complete options as the buyer types. | |
| **4** | The buyer selects a search term from the suggestions or proceeds with the entered search term. | |
| **5** | The buyer can apply filters (e.g., time, price range, experience, etc.) to refine the search results. | |
| **6** | The system processes the search query and retrieves a list of products or services from various sellers based on the search criteria. | |
| **7** | The buyer can sort the search results (e.g., by relevance, price, ratings). | |
| **8** | The buyer scrolls through the list of products or services, viewing details such as price, seller information, and ratings. | |
| **9** | The buyer selects a service to view more details or continues browsing other products or services. | |
| **10** | The use case ends when the buyer completes the search or selects a product or a service for further action (e.g., viewing service details or adding it to the cart). | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 4, if the buyer does not find a relevant auto-suggestion, they can modify or refine the search term manually. | |
| 2 | In step 6, the buyer may choose to clear all filters and perform an unfiltered search to see a broader range of products. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 7, if no products or services match the search criteria, the system displays a "No products/services found" message and suggests alternative products or search terms. | |
| 2 | If the system experiences an error during the search process, an error message is displayed, and the buyer can retry the search. | |

### 

### Buyers can list the products they want with the requirement details.

| **Identifier** | | UC-002 |
| --- | --- | --- |
| **Purpose** | | Buyers can list a desired product on the platform by providing specific requirement details, such as budget, condition of the product, and other relevant preferences. |
| **Pre-conditions** | | The user is logged into the platform.  The platform supports buyer requests for listing desired products. |
| **Post-conditions** | | The buyer’s product request is successfully listed on the platform.  Sellers are able to view the buyer's request and respond with product offers that match the buyer's requirements. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer navigates to the "Request a Product" or "Post a Requirement" section. | |
| **2** | The system displays a form for the buyer to fill out the product requirements. | |
| **3** | The buyer specifies the product type or model they are looking for. | |
| **4** | The buyer provides a budget range for the desired product. | |
| **5** | The buyer specifies the preferred condition of the product (e.g., new, used, refurbished). | |
| **6** | The buyer can provide additional details, such as preferred brand, size, color, or any special requirements. | |
| **7** | The system may prompt the buyer to upload reference images or examples of the desired product. | |
| **8** | The buyer reviews all the entered details and confirms the listing. | |
| **9** | The system processes the request and posts it on the platform, making it visible to sellers. | |
| **10** | The use case ends when the buyer’s request is successfully listed. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 5, if the buyer is unsure about the budget, they can leave the field open, and the system will notify sellers to provide offers within a wide range. | |
| 2 | In step 7, if the buyer does not have specific additional details, they can leave the optional fields empty and proceed with a general request. | |
| 3 | In step 9, if the buyer is not ready to submit the request, they can save it as a draft and return to complete it later. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 10, if the system encounters an error while posting the request, the buyer is notified and can retry submitting the request. | |
| 2 | If the platform is unable to match sellers with the buyer’s specific requirements, the system may suggest alternative sellers or similar products available on the platform. | |

### 

### Buyers can respond to the offers received by sellers

| **Identifier** | | UC-003 |
| --- | --- | --- |
| **Purpose** | | Buyers can view and respond to offers from sellers based on their product requests. |
| **Pre-conditions** | | The buyer has listed a product request on the platform.  Sellers have submitted offers in response to the buyer's request. |
| **Post-conditions** | | The buyer successfully responds to the seller's offers (e.g., accepting, rejecting, or negotiating).  The status of the offer is updated accordingly in the system. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer navigates to the "My Requests" or "Offers Received" section. | |
| **2** | The buyer selects the relevant product request for which they have received seller offers. | |
| **3** | The system displays a list of offers from various sellers, including details such as price, product condition, and seller ratings. | |
| **4** | The buyer clicks on a specific offer to view detailed information about the product and seller’s terms. | |
| **5** | The buyer reviews the offer, considering factors like price, product details, and seller reputation. | |
| **6** | The buyer can respond by selecting one of the following options:   * Accept the offer and proceed with the purchase. * Reject the offer and provide feedback or a reason for rejection. * Initiate a negotiation by suggesting alternative terms (e.g., lower price or different product conditions). | |
| **7** | If the buyer accepts the offer, they proceed to the payment and shipping steps as per the platform's standard process. | |
| **8** | If the buyer negotiates, the system sends the counteroffer to the seller, and the process continues until both parties reach an agreement or the buyer/seller declines further negotiation. | |
| **9** | If the buyer rejects the offer, the system updates the status and notifies the seller. | |
| **10** | The buyer can repeat steps 4 to 9 for other offers if multiple sellers have responded. | |
| **11** | The use case ends when the buyer has responded to all offers or completed the transaction with a seller. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 6, the buyer can choose to mark an offer for later review if they are not ready to respond immediately. | |
| 2 | In step 8, if the seller rejects the counteroffer, the buyer can choose to make another counteroffer or reject the offer entirely. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 7, if the system encounters an issue during the acceptance or payment process, an error message is displayed, and the buyer can retry or contact support. | |
| 2 | If the seller fails to respond to a negotiation in step 8 within a specified time frame, the system notifies the buyer and suggests reviewing other offers. | |

### 

### Buyers can reach out to sellers in order to ask more about a product

| **Identifier** | | UC-004 |
| --- | --- | --- |
| **Purpose** | | Buyers can be shown or redirected to some contact platform (email, message etc) so they can contact sellers to ask for more information about a product. |
| **Pre-conditions** | | The buyer is logged into the platform.  The product of interest is listed by a seller. |
| **Post-conditions** | | The buyer’s inquiry is sent to the seller, and the seller can respond with additional information.  A communication channel is established between the buyer and the seller. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer navigates to the product listing they are interested in. | |
| **2** | The buyer clicks on the product to view its details and information provided by the seller. | |
| **3** | The buyer scrolls down to find the "Contact Seller" or "Get Contact Info" button. | |
| **4** | The buyer clicks on the "Contact Seller" button to open a communication interface (e.g., whatsapp, email etc). | |
| **5** | The system displays a form or chat interface of a whatsapp or email etc where the buyer can enter their question or request for more information (e.g., product specifications, shipping details, etc.). | |
| **6** | The buyer types their inquiry into the provided text box, asking for clarification or additional details about the product. | |
| **7** | The buyer reviews the message and sends it. | |
| **8** | The seller is notified of the buyer's message. | |
| **9** | The seller receives the inquiry and can respond through the same communication platform.. | |
| **10** | The use case ends when the buyer has received the required information or the conversation is closed. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 5, if the buyer decides not to ask a question, they can close the inquiry form and continue browsing the product. | |
| 2 | In step 9, if the seller does not respond within a certain time frame, the buyer can send a follow-up message or browse other products. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 7, if the system fails to send the message due to a technical error, the buyer is notified and can retry sending the inquiry. | |
| 2 | In step 9, if the seller’s account is inactive or unavailable, the buyer is notified, and the inquiry cannot be completed. | |
| 3 | In case of communication issues (e.g., the seller is unresponsive), the system may prompt the buyer with an option to contact support or look for alternative sellers. | |

### 

### Buyers can update or delete their existing listing showing the previous price in case of price update.

| **Identifier** | | UC-005 |
| --- | --- | --- |
| **Purpose** | | Buyers can modify or remove their existing product request listings on the platform. If the buyer updates the price then the previous price should also be shown along with the new price. |
| **Pre-conditions** | | The buyer is logged into the platform.  The buyer has previously created a product request listing. |
| **Post-conditions** | | The listing is updated with the new information or is removed from the platform.  If deleted, the listing is no longer visible to sellers. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer navigates to the "My Listings" or "My Requests" section. | |
| **2** | The buyer views a list of their existing product request listings. | |
| **3** | The buyer selects the listing they wish to update or delete. | |
| **4** | To update the listing, the buyer clicks on the "Edit" button next to the selected listing. | |
| **5** | The system displays the product request form with the current details pre-filled. | |
| **6** | The buyer modifies the necessary fields, such as the budget, product condition, or other requirement details. | |
| **7** | The buyer reviews the updated information and confirms the changes by clicking the "Update" or "Save" button. | |
| **8** | The system processes the changes and updates the listing on the platform, making the new details visible to sellers along with the previous price in case the price is also changed. | |
| **9** | If the buyer wants to delete the listing, they click the "Delete" button next to the listing. | |
| **10** | The system prompts the buyer with a confirmation message asking if they are sure about deleting the listing. | |
| **11** | The buyer confirms the deletion. | |
| **12** | The system removes the listing from the platform, and it is no longer visible to sellers. | |
| **13** | The use case ends when the listing is successfully updated or deleted. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 4, if the buyer accidentally selects the wrong listing, they can cancel the update or deletion and return to the "My Listings" section. | |
| 2 | In step 6, if the buyer decides not to update the listing, they can discard the changes and exit the editing mode. | |
| 3 | In step 11, if the buyer decides not to delete the listing, they can cancel the action and keep the listing active. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 8, if the system encounters an error while updating the listing, the buyer is notified and can retry the update. | |
| 2 | In step 12, if the system fails to delete the listing due to a technical issue, the buyer is notified, and the listing remains active until the issue is resolved. | |

### View all the listed services

| **Identifier** | | UC-006 |
| --- | --- | --- |
| **Purpose** | | The seller views all the services they have listed on the platform to track, manage, or update their listings. |
| **Pre-conditions** | | The seller is logged into their account.  The seller has previously listed at least one service on the platform. |
| **Post-conditions** | | The seller is able to see all of their active, pending, and past service listings and take further actions such as editing, updating, or deleting them. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller navigates to the "My Services" section from the dashboard. | |
| **2** | The system displays a list of all services the seller has posted, categorized as active, pending, or past listings. | |
| **3** | The seller reviews the list of services. | |
| **4.** | The seller clicks on any specific service to view its details (e.g., description, price, availability). | |
| **5.** | The system displays the details of the selected service. | |
| **6.** | The seller may choose to edit, update, or delete the service from this screen. | |
| **7.** | If no action is taken, the seller returns to the full list of services. | |
| **8.** | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 2, the seller may choose to filter the service listings by status (e.g., active only, past services). | |
| **Step #** | **Exception Paths** | |
| 1. | In step 2, if the seller has not listed any services yet, the system displays a “No Services Listed” message, and the seller may be prompted to create a new service listing. | |

### List a service for selling

| **Identifier** | | UC-007 |
| --- | --- | --- |
| **Purpose** | | The seller lists a service they wish to offer, providing details such as service description, pricing, and availability. |
| **Pre-conditions** | | The seller is logged into their account.  The seller has service details (description, pricing, availability, etc.) ready for input. |
| **Post-conditions** | | The service is successfully listed, making it visible to potential buyers on the platform. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The seller navigates to the "List a Service" page. | |
|  | The system prompts the seller to enter the service details (title, description, hourly rate, availability). | |
|  | The seller fills in the required service information. | |
| **4.** | The system asks the seller to upload any supporting documents or images (if applicable). | |
| **5.** | The seller uploads relevant media (optional). | |
| **6.** | The system prompts the seller to review the listing details. | |
| **7.** | The seller reviews and confirms the listing. | |
| **8.** | The system submits the service listing and displays a confirmation message. | |
| **9.** | The system makes the listing available on the platform for buyers to view. | |
| **10.** | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 7, the seller may choose to edit the service details before final confirmation. | |
| 2. | In step 8, the seller may cancel the listing submission and return to the service listing page without saving changes. | |
| **Step #** | **Exception Paths** | |
| 1. | In step 5, if there is an issue with the media upload (e.g., invalid format or file size), the system displays an error, and the seller must re-upload the file before proceeding to step 6. | |

### Send Proposals to Buyers for Service Requests

| **Identifier** | | UC-008 |
| --- | --- | --- |
| **Purpose** | | The seller sends a proposal in response to a buyer’s service request posted on the platform. |
| **Pre-conditions** | | The seller is logged into their account.  The buyer has posted a request for a service that matches the seller’s offerings. |
| **Post-conditions** | | The proposal is successfully sent to the buyer, and the buyer is notified of the seller’s offer. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The seller navigates to the "Service Requests" section. | |
|  | The system displays a list of buyer service requests relevant to the seller’s offerings. | |
|  | The seller selects a buyer request to send a proposal. | |
| **4.** | The system displays the details of the buyer’s request. | |
| **5.** | The seller clicks on the "Send Proposal" button. | |
| **6.** | The system prompts the seller to input proposal details (price, scope of service, time frame). | |
| **7.** | The seller fills in the proposal details and submits the proposal. | |
| **8.** | The system asks the seller to confirm the submission. | |
| **9.** | The seller confirms the proposal submission. | |
| **10.** | The system sends a notification to the buyer that the seller has sent a proposal. | |
| **11.** | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 3, the seller may filter service requests based on specific criteria such as location, budget, or timeline. | |
| 2. | In step 9, the seller may choose to edit the proposal before confirming submission. | |
| **Step #** | **Exception Paths** | |
| 1. | In step 7, if the seller’s input is invalid (e.g., price outside the buyer’s budget), the system displays an error and prompts the seller to revise the proposal details before proceeding to step 9. | |

### Negotiate Terms of Service with Buyer

| **Identifier** | | UC-009 |
| --- | --- | --- |
| **Purpose** | | The seller negotiates the terms of the service (price, delivery time, etc.) with the buyer through the platform. |
| **Pre-conditions** | | The seller is logged into their account.  The buyer has received a proposal from the seller or has requested a negotiation.  Both parties are able to communicate through the platform. |
| **Post-conditions** | | A mutually agreed-upon proposal is submitted, or the negotiation is terminated by either party. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer reviews the seller’s proposal and sends a counter-offer or requests changes. | |
| **2** | The seller receives a notification of the buyer’s response. | |
| **3** | The seller views the buyer’s requested changes (e.g., new price, delivery time). | |
| **4.** | The seller clicks on the "Negotiate" button. | |
| **5.** | The system prompts the seller to input revised terms (price, service scope, time frame). | |
| **6.** | The seller revises the proposal and submits the new terms. | |
| **7.** | The system sends the updated proposal to the buyer. | |
| **8.** | The buyer receives the updated proposal and reviews it. | |
| **9.** | The negotiation continues until both parties reach an agreement or either party cancels the negotiation. | |
| **10.** | Once agreed, the system finalizes the proposal and updates the service request status. | |
| **11.** | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 3, the seller may reject the buyer's counter-offer without proposing new terms, terminating the negotiation. | |
| **Step #** | **Exception Paths** | |
| 1. | In step 9, if either party cancels the negotiation at any point, the system notifies the other party, and the negotiation is terminated. | |

### Update Availability or Details of a Posted Service

| **Identifier** | | UC-010 |
| --- | --- | --- |
| **Purpose** | | The seller updates the availability, pricing, or other details of a service that they have already posted on the platform. The system should preserve the last price and update it accordingly wherever required. |
| **Pre-conditions** | | The seller is logged into their account.  The seller has an existing service listed on the platform.  The service is still active and available for editing. |
| **Post-conditions** | | The updated service details are reflected in the listing and are visible to potential buyers. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller navigates to the "My Services" section. | |
| **2** | The system displays the list of services posted by the seller. | |
| **3** | The seller selects the service they wish to update. | |
| **4.** | The system displays the current details of the selected service. | |
| **5.** | The seller clicks on the "Edit" button. | |
| **6.** | The system prompts the seller to update the service details (e.g., availability, price, description). | |
| **7.** | The seller updates the desired fields. | |
| **8.** | The system asks the seller to review the updated service details. | |
| **9.** | The seller confirms and submits the changes. | |
| **10.** | The system updates the service listing and displays a confirmation message. | |
| **11.** | The updated details are made visible to buyers. | |
| **12.** | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 7, the seller may choose to revert the updates before submitting the changes, returning the listing to its original state. | |
| 2. | In step 9, the seller may cancel the changes and exit without saving, returning to the "My Services" section. | |
| **Step #** | **Exception Paths** | |
| 1. | In step 5, if the service is no longer active (e.g., expired or deactivated), the system displays an error message, and the seller cannot proceed with the update. | |

### Search for Buyer Listings by Sellers

| **Identifier** | | UC-011 |
| --- | --- | --- |
| **Purpose** | | The seller searches for listings made by buyers to find relevant service or product requests and approach potential clients. |
| **Pre-conditions** | | The seller is logged into their account.  Buyers have created listings with product/service requests.  The seller has access to a search or filter function on the platform. |
| **Post-conditions** | | The seller is able to view relevant buyer listings, select those of interest, and potentially send proposals or messages to the buyers. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1.** | The seller navigates to the "Search Buyer Listings" section. | |
| **2** | The system displays a search bar and filters (e.g., service type, budget, location). | |
| **3** | The seller enters search criteria or selects filters based on their preferences. | |
| **4.** | The system processes the input and displays relevant buyer listings matching the criteria. | |
| **5.** | The seller reviews the list of buyer service/product requests. | |
| **6.** | The seller selects a specific buyer listing to view more details. | |
| **7.** | The system displays the details of the buyer’s listing (e.g., service requirements, budget, deadline). | |
| **8.** | The seller may choose to send a proposal or message to the buyer regarding the listing. | |
| **9.** | The seller sends the proposal or message to the buyer. | |
| **10.** | The use case ends. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 3, the seller may opt to view all buyer listings without using specific filters and manually browse through them. | |
| 2. | In step 7, the seller may save the listing to view it later without sending a proposal or message immediately. | |
| **Step #** | **Exception Paths** | |
| 1. | In step 4, if there are no buyer listings that match the search criteria, the system displays a “No Results” message, and the seller can modify the search or filter. | |

### View Dashboard for Sellers

| **Identifier** | | UC-012 |
| --- | --- | --- |
| **Purpose** | | Sellers can access a dashboard that summarizes their listed services and proposals, allowing them to manage their activities on the platform. |
| **Pre-conditions** | | The seller is logged into their account.  The seller has listed services or sent proposals to buyers. |
| **Post-conditions** | | The seller can view a summarized overview of their active, pending, and past listings, as well as the status of proposals they have sent to buyers. They can also take actions such as updating listings or checking proposal statuses. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1.** | The seller navigates to the seller's dashboard. | |
|  | The system displays a summary of the seller's active, pending, and past service listings. | |
|  | The system also displays a summary of proposals sent to buyers and their status (e.g., accepted, pending, declined). | |
| **4.** | The seller clicks on a specific service listing to view its details, edit, or update it. | |
| **5.** | The seller clicks on a proposal to view the buyer's response or check its status. | |
| **6.** | The seller can take further actions such as updating listings or sending follow-up messages for pending proposals. | |
| **7.** | The seller may also use the dashboard to create new service listings or send additional proposals. | |
| **8.** | The use case ends when the seller logs out or navigates away from the dashboard.. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 2, the seller may filter listings and proposals by category, date, or status (e.g., only active listings). | |
| **Step #** | **Exception Paths** | |
| 1. | In step 3, if the seller has no active listings or proposals, the system displays a "No Active Listings/Proposals" message and prompts the seller to create new ones. | |
| 2. | In step 5, if the proposal has been rejected or the listing is no longer available, the system informs the seller and prevents further actions on that proposal or listing. | |

### Monitor System Performance

| **Identifier** | | UC-013 |
| --- | --- | --- |
| **Purpose** | | The system administrator ensures the smooth operation of the marketplace by monitoring the system’s performance. |
| **Pre-conditions** | | The admin is logged into the admin panel. |
| **Post-conditions** | | System performance metrics including business KPIs and Platform KPIs are recorded, and any performance issues are flagged for resolution. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The admin logs into the admin panel. | |
| **2** | The admin navigates to the system monitoring section. | |
| **3** | The admin reviews the current system performance metrics (e.g., uptime, response time). The admin also views platform and business KPIs. | |
| **4** | If issues are identified, the admin flags them for resolution or initiates troubleshooting steps. | |
| **5** | The admin logs out after monitoring is completed. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 3, the admin can also choose to generate a detailed report of system performance metrics instead of only reviewing them. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 4, if no issues are detected, the admin can simply mark the performance check as complete and skip the troubleshooting steps. | |

### 

### Resolve User Complaints

| **Identifier** | | UC-014 |
| --- | --- | --- |
| **Purpose** | | The admin handles user complaints or reported issues, ensuring any problems are resolved in a timely manner. |
| **Pre-conditions** | | The admin is logged into the admin panel and there are user complaints to address. |
| **Post-conditions** | | Complaints are resolved, and users are notified of the resolution. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The admin logs into the admin panel. | |
| **2** | The admin navigates to the "User Complaints" section. | |
| **3** | The admin reviews the list of complaints and prioritizes them based on urgency. | |
| **4** | The admin investigates each issue and takes appropriate action (e.g., contacting the user or resolving technical issues). | |
| **5** | The admin updates the complaint status to "resolved" and notifies the user. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 3, the admin can forward certain complaints to another department if they fall outside the admin’s area of responsibility. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 4, if the complaint cannot be resolved immediately, the admin informs the user of the delay and provides an estimated resolution time. | |

### 

### Approve or Decline Ads

| **Identifier** | | UC-015 |
| --- | --- | --- |
| **Purpose** | | The admin manages the advertisements on the platform, approving or declining ads submitted by users. |
| **Pre-conditions** | | The admin is logged into the admin panel and there are pending ad requests. |
| **Post-conditions** | | Approved ads are posted on the platform, and declined ads are removed from the approval queue. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The admin logs into the admin panel. | |
| **2** | The admin navigates to the "Pending Ads" section. | |
| **3** | The admin reviews the details of each ad. | |
| **4** | The admin decides to approve or decline each ad based on the platform’s policy. | |
| **5** | Approved ads are published, and declined ads are removed from the queue. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 3, the admin can choose to save the ad for further review, instead of approving or declining it immediately. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 5, if the ad contains prohibited content, the admin can flag the ad and report it to a higher authority for review. | |

### 

### Manage User Accounts

| **Identifier** | | UC-0116 |
| --- | --- | --- |
| **Purpose** | | The system administrator manages user accounts, including account approval, deactivation, and reactivation. |
| **Pre-conditions** | | The admin is logged into the admin panel and there are pending user actions (e.g., account creation or deactivation requests). |
| **Post-conditions** | | User accounts are updated based on admin actions (e.g., activated or deactivated). |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The admin logs into the admin panel. | |
| **2** | The admin navigates to the "User Management" section. | |
| **3** | The admin reviews the list of user accounts, including pending verifications or deactivation requests. | |
| **4** | The admin takes the appropriate action (e.g., verify new account, deactivate existing account). | |
| **5** | The system reflects the changes to the user accounts. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 3, the admin can choose to add user notes for further reference, instead of immediately processing requests. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 4, if the user has violated the terms and conditions, the admin can temporarily suspend the account instead of directly deactivating or approving it. | |

### 

### Update Website Content

| **Identifier** | | UC-017 |
| --- | --- | --- |
| **Purpose** | | The system administrator updates the design or content of the website to ensure it stays relevant and functional. |
| **Pre-conditions** | | The admin is logged into the admin panel and there are new design or content updates to be applied including logo,colors etc. |
| **Post-conditions** | | The website content is updated, and users can view the changes. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The admin logs into the admin panel. | |
| **2** | The admin navigates to the "Content Management" section. | |
| **3** | The admin reviews the sections to be updated (e.g., homepage, terms and conditions). | |
| **4** | The admin uploads or edits the content as needed. | |
| **5** | The updates are reflected live on the website after confirmation. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 3, the admin can schedule the content update to be published at a future date rather than immediately. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 4, if the content update violates the formatting rules, the system displays an error message, and the update cannot proceed until corrected. | |

### Manage Google Ads and Analyze Ad Performance

| **Identifier** | | UC-018 |
| --- | --- | --- |
| **Purpose** | | The system admin manages the integration of Google ads on the platform, sets ad parameters, and monitors ad performance through statistics. |
| **Pre-conditions** | | The admin is logged into the admin panel and has access to Google Ads integration features |
| **Post-conditions** | | Google ads are updated, and performance statistics are recorded for further analysis. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The admin logs into the admin panel. | |
| **2** | The admin navigates to the "Google Ads Management" section. | |
| **3** | The admin sets or updates ad parameters (e.g., ad type, target audience, budget). | |
| **4** | The admin activates the ads and monitors real-time performance statistics (e.g., clicks, impressions). | |
| **5** | The admin generates reports to analyze ad effectiveness and optimize performance. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 3, the admin can choose to schedule ads to run during specific periods instead of setting them live immediately. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 4, if the Google Ads API encounters an error or fails to sync, an error message is displayed, and the admin must resolve the issue before proceeding with ad updates. | |

### Listing a tangible Product

| **Identifier** | | UC-019 |
| --- | --- | --- |
| **Purpose** | | Allow a seller to list a tangible product for sale on the platform. |
| **Pre-conditions** | | The seller is logged into the platform.  The seller has a product to list with necessary details such as name, price, and description. |
| **Post-conditions** | | The product is successfully listed and is visible to potential buyers on the platform.  The product listing is complete with all details, including pricing, images, and description. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller navigates to the product listing page on the platform. | |
| **2** | | The seller clicks on "Add New Product" to start the product listing process. | | --- | | |
| **3** | The seller enters product details such as product name, description, and price. | |
| **4** | The seller uploads images of the product to give buyers a visual representation. | |
| **5** | The seller selects or enters additional information, such as stock quantity, category, and delivery options. | |
| **6** | The seller clicks "Submit" to create the product listing. | |
| **7** | The system confirms that the product has been successfully listed and visible to buyers. | |
| **8** | The use case ends when the seller sees a confirmation message that the product is live on the marketplace. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 4, the seller may choose to list the product without images, but the system will prompt them to add one later. | |
| 2 | In step 5, the seller can opt to save the product as a draft and complete the listing at a later time. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 6, if the seller submits incomplete or incorrect product details, the system displays an error and prompts for corrections. | |
| 2 | If the system experiences a failure during listing, an error message is displayed, and the product is not listed until the issue is resolved. | |

### 

### 

### Seller can respond to buyers

| **Identifier** | | UC-020 |
| --- | --- | --- |
| **Purpose** | | To allow the seller to respond to an offer made by a buyer by choosing one of three options: **Accept**, **Decline**, or **Revise** the offer, facilitating effective negotiation. |
| **Pre-conditions** | | The seller is logged into the platform.  The buyer has submitted an offer for a product listed by the seller.  The “Revise Offer” feature is integrated into the platform. |
| **Post-conditions** | | The seller’s response (Accept, Decline, or Revise) is communicated to the buyer.  If accepted, the transaction proceeds to the next step (e.g., payment processing or shipping).  If revised, the updated offer is sent to the buyer for review.  If declined, the buyer is notified, and no further action is required unless re-initiated. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller logs into the platform and navigates to the "Offers Received" section. | |
| **2** | The seller selects the offer they want to respond to. | |
| **3** | The system displays the offer details, including the price and any additional terms from the buyer. | |
| **4** | The seller is presented with three response options: **Accept**, **Decline**, or **Revise**. | |
| **5** | The seller chooses to either accept, decline, or revise the offer. | |
| **6** | | * If the seller accepts, the system confirms the acceptance and notifies the buyer. | | --- |  * If the seller declines, the system sends a notification to the buyer and closes the negotiation. * If the seller revises, they are prompted to modify the offer (price, quantity, etc.), and the system sends the revised offer back to the buyer for approval. | |
| **7** | The system updates the offer status based on the seller’s action (Accepted, Declined, or Revised) and notifies the buyer accordingly. | |
| **8** | The use case ends once the response has been successfully submitted and the buyer has been notified. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | The seller may skip the offer review and respond immediately by selecting an option (Accept, Decline, or Revise) from the initial notification without viewing all offer details. | |
| 2 | If the seller chooses to revise the offer, they can adjust any details (e.g., price or quantity) before submitting. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | If the seller attempts to respond to an expired or withdrawn offer, the system displays an error message, and the response is not processed. | |
| 2 | If the system fails to submit the seller’s response due to a technical error, the seller is prompted to retry. | |

### 

### Rating a buyer

| **Identifier** | | UC-021 |
| --- | --- | --- |
| **Purpose** | | Allow the seller to provide a rating for the buyer based on the overall experience, helping build trust and transparency in future transactions. |
| **Pre-conditions** | | The seller has successfully completed a transaction with the buyer.  A rating and review system is integrated into the platform. |
| **Post-conditions** | | The seller’s rating is recorded and displayed in the buyer’s profile.  Future sellers can view this rating when engaging with the buyer. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller logs into the platform and navigates to the "Completed Transactions" section. | |
| **2** | The seller selects the transaction with the buyer they wish to rate. | |
| **3** | The system displays a "Rate Buyer" option, allowing the seller to click and proceed with the rating. | |
| **4** | The seller rates the buyer on a predefined scale (e.g., 1-5 stars) based on their experience. | |
| **5** | The seller can optionally provide additional feedback or comments about the transaction. | |
| **6** | The system confirms the submission of the rating and updates the buyer’s profile with the new rating. | |
| **7** | The seller can view the rating posted and finalize the interaction. | |
| **8** | The use case ends once the rating has been successfully submitted and recorded in the buyer’s profile. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | In step 5, the seller may skip providing detailed feedback and only submit a star rating. | |
| 2 | The seller can update their rating if the platform allows modification of ratings within a specified timeframe. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | If the system fails to record the rating, the seller is shown an error message and asked to retry. | |
| 2 | If the seller tries to rate a buyer before the transaction is completed, the system restricts the rating option. | |

### 

### Sending an Ad offer to the buyer

| **Identifier** | | UC-022 |
| --- | --- | --- |
| **Purpose** | | Allow the seller to send advertisements for their products to potential buyers, responding to the buyers’ specific interests. |
| **Pre-conditions** | | The seller has products available for sale on the platform.  The seller has access to potential buyers' interests or requests on the platform. |
| **Post-conditions** | | The potential buyer receives the ad and can review it.  The ad proposal is recorded in the system for the buyer to accept or reject. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller logs into the platform and navigates to the "Create Ads" section. | |
| **2** | The seller reviews the list of products available for advertisement. | |
| **3** | The system displays product details, including specifications, price, and images. | |
| **4** | The seller selects a product to advertise and clicks the "Send Ad" button. | |
| **5** | The system prompts the seller to input ad details, including a brief description, target audience, and any promotional offers. | |
| **6** | The seller submits the ad, which is sent directly to the potential buyer’s homepage. | |
| **7** | The system confirms that the ad has been sent, and the seller can view it in their "Sent Ads" section. | |
| **8** | The use case ends when the ad is successfully sent to the potential buyers. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | The seller can modify or withdraw the ad if the buyer hasn’t responded yet. | |
| 2 | The seller may send multiple ads for different products to the same buyer if more than one product matches the buyer's interests. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | If the system detects that the ad content does not comply with platform guidelines, a warning is shown to the seller. | |
| 2 | If the ad fails to send due to a system error, the seller is notified, and the system prompts them to retry. | |

### 

* + 1. Updating the details of already listed tangible product

| **Identifier** | | UC-023 |
| --- | --- | --- |
| **Purpose** | | Allow the seller to update the details of a product they previously listed, including price, description, and images. |
| **Pre-conditions** | | The seller is logged into the platform.  The seller has already listed a product on the platform. |
| **Post-conditions** | | The updated product information is saved and reflected in the product listing.  Buyers viewing the product see the updated details.  Buyers viewing the product see the last preserved price. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The seller logs into their account on the platform. | |
| **2** | The seller navigates to the "Manage Products" section. | |
| **3** | The seller selects the product they want to update from the list of their listings. | |
| **4** | The system displays the product details, including current price, description, and images. | |
| **5** | The seller clicks the "Edit" button to update any of the product details. | |
| **6** | The seller modifies the necessary information (price, description, images, etc.). | |
| **7** | The seller submits the changes by clicking the "Save" button. | |
| **8** | The system saves the updated product information and reflects the changes in the product listing. | |
| **9** | The use case ends when the seller successfully updates the product and the changes are visible to buyers with the last preserved price. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | The seller can choose to update only specific details (e.g., only the price or images) without altering others. | |
| 2 | The seller may preview the updated listing before submitting the changes to ensure accuracy. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | If the seller fails to provide valid data (e.g., incorrect image format, invalid price), the system shows an error. | |
| 2 | If the system encounters an issue saving the updates, the seller is notified and asked to retry the update. | |

### 

### Listing a Service

| **Identifier** | | UC-025 |
| --- | --- | --- |
| **Purpose** | | Enable buyers to create and post specific service or product requests, detailing their unique requirements to attract relevant proposals from sellers. |
| **Pre-conditions** | | The buyer is registered and authenticated on the platform.  The buyer has access to the service listing feature.  Necessary permissions and account status are verified. |
| **Post-conditions** | | The buyer is registered and authenticated on the platform.  The buyer has access to the service listing feature.  Necessary permissions and account status are verified. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The buyer navigates to the "Post a Service Request" section from their dashboard. | |
|  | The buyer fills out the service request form, specifying details such as service type, description, budget, deadline, and any additional requirements or preferences. | |
|  | The buyer selects relevant categories and tags to ensure the request reaches the appropriate sellers. | |
| **4.** | The buyer reviews the entered information for accuracy and completeness. | |
| **5.** | The buyer submits the service request by clicking the "Post Request" button. | |
| **6.** | The system validates the input data and confirms the successful listing of the service request. | |
| **7.** | The system notifies relevant sellers about the new service request based on the selected categories and tags. | |
| **8.** | The buyer can view the newly listed service request in their "My Service Requests" dashboard. | |
| **9.** |  | |
| **10.** |  | |
| **11.** |  | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | In step 2, the buyer may choose to save the service request as a draft and post it later. | |
| 2. | In step 3, if the system detects missing or incorrect information, it prompts the buyer to revise the request before submitting. | |
| 3. | In step 6, the buyer may preview the service request before final submission to ensure all details are correct. | |
| **Step #** | **Exception Paths** | |
| 1. | In step 3, if file uploads fail due to size or format restrictions, the system displays an error message prompting the buyer to adjust the files. | |
| 2. | In step 6, if mandatory fields are incomplete or contain invalid data, the system highlights the issues and prompts the buyer to rectify them. | |
| 3. | In step 7, if the system encounters an error while posting the request, an error message is displayed, and the buyer is prompted to retry. | |

### Reviewing Proposals (Accept/Deny)

| **Identifier** | | UC-026 |
| --- | --- | --- |
| **Purpose** | | Allow buyers to evaluate proposals submitted by sellers in response to their service requests and make informed decisions by accepting or rejecting proposals. |
| **Pre-conditions** | | The buyer has posted one or more service requests.  Sellers have submitted proposals in response to these service requests.  The buyer is authenticated and authorized to review proposals. |
| **Post-conditions** | | The buyer has either accepted or rejected each reviewed proposal.  Accepted proposals may lead to further actions such as contract signing or payment processing.  Rejected proposals are marked accordingly, and sellers are notified of the decision. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The buyer accesses their "My Service Requests" dashboard to view all active service listings. | |
|  | The buyer selects a specific service request to view all received proposals. | |
|  | The buyer reviews each proposal's details, including seller information, pricing, timelines, and any attached documents or portfolios. | |
| **4.** | For each proposal, the buyer assesses its suitability based on their requirements and criteria. | |
| **5.** | The buyer decides to either accept or reject the proposal by clicking the corresponding "Accept" or "Reject" button. | |
| **6.** | If accepted, the system confirms the acceptance and may prompt the buyer to proceed with next steps such as contract formalization. | |
| **7.** | If rejected, the system updates the proposal status and notifies the seller of the rejection. | |
| **8.** | The buyer continues reviewing other proposals until all desired decisions (acceptances or rejections) are made. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | **Step 3:** The buyer can sort proposals based on criteria like price, rating, or submission date to facilitate easier review. | |
| 2. | **Step 4:** The buyer may choose to request additional information or clarifications from the seller before making a decision. | |
| 3. | **Step 5:** The buyer can temporarily save their decision on a proposal and revisit it later for final action. | |
| **Step #** | **Exception Paths** | |
| 1. | **Step 3:** If proposal details are incomplete or missing critical information, the system prompts the buyer to request more details from the seller. | |
| 2. | **Step 5:** If the system encounters an error while processing the accept/reject action, an error message is displayed, and the buyer is prompted to retry. | |
| 3. | **Step 7:** If notifying the seller fails due to a system error, the system logs the issue and retries the notification process automatically. | |

* + 1. Displaying All Listings on the Landing Page

| **Identifier** | | UC-027 |
| --- | --- | --- |
| **Purpose** | | Provide buyers with an overview of all available service requests on the platform's landing page, enabling easy browsing, filtering, and discovery of relevant listings. |
| **Pre-conditions** | | The platform has active service listings posted by buyers.  Buyers are accessing the platform's landing page.  The landing page is designed to display service listings dynamically. |
| **Post-conditions** | | Buyers can view, search, and interact with service listings on the landing page.  Service listings are presented in an organized and user-friendly manner, facilitating discovery and engagement. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | A buyer navigates to the platform's landing page either by visiting the website or opening the mobile app. | |
| **2** | The landing page loads and displays a list/grid of active service requests, highlighting key details such as title, category, budget, and deadline. | |
| **3** | The buyer uses the search bar to enter keywords related to the services they are interested in. | |
| **4** | The buyer applies filters (e.g., category, budget range, location, deadline) to narrow down the listings displayed. | |
| **5** | The buyer browses through the filtered listings, viewing summaries and essential information for each service request. | |
| **6** | The buyer clicks on a specific service listing to view detailed information and possibly take further actions such as saving the listing. | |
| **7** | The buyer can also sort the listings based on criteria like relevance, newest, highest budget, or most responses. | |
| **8** | The buyer may interact with additional features such as featured listings, trending service requests, or recommended services based on their activity. | |
| **9** | The buyer can paginate through multiple pages of listings or use infinite scrolling if implemented. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | **Step 3:** The buyer can use advanced search options to combine multiple keywords and criteria for a more precise search. | |
| 2. | **Step 4:** The buyer can save specific filter settings for future visits to quickly access preferred types of service requests. | |
| 3. | **Step 6:** Instead of clicking to view details, the buyer can hover over a listing to see a quick preview of more information. | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | **Step 2:** If no active service listings are available, the landing page displays a message indicating the absence of listings and may prompt the buyer to post a service request. | |
| 2 | **Step 3:** If the search query yields no results, the system suggests alternative keywords or prompts the buyer to adjust their search criteria. | |
| 3. | **Step 4:** If filter parameters are invalid or no listings match the selected filters, the system notifies the buyer and suggests modifying the filters. | |
| 4. | **Step 7:** If sorting fails due to a system error, the listings revert to a default order, and the system notifies the buyer of the issue. | |
| 5. | **Step 9:** If pagination or infinite scrolling fails to load additional listings, the system displays an error message and offers a retry option. | |

* + 1. Rating the sellers

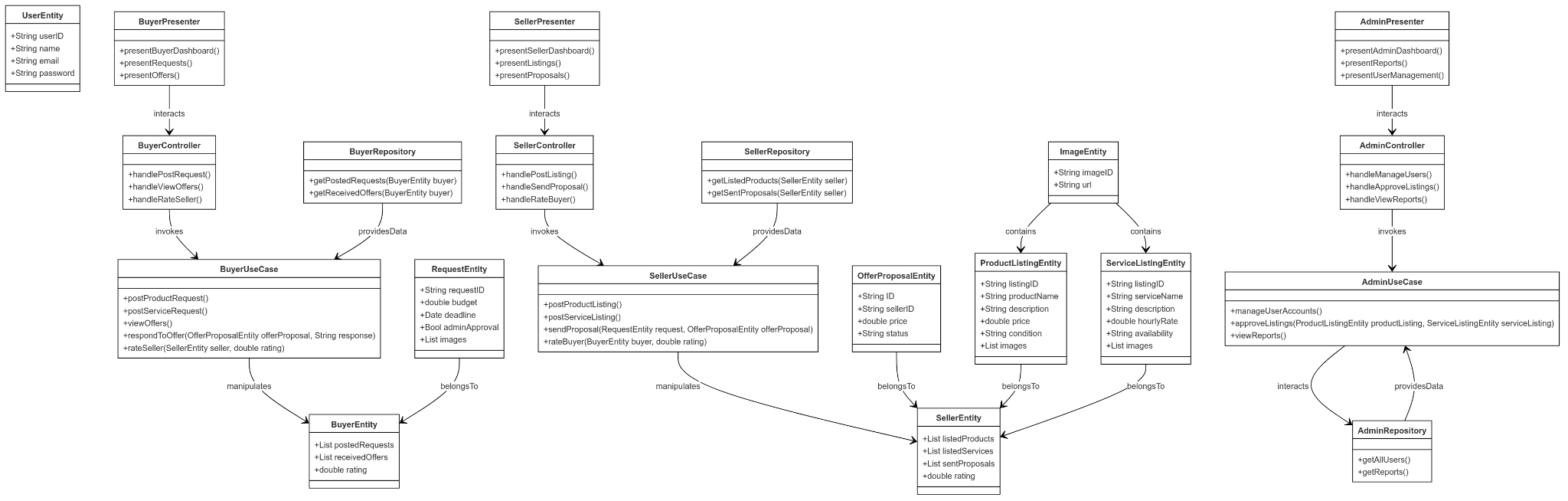
| **Identifier** | | UC-028 |
| --- | --- | --- |
| **Purpose** | | Buyers can rate the seller after the completion of a purchase. |
| **Pre-conditions** | | The buyer has successfully received the completed service.  The service is marked as completed on the platform. |
| **Post-conditions** | | The buyer's ratings are recorded and visible on the seller's profile. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| **1** | The buyer receives a notification that the service has been completed. | |
| **2** | The buyer navigates to the "Completed Orders" section. | |
| **3** | The buyer selects the service and clicks on the "Rate the seller" button. | |
| **4** | The buyer rates the seller using a star or points system (e.g., 1 to 5 stars). | |
| **5** | The system confirms the submission of the rating. | |
| **6** | The seller is notified of the buyer's rating. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1 | The buyer may choose to not rate the seller. | |
| 2 | The buyer may save rating for later | |
|  | | |
| **Step #** | **Exception Paths** | |
| 1 | In step 6, if the system fails to submit the rating, the buyer is prompted to retry. | |

### 

# Class Diagram

## Diagram

* + 1. Links: [Class DGR](https://www.mermaidchart.com/app/projects/0c2ab960-f537-4aec-b945-3c65be4df284/diagrams/a6ead1ea-4aa7-454c-be27-f8b4a8a7c6be/version/v0.1/edit)



## Description

### **User Class**

**Purpose**: User Class represents the base class for all users in the system, including buyers and sellers. Contains general attributes and functions related to user authentication and interaction with the platform.

**Attributes**:

userID: Unique identifier for each user.

name: The name of the user.

email: The email address of the user.

password: The user's password.

**Operations**:

login(): Allows the user to log into the platform.

logout(): Logs the user out of the platform.

### **Buyer Class**

**Purpose**: Buyer Class represents users who post requests for products and services. Buyers interact with sellers by viewing offers and proposals, and can update their listings or rate sellers after transactions.

**Attributes**:

postedProductRequests: A list of product requests posted by the buyer.

postedServiceRequests: A list of service requests posted by the buyer.

receivedOffers: A list of offers received from sellers.

rating: The buyer's average rating based on reviews by sellers.

**Operations**:

postProductRequest(): Allows the buyer to post a product request.

postServiceRequest(): Allows the buyer to post a service request.

viewOffers(): Allows the buyer to view offers from sellers.

respondToOffer(Offer offer, String response): Allows the buyer to respond to offers by accepting, rejecting, or negotiating.

updateListing(): Allows the buyer to update their existing requests or listings.

rateSeller(Seller seller, double rating): Allows the buyer to rate a seller after a transaction.

### **Seller Class**

**Purpose**: Seller Class represents users who list products or services for sale. Sellers can post new listings, send proposals to buyers, negotiate offers, and rate buyers after transactions.

**Attributes**:

listedProducts: A list of product listings posted by the seller.

listedServices: A list of service listings posted by the seller.

sentProposals: A list of proposals sent to buyers in response to their requests.

rating: The seller's average rating based on reviews by buyers.

**Operations**:

postProductListing(): Allows the seller to post a new product listing.

postServiceListing(): Allows the seller to post a new service listing.

sendProposal(ServiceRequest request, Proposal proposal): Allows the seller to send proposals in response to buyer requests.

negotiate(Proposal proposal, String terms): Allows the seller to negotiate terms of a proposal with the buyer.

updateListingDetails(Listing listing): Allows the seller to update details of an existing listing.

rateBuyer(Buyer buyer, double rating): Allows the seller to rate a buyer after a transaction.

### **ProductRequest Class**

**Purpose**: ProductRequest Class represents a request posted by a buyer for a specific product. Contains details such as the product name, budget, and condition.

**Attributes**:

requestID: Unique identifier for the product request.

productName: Name of the requested product.

description: A description of the product the buyer is looking for.

budget: The budget range for the product.

condition: Desired condition of the product (e.g., new, used, refurbished).

deadline: Deadline for receiving offers on the request.

**Operations**:

addAdditionalDetails(): Allows the buyer to add extra information to the product request.

### **ServiceRequest Class**

**Purpose**: ServiceRequest Class represents a request posted by a buyer for a service. Contains details such as the service name, budget, and specific requirements.

**Attributes**:

requestID: Unique identifier for the service request.

serviceName: Name of the requested service.

description: A description of the service the buyer is looking for.

budget: The budget range for the service.

requirements: Specific requirements for the requested service.

deadline: Deadline for receiving offers on the service request.

**Operations**:

uploadReferenceFiles(): Allows the buyer to upload reference documents or files related to the service.

viewProposals(): Allows the buyer to view proposals received from sellers.

### **Offer Class**

**Purpose**: Offer Class represents an offer made by a seller in response to a product or service request from a buyer. Contains details about the offer, including price and status.

**Attributes**:

offerID: Unique identifier for the offer.

productOrServiceID: Identifier of the product or service related to the offer.

sellerID: The ID of the seller who made the offer.

price: The proposed price for the product or service.

status: The current status of the offer (e.g., pending, accepted, declined, negotiated).

**Operations**:

sendOffer(): Allows the seller to send the offer to the buyer.

acceptOffer(): Allows the buyer to accept the offer.

declineOffer(): Allows the buyer to decline the offer.

negotiateOffer(String newTerms): Allows the buyer to negotiate the offer terms.

### **Proposal Class**

**Purpose**: Offer Class represents a formal proposal submitted by a seller in response to a buyer's service or product request. Includes the price and conditions.

**Attributes**:

proposalID: Unique identifier for the proposal.

requestID: The ID of the service or product request the proposal is responding to.

sellerID: The ID of the seller who submitted the proposal.

price: The price proposed by the seller.

conditions: Any specific conditions or terms related to the proposal.

status: The status of the proposal (e.g., pending, accepted, declined, negotiated).

**Operations**:

submitProposal(): Allows the seller to submit a proposal to the buyer.

updateProposal(): Allows the seller to update an existing proposal.

viewBuyerResponse(): Allows the seller to view the buyer's response to the proposal.

### **ProductListing Class**

**Purpose**: ProductListing Class represents a product listing posted by a seller, containing details like price, description, and associated images.

**Attributes**:

listingID: Unique identifier for the product listing.

productName: Name of the product listed for sale.

description: Description of the product.

price: Price of the product.

condition: Condition of the product (e.g., new, used, refurbished).

category: Category under which the product is listed.

images: A list of images associated with the product.

**Operations**:

updateListingDetails(): Allows the seller to update the details of the product listing.

deleteListing(): Allows the seller to remove the product listing.

### **ServiceListing Class**

**Purpose**: ServiceListing Class represents a service listing posted by a seller, including details about the service offered and its availability.

**Attributes**:

listingID: Unique identifier for the service listing.

serviceName: Name of the service listed for sale.

description: Description of the service.

hourlyRate: The hourly rate charged for the service.

availability: The availability of the service.

**Operations**:

uploadServiceDetails(): Allows the seller to upload additional details about the service.

updateServiceDetails(): Allows the seller to update the service listing.

removeServiceListing(): Allows the seller to remove the service listing from the platform.

### **Admin Class**

**Purpose**: Admin Class represents platform administrators who manage the platform, oversee user accounts, approve or decline ads, monitor system performance, and resolve user complaints.

**Attributes**:

adminID: Unique identifier for the admin.

name: The name of the admin.

**Operations**:

viewReports(): Allows the admin to view platform reports.

manageUserAccounts(): Allows the admin to manage user accounts (activation, suspension, etc.).

monitorSystemPerformance(): Allows the admin to monitor platform performance.

resolveUserComplaints(): Allows the admin to resolve user complaints.

approveOrDeclineAds(): Allows the admin to approve or decline ads posted on the platform.

manageWebsiteContent(): Allows the admin to update the content of the website.

analyzeGoogleAds(): Allows the admin to manage and analyze Google Ads performance.

### **Image Class**

**Purpose**: Image Class handles the media associated with product listings, allowing sellers to upload and delete images of their products.

**Attributes**:

imageID: Unique identifier for the image.

url: URL of the image file.

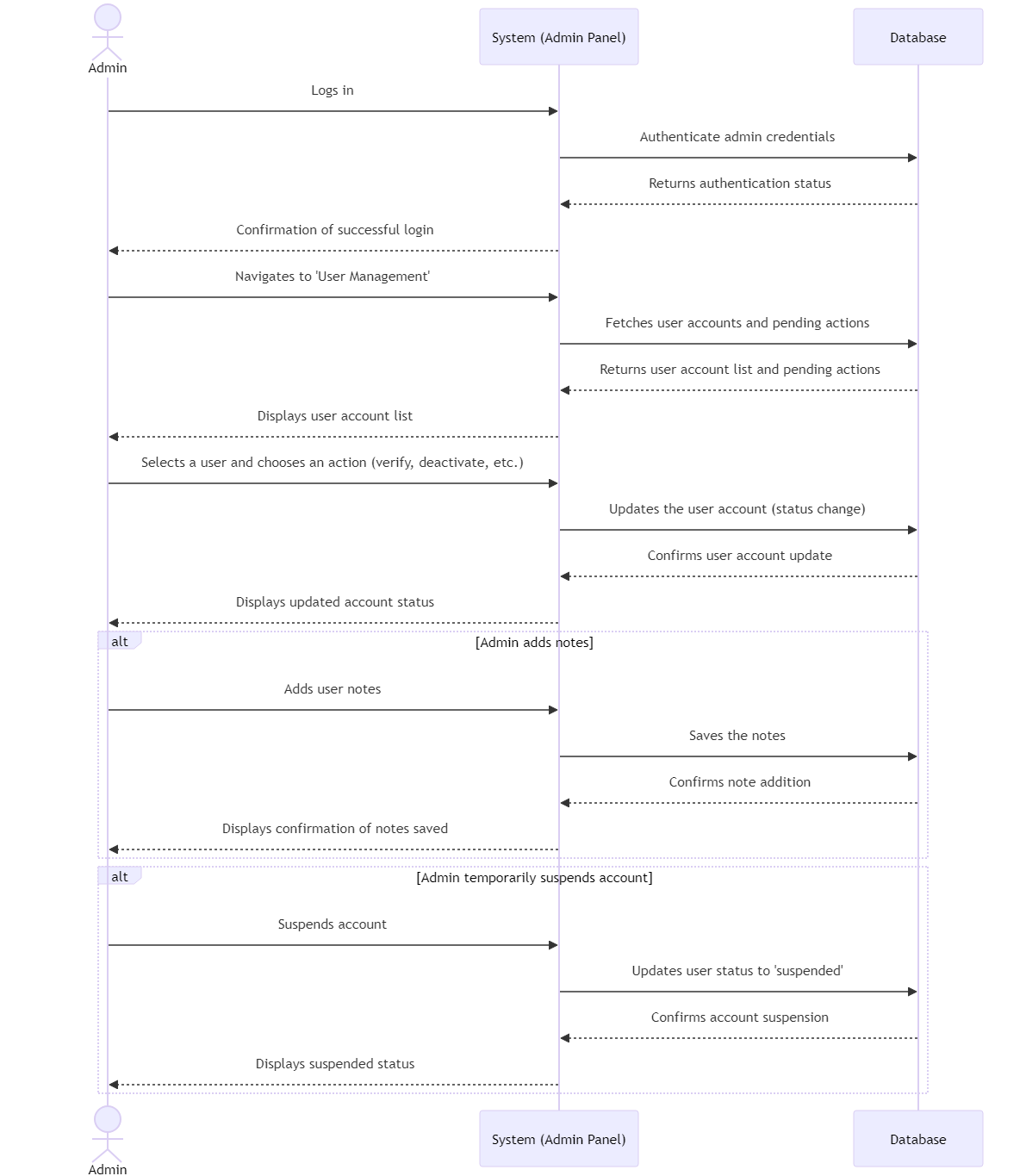
**Operations**:

uploadImage(): Allows the seller to upload images of their products.

deleteImage(): Allows the seller to delete images from the listing.

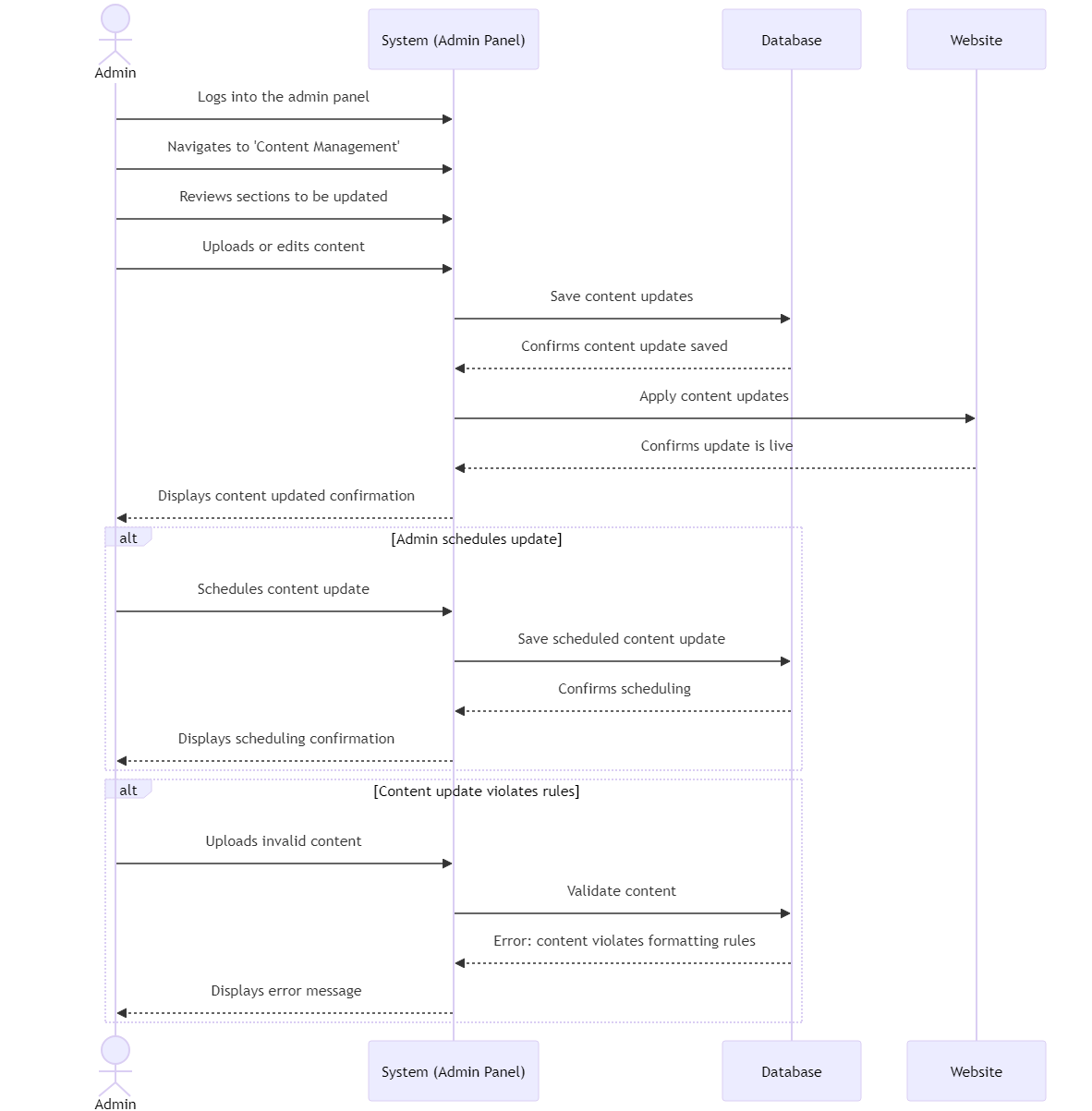
# Sequence Diagrams

## Manage User Accounts

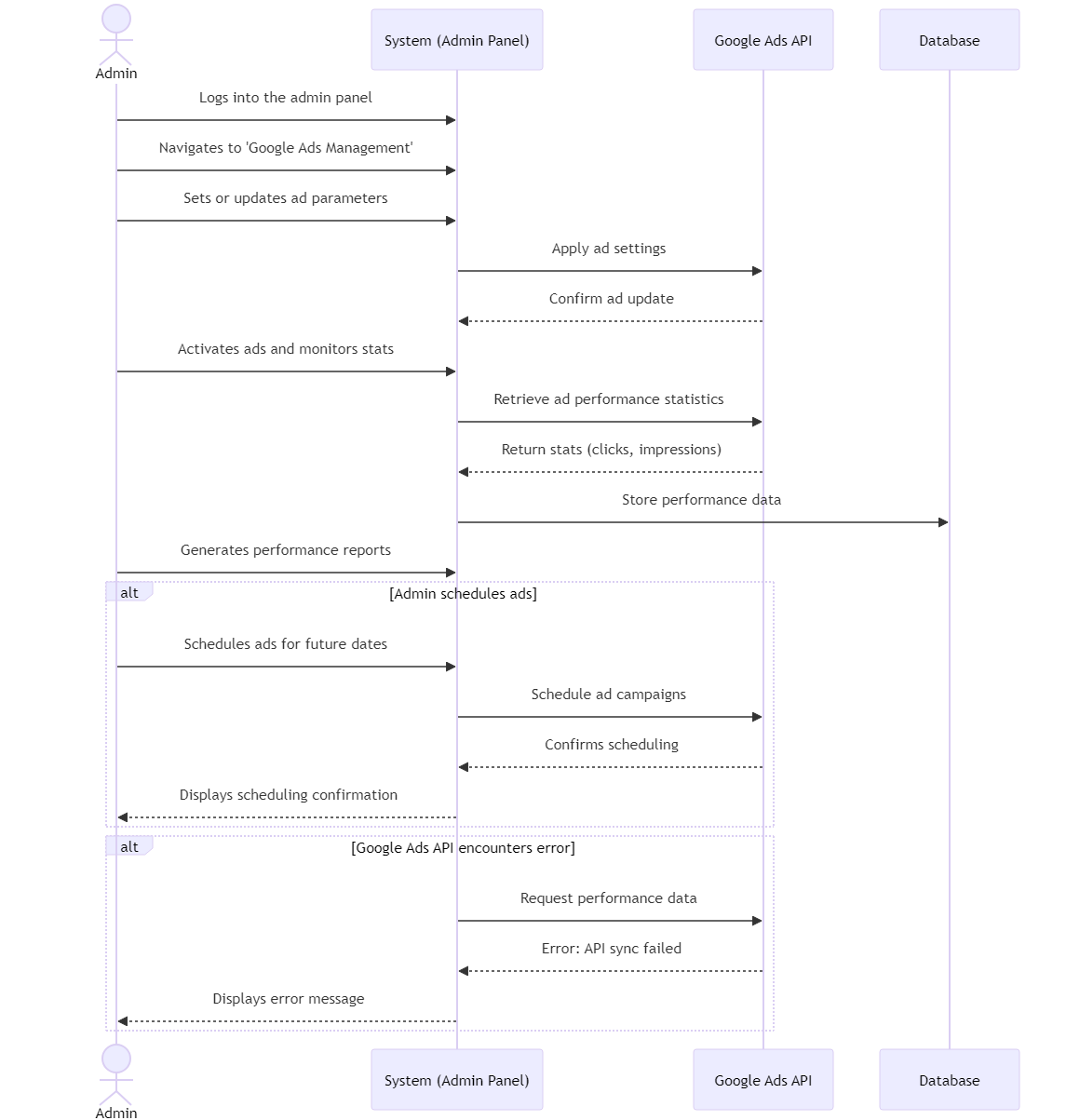


## Update Website Content

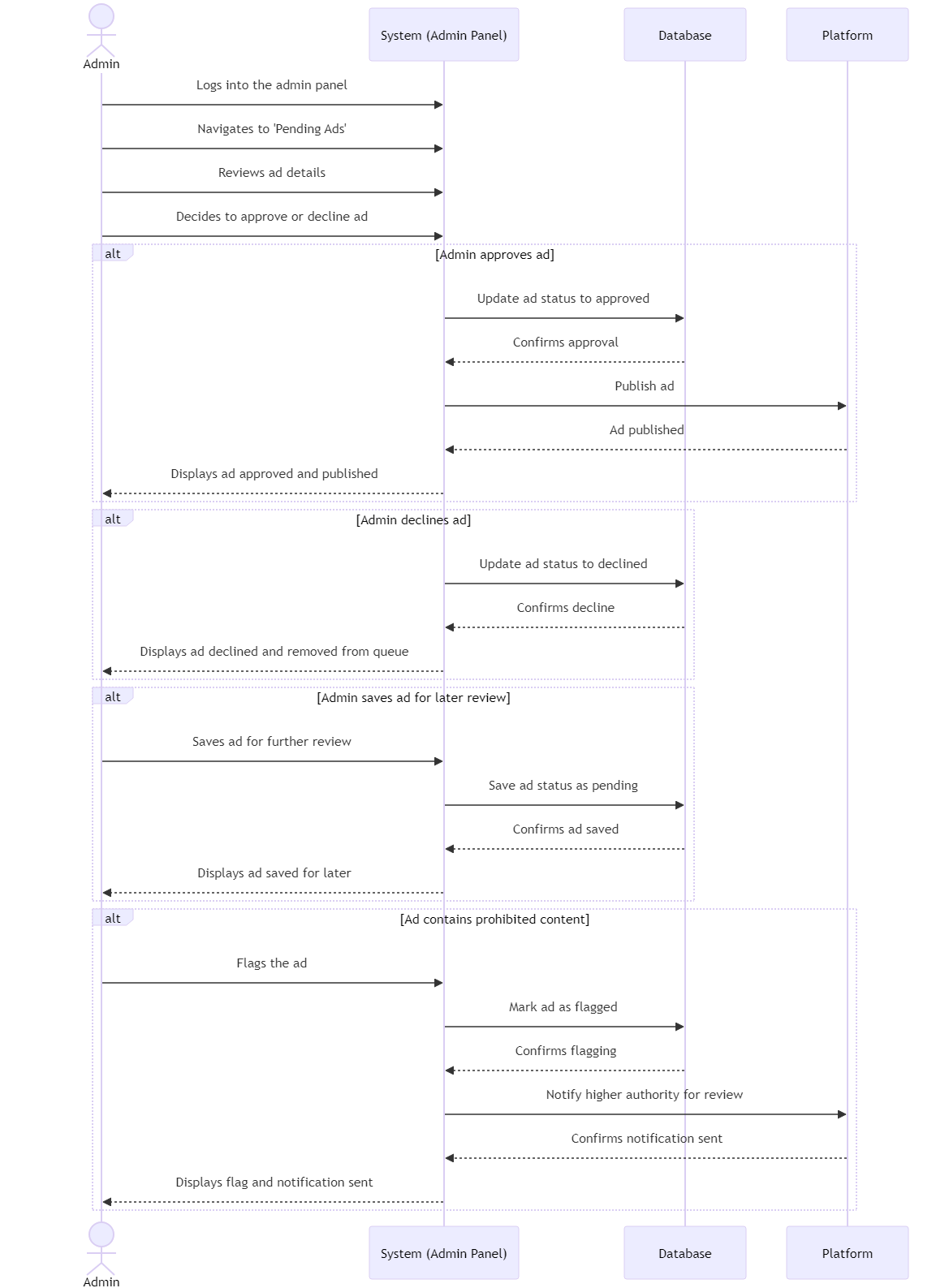
<Draw the sequence diagram of the use case using standard UML notation>



## Manage Google Ads

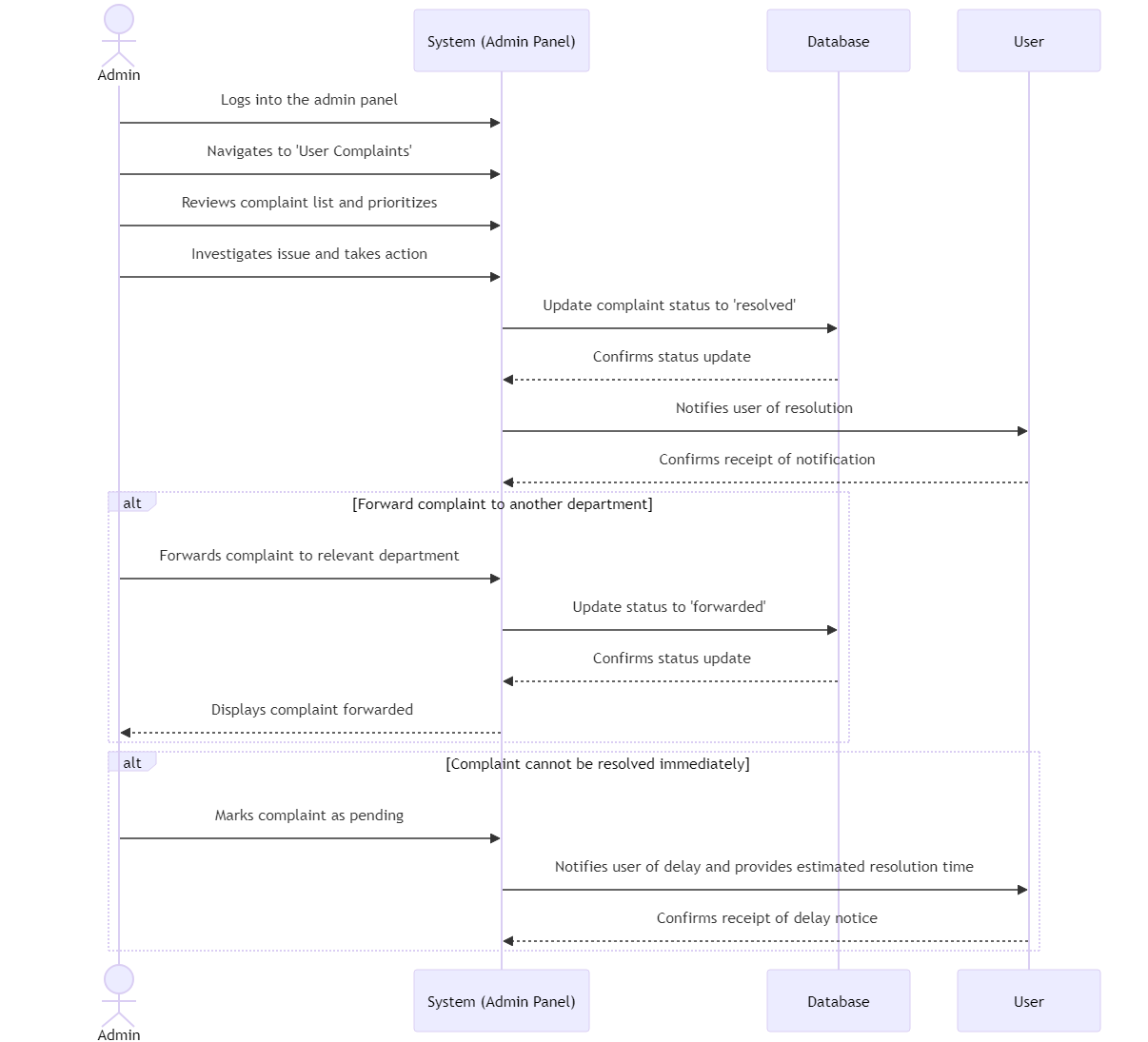


## Approve or Decline Ads

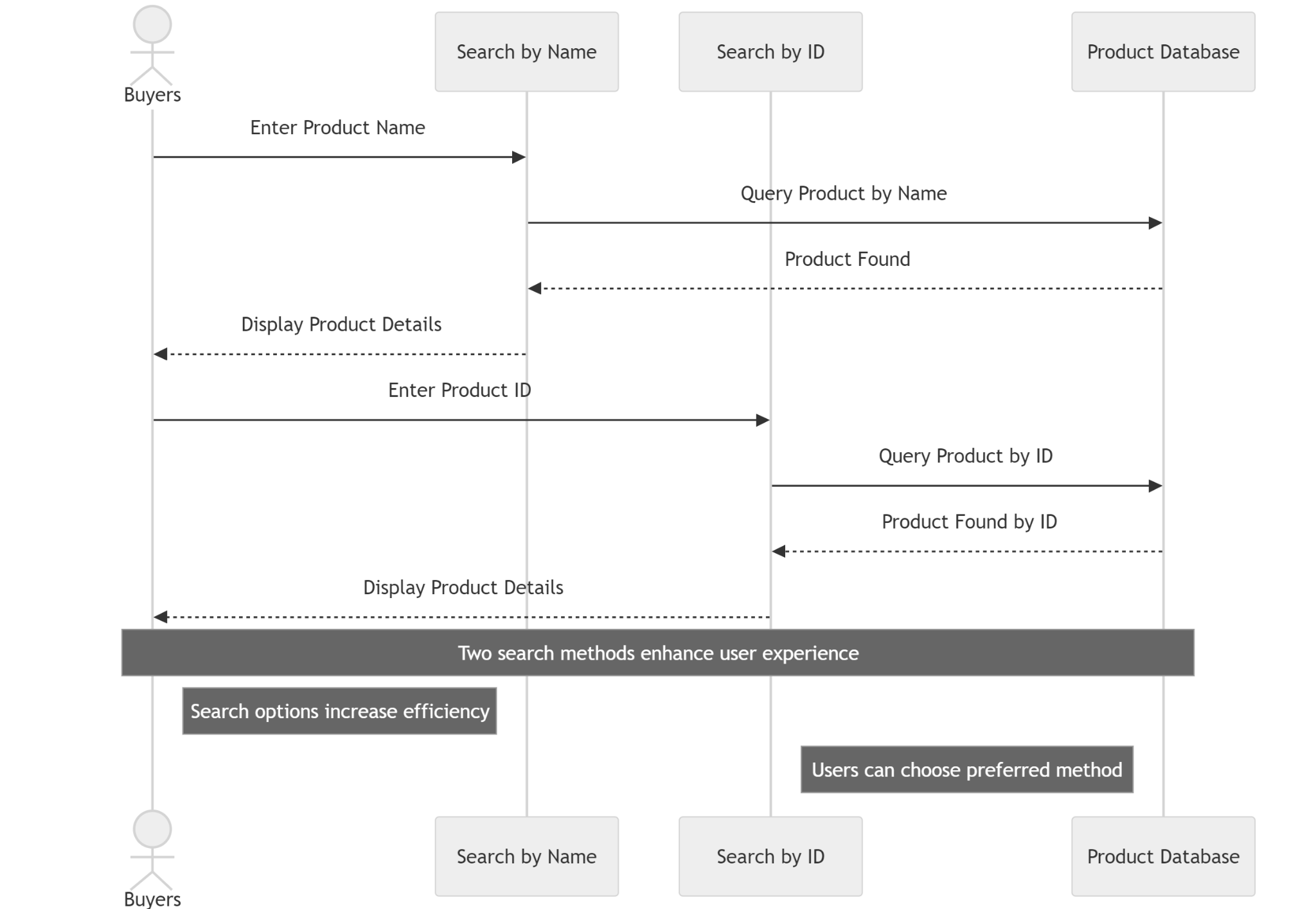


## 

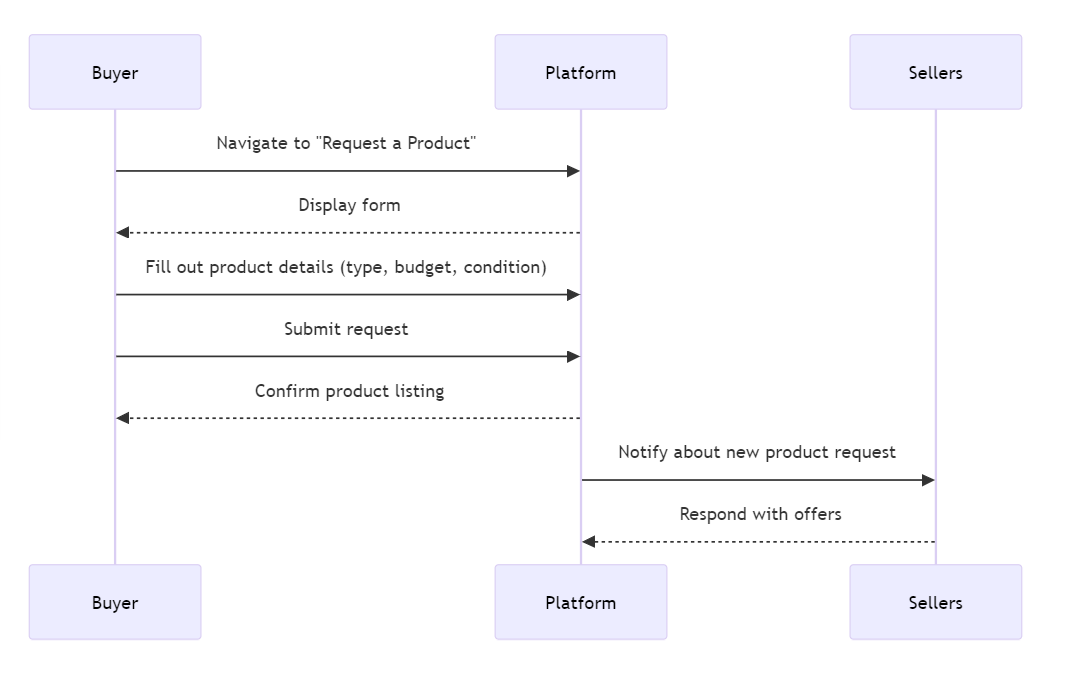
## Resolve User Complaints



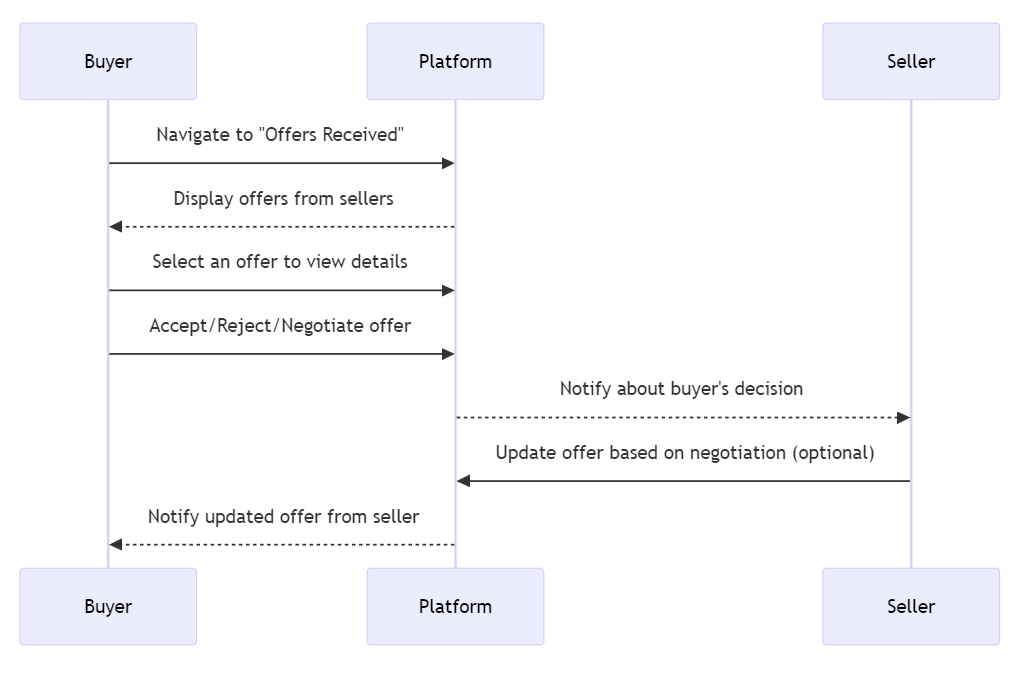
## Buyers can search seller’s listings



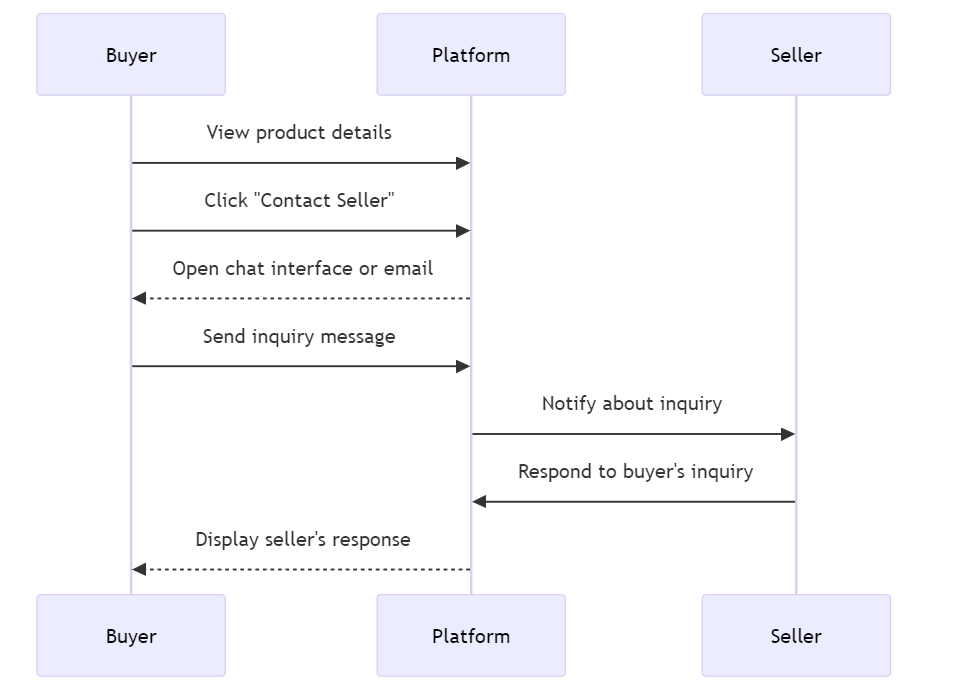
## Buyers can make the listings



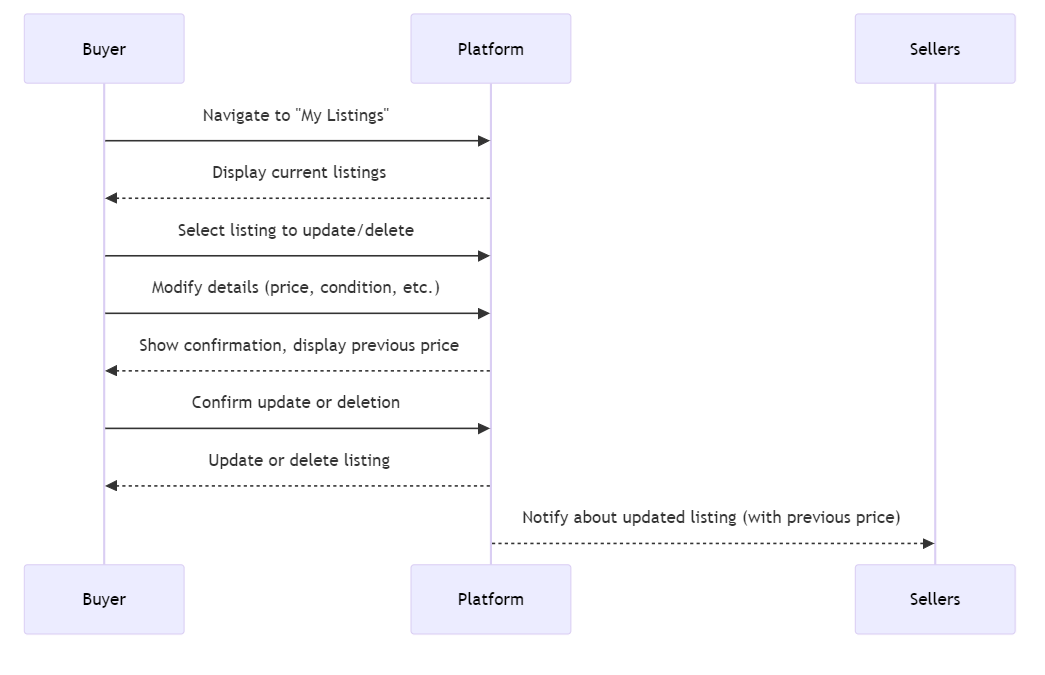
## Buyers can respond to sellers’ offers



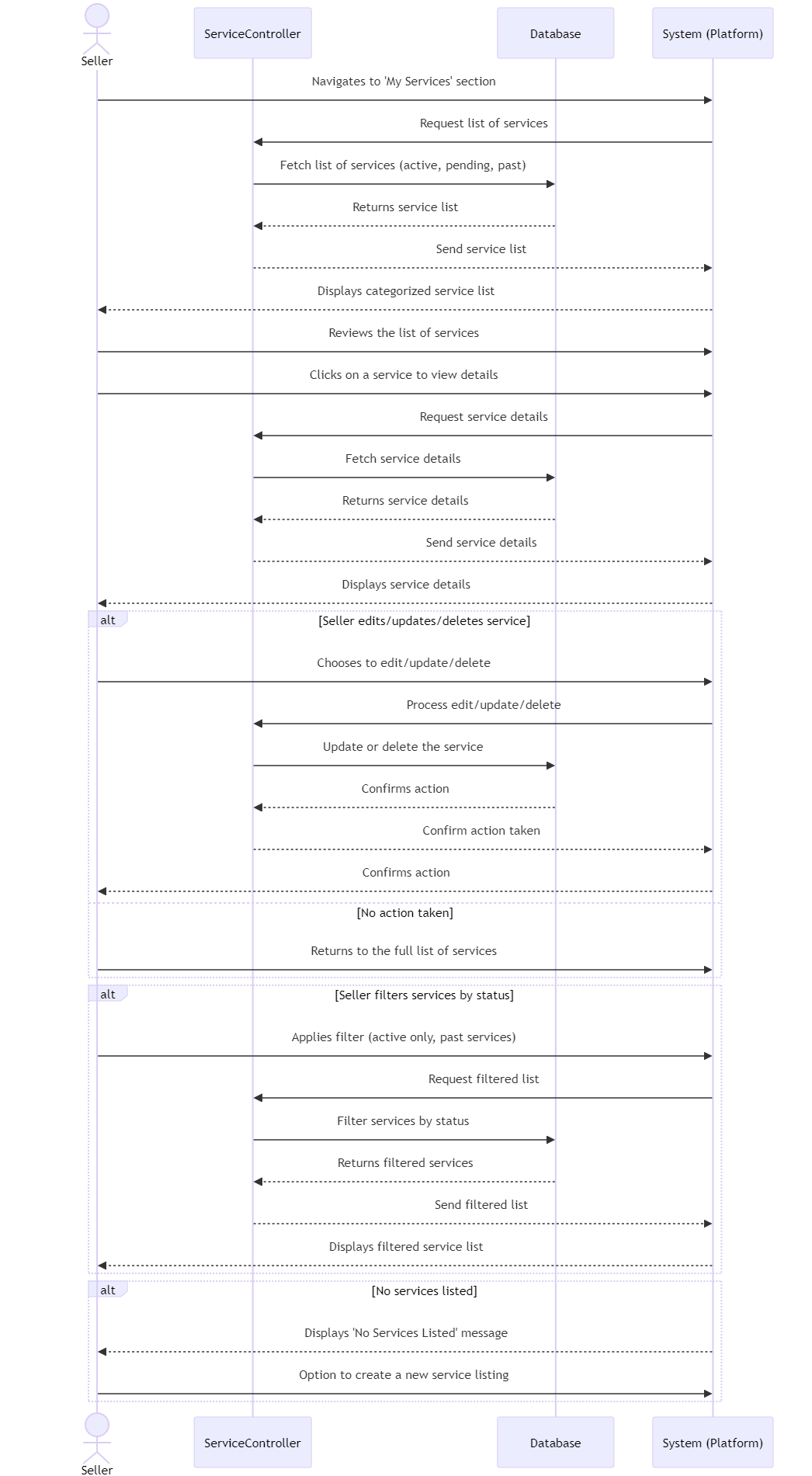
## Buyers can reach out to sellers



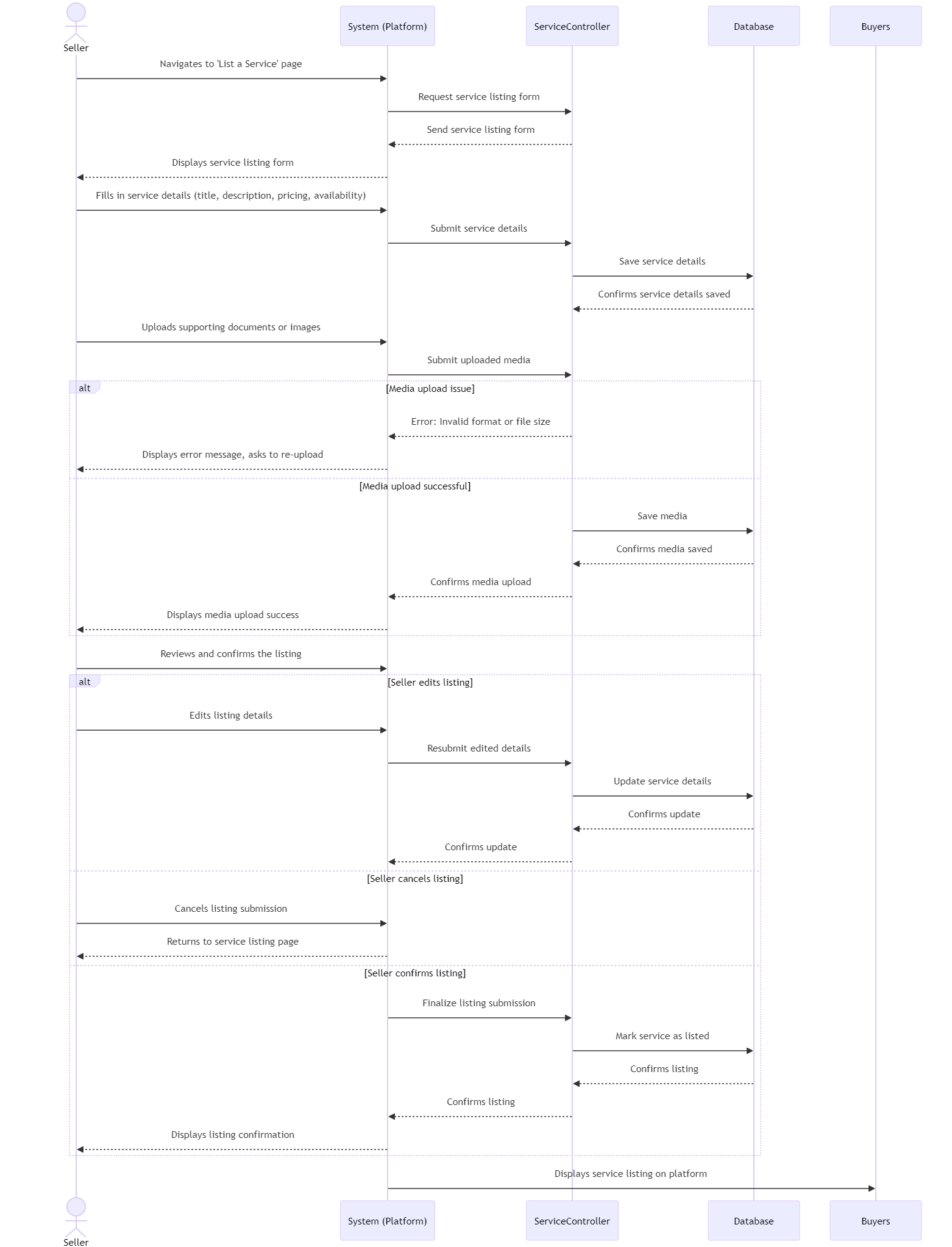
## Buyers can update the pre existing listing



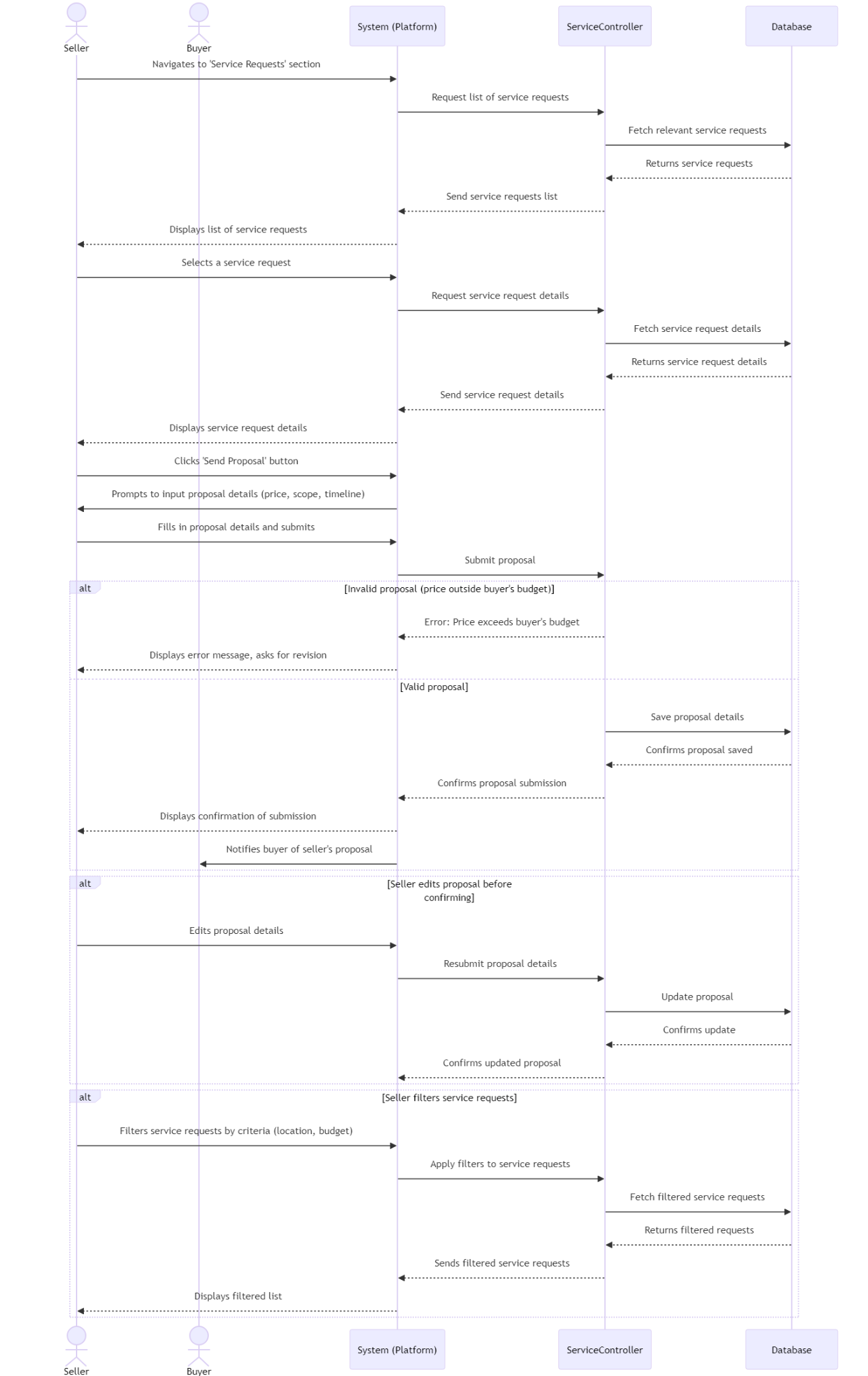
## View All Listed Services



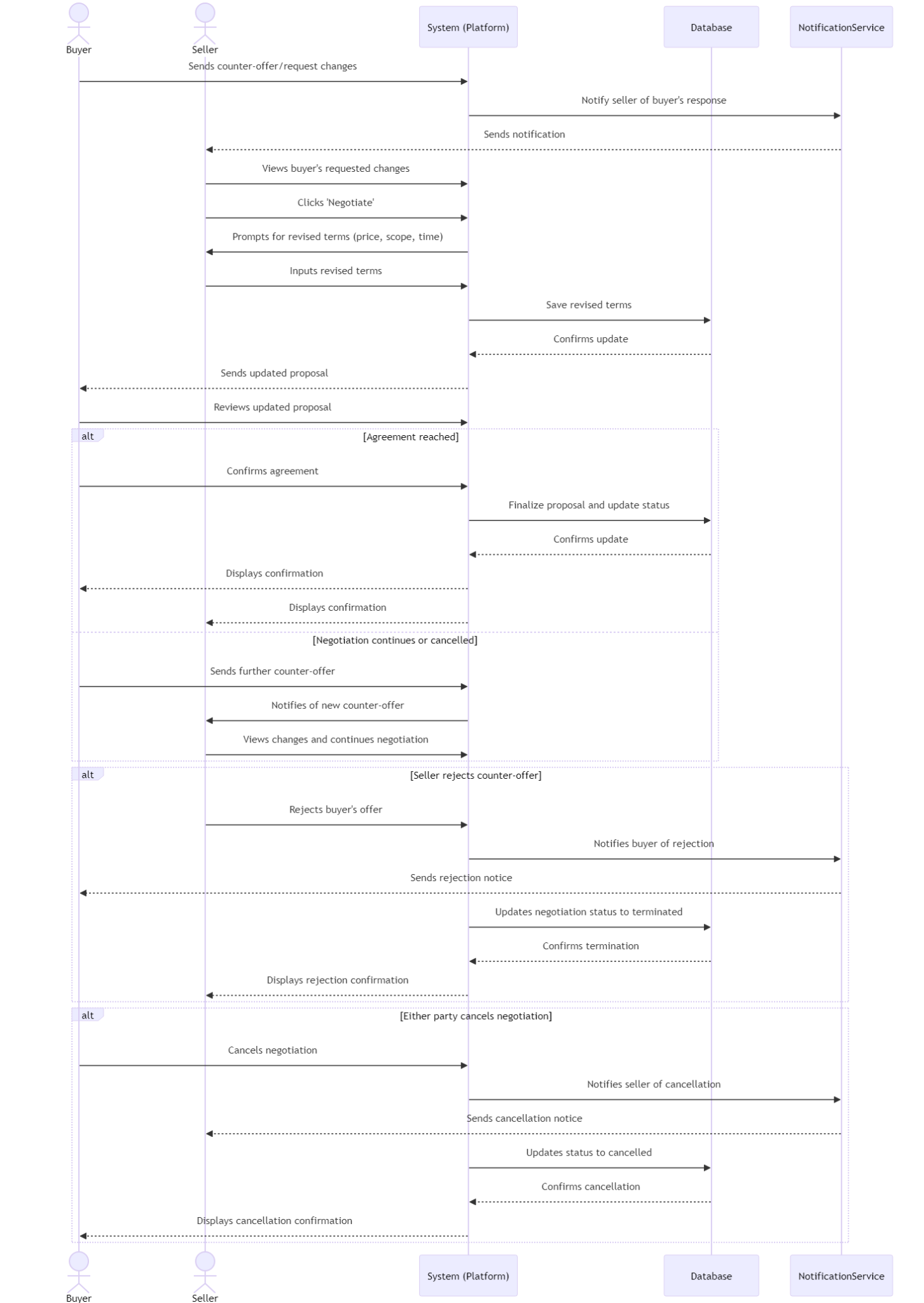
## List a Service for Selling



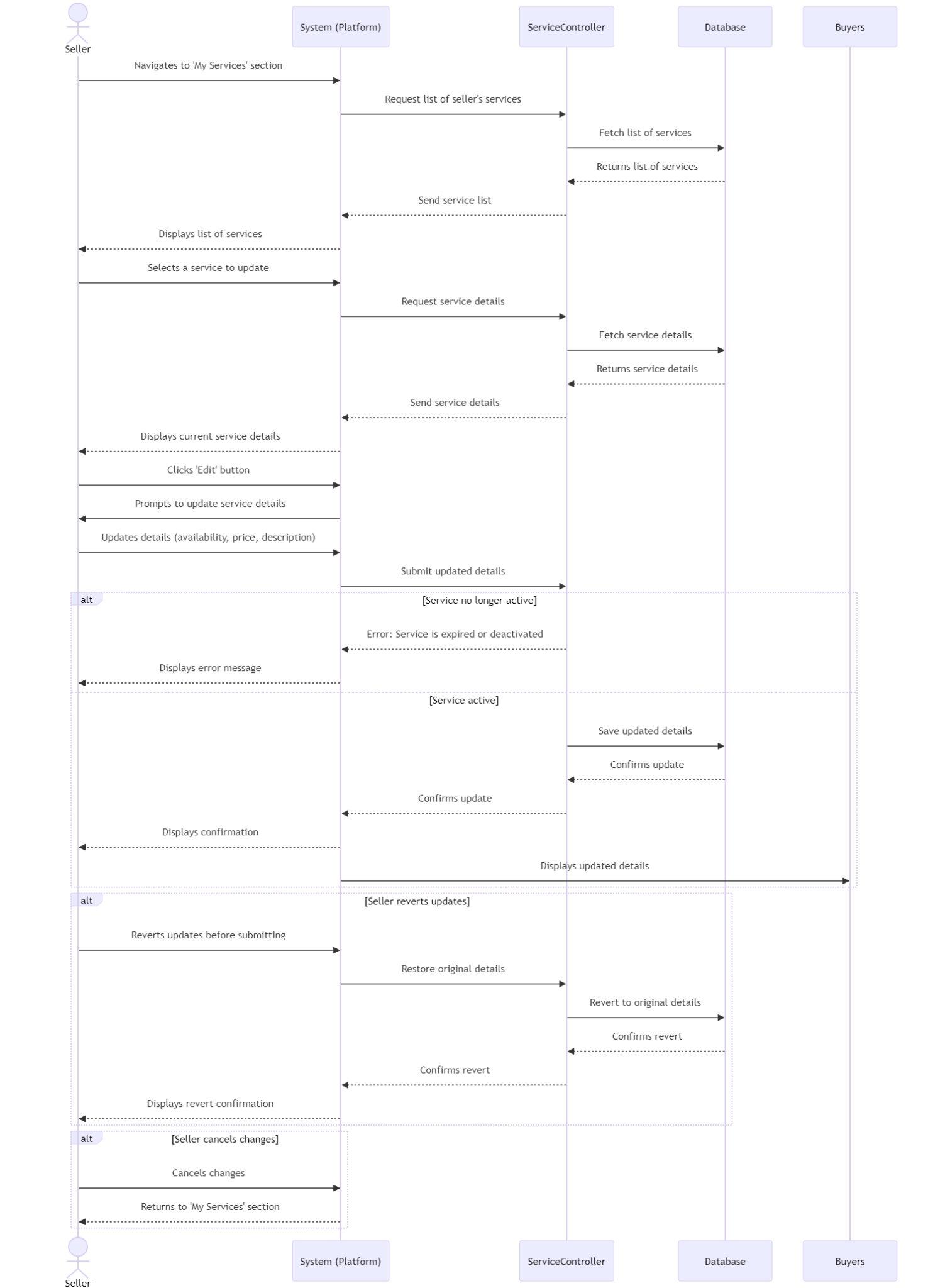
## Send Proposals to Buyers for Service Requests



## Negotiate Terms of Service with Buyer



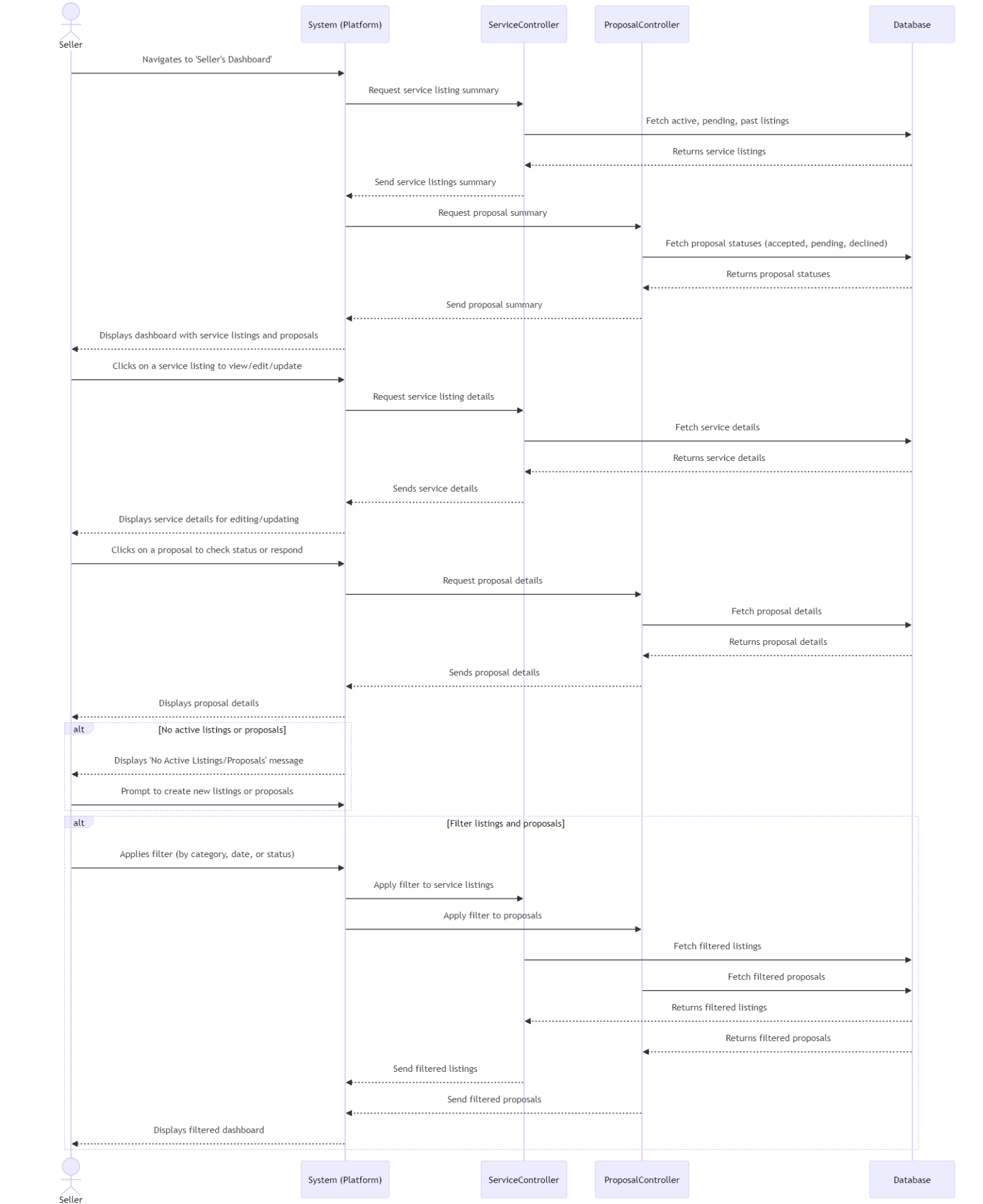
## Update Availability or Details of a Posted Service



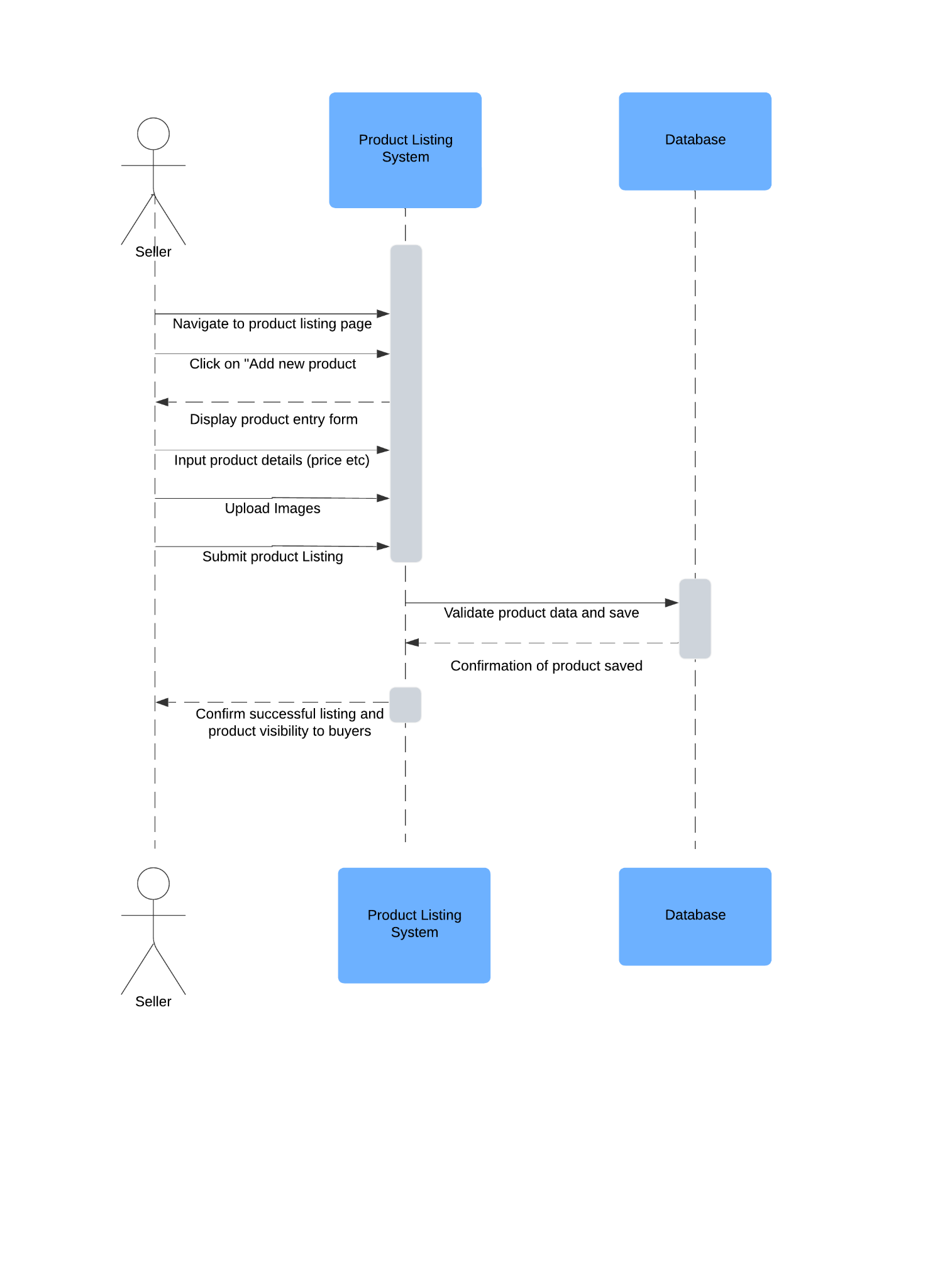
## Search for Buyer Listings by Sellers



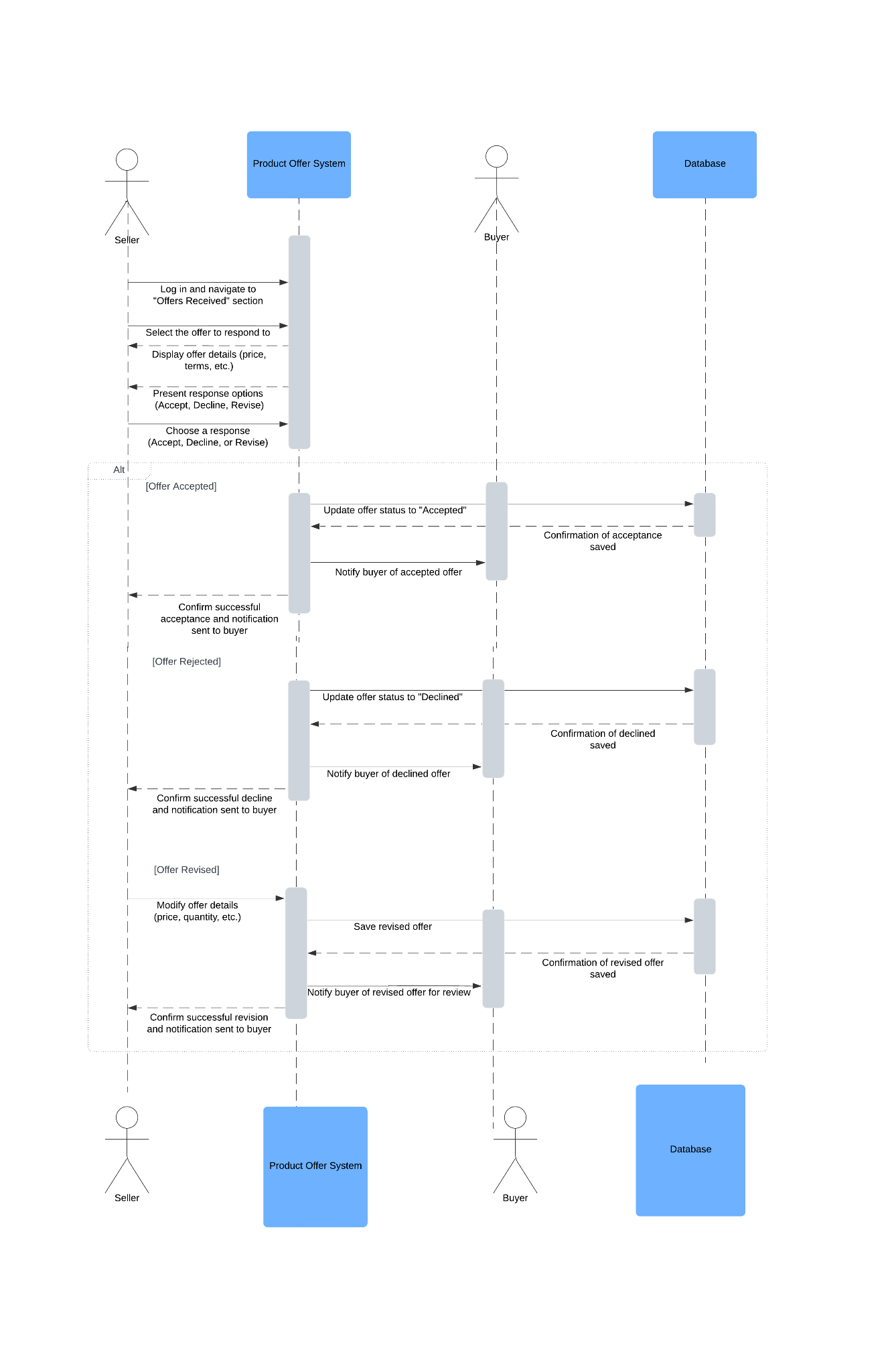
## View Dashboard for Sellers



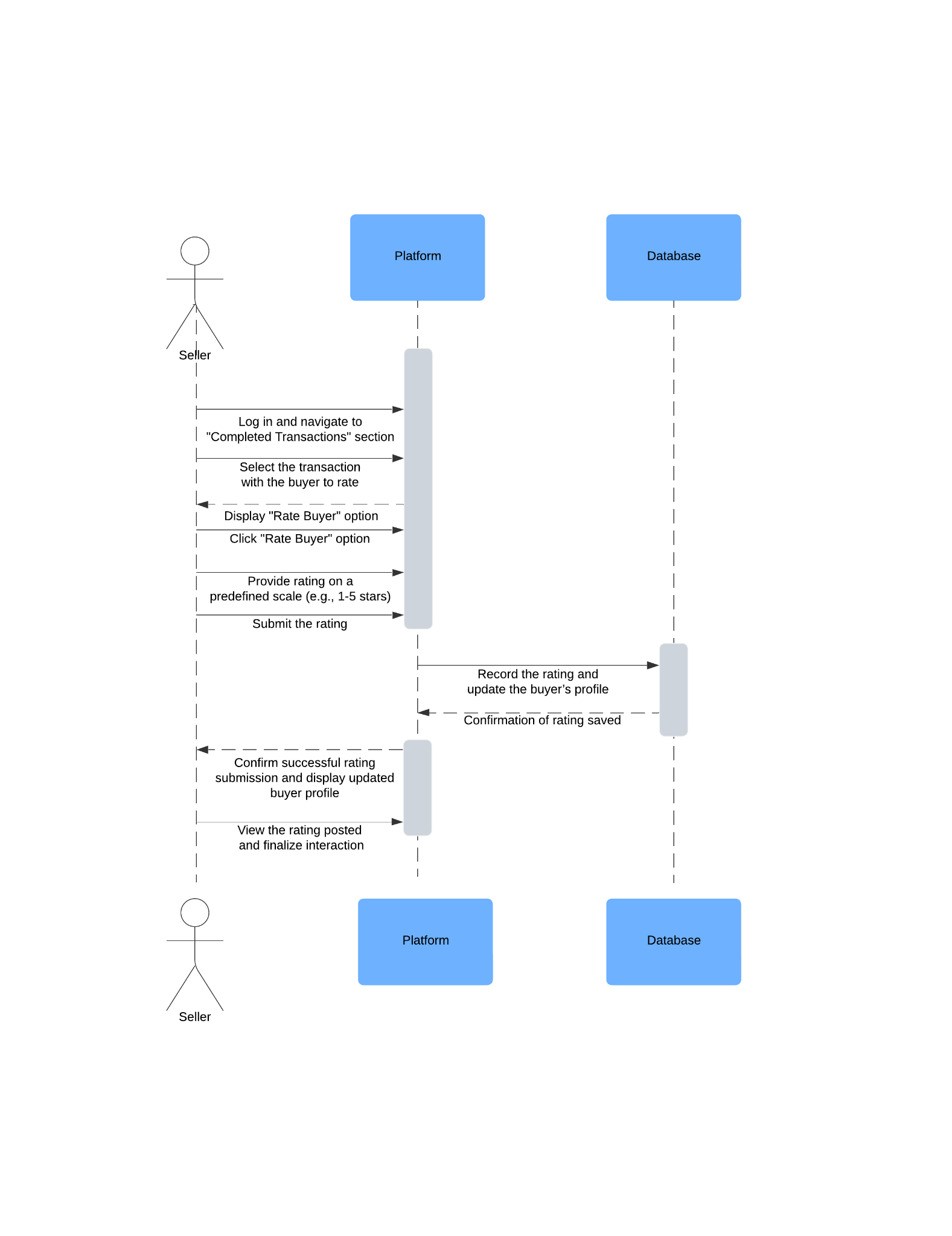
**5.18 Seller can list a tangible product**



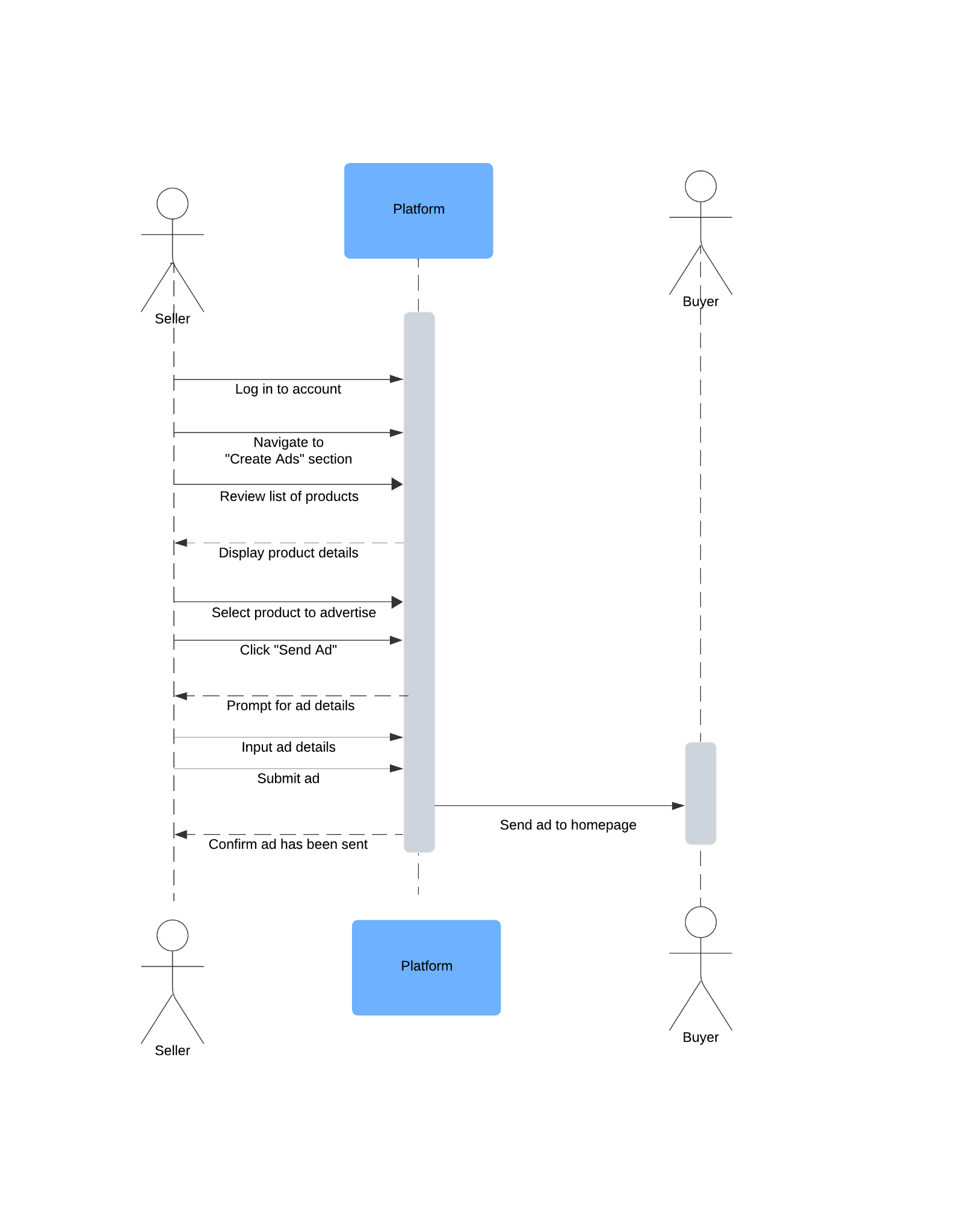
**5.19 Seller can respond to buyer’s offers**

****

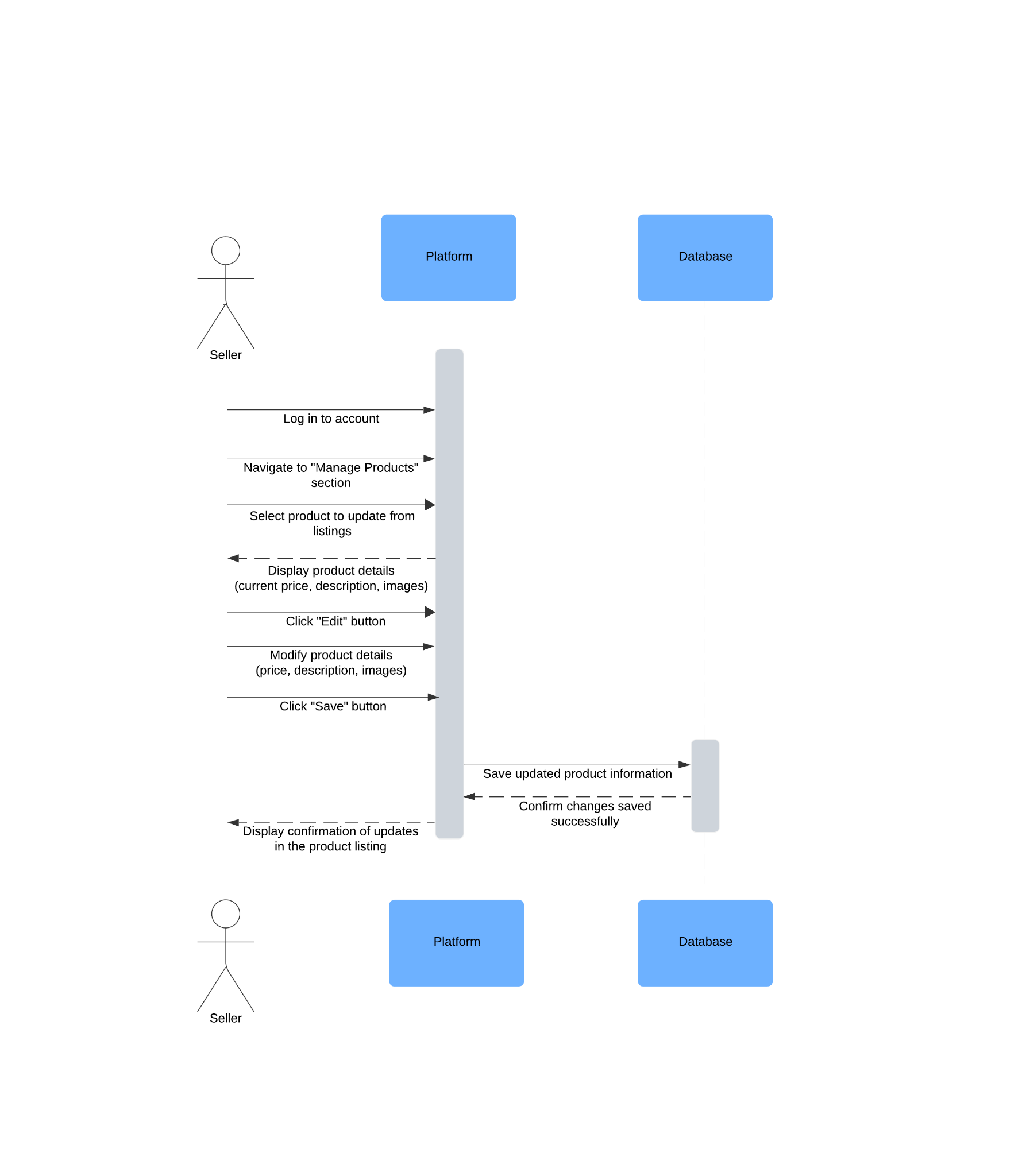
**5.20 Seller can rate buyers**

****

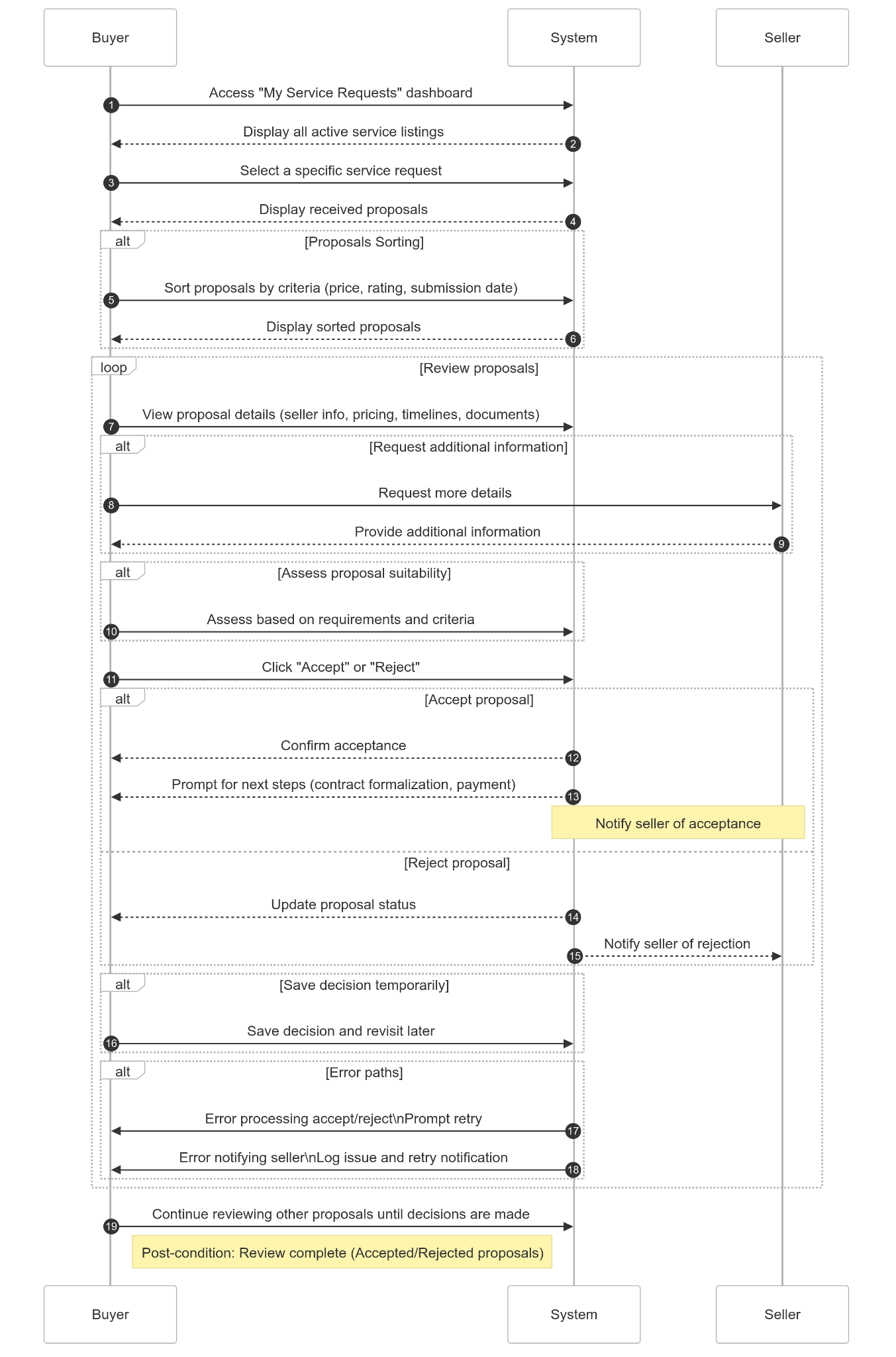
**5.21 Seller can send ad offers to buyers**

****

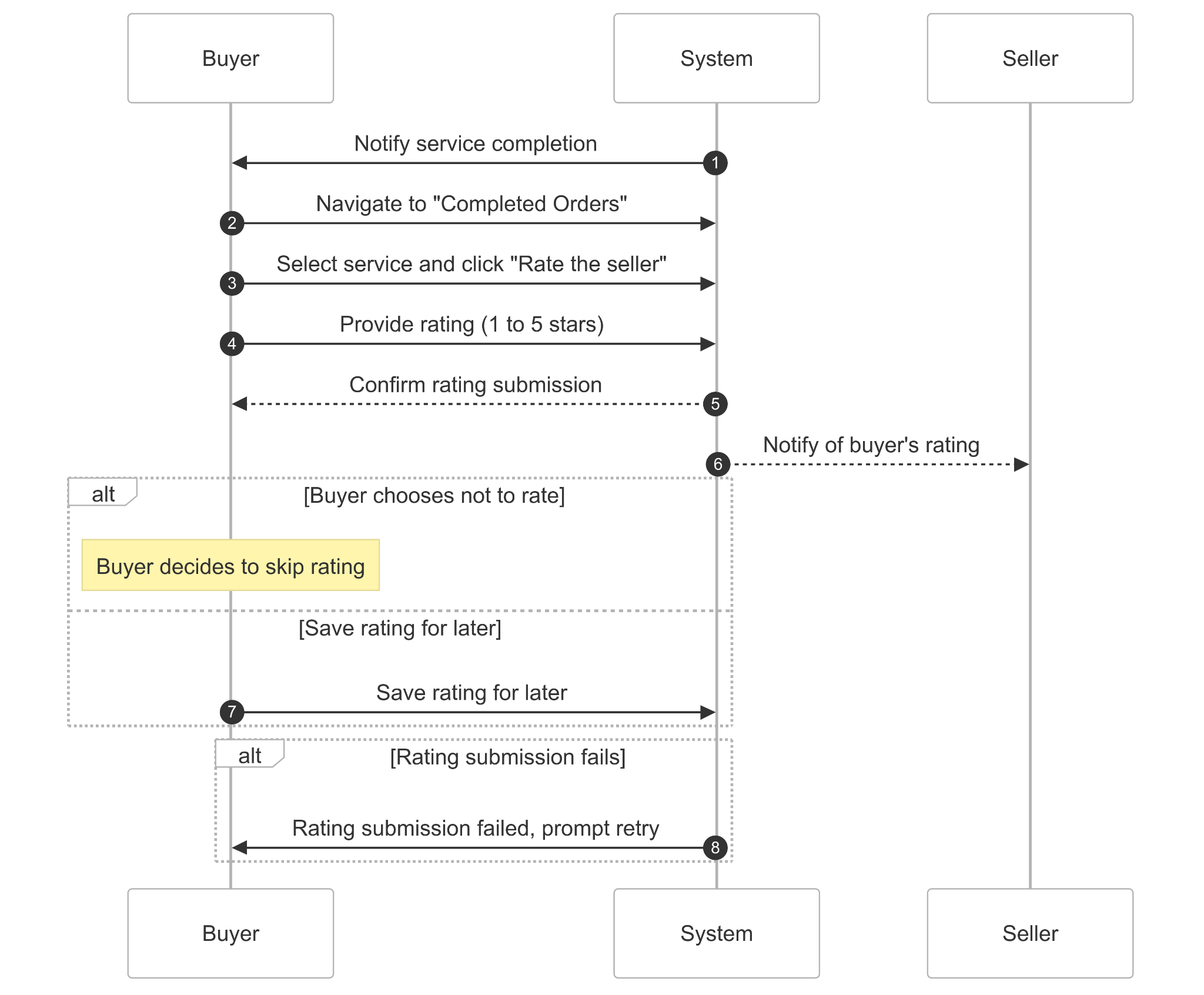
**5.22 Seller can update already listed product**

****

**5.23 Buyers review sellers proposals (accept/reject)**

****

**5.24 Buyers rating seller**

****

# State Diagrams

## Diagram details

**6.1.1 Create/Edit Listing State Machine**

| **State** | **Description** |
| --- | --- |
| Waiting | The system is waiting for user input. The display shows the current time. |
| NavigatingToForm | The user is directed to the form to create or edit a listing. The display shows the form. |
| FillingForm | The user is filling out the details of the listing. The display shows the form with input fields for details. |
| ReviewingDetails | The system displays the form for the user to review the entered details. The display shows a preview of the form with user input. |
| Validating | The system validates the data entered by the user. The display shows a loading or progress icon during the validation process. |
| DraftSaved | The listing is saved as a draft. The display shows a confirmation message indicating the draft has been saved. |
| DetailsPreview | The user previews the listing before final confirmation. The display shows a detailed preview of the listing. |
| PostingError | There is an error during the listing submission process. The display shows an error message indicating failure to post the listing. |
| Confirmation | The listing has been confirmed and published successfully. The display shows a confirmation message. |

| **Stimulus/Message** | **Description** |
| --- | --- |
| Waiting | The user navigates to the Create/Edit Listing page. |
| NavigatingToForm | The user begins the process of creating or editing a listing. |
| FillingForm | The user enters details into the form. |
| ReviewingDetails | The user reviews the details and submits the form. |
| Validating | The user has submitted the form, and the system starts validation. |
| DraftSaved | The user saves the listing as a draft. |
| DetailsPreview | The user views the preview of the listing. |
| PostingError | The system encounters an issue during listing submission. |
| Confirmation | The system confirms that the listing has been successfully submitted and published. |

**6.1.2 Offer State Machine**

| **State** | **Description** |
| --- | --- |
| OfferReceived | The seller has received an offer. The display shows details of the offer. |
| ReviewingOffer | The seller is reviewing the offer details. The display shows the offer for review. |
| RevisingOffer | The seller revises the offer. The display shows the modified offer details. |
| AcceptingOffer | The seller accepts the offer. The display shows a confirmation message indicating acceptance. |
| OfferAccepted | The offer has been accepted and confirmed. The display shows a confirmation message. |
| RejectingOffer | The seller rejects the offer. The display shows a rejection message. |
| OfferRejected | The offer has been rejected. The display shows a confirmation of rejection. |
| RevisedOfferSent | The seller sends a revised offer to the buyer. The display shows a confirmation of the revised offer being sent. |
| OfferExpiredOrWithdrawn | The offer has expired or been withdrawn. The display shows an error message indicating the offer is no longer valid. |

| **Stimulus/Message** | **Description** |
| --- | --- |
| OfferReceived | A buyer has submitted an offer. |
| ReviewingOffer | The seller starts to review the received offer. |
| RevisingOffer | The seller makes revisions to the offer. |
| AcceptingOffer | The seller accepts the buyer’s offer. |
| OfferAccepted | The system confirms successful acceptance of the offer. |
| RejectingOffer | The seller declines the offer. |
| OfferRejected | The system confirms that the offer has been rejected. |
| RevisedOfferSent | The seller has submitted a revised offer to the buyer. |
| OfferExpiredOrWithdrawn | The offer has either expired or been withdrawn by the buyer or seller. |

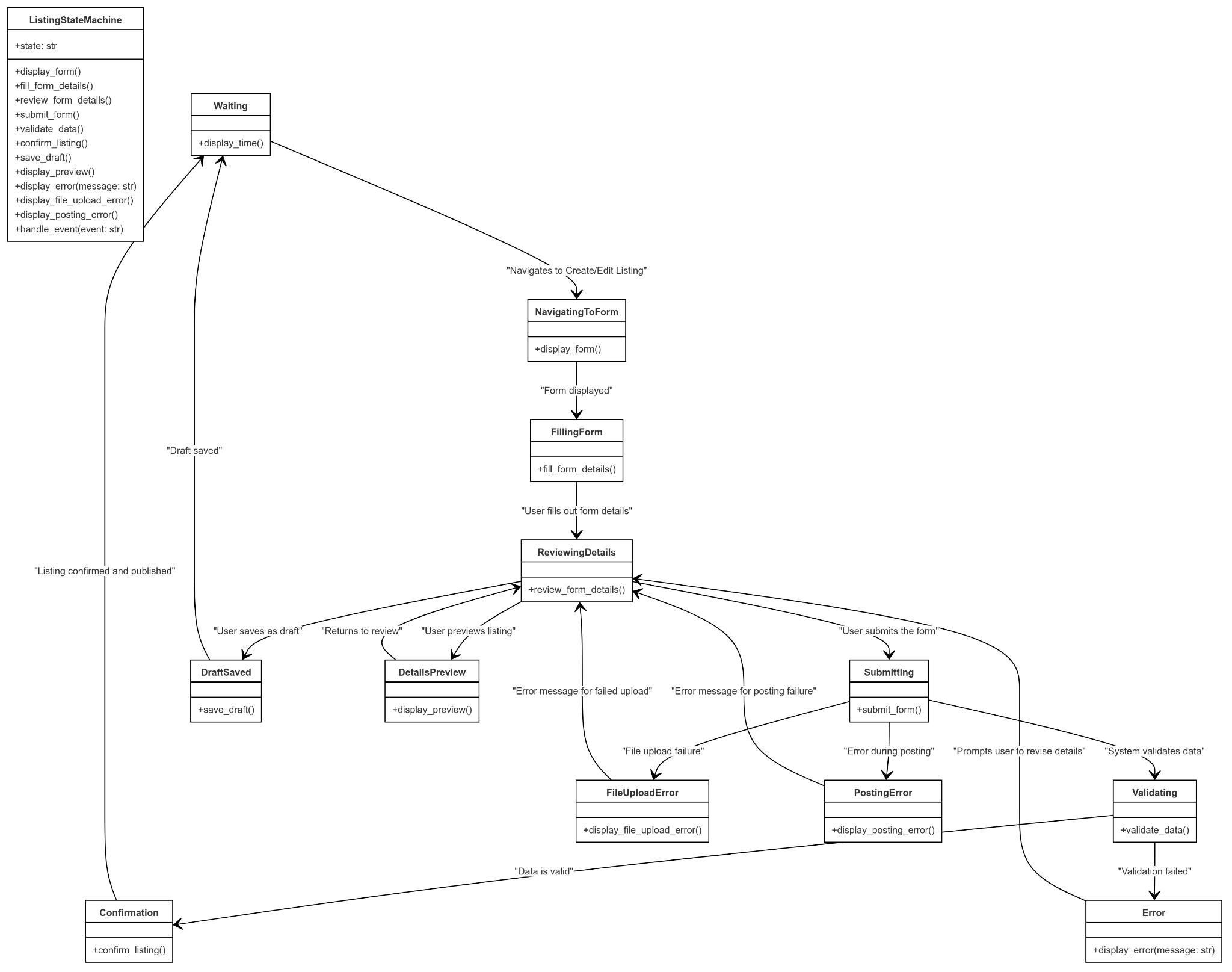
**6.1.3 Admin Pending Actions State Machine**

| **State** | **Description** |
| --- | --- |
| Waiting | The system is waiting for the admin to take action. The display shows the current time. |
| NavigatingToPendingActions | The admin navigates to the Pending Actions or Reported Items section. The display shows a list of pending actions or items for review. |
| ViewingPendingActions | The system displays a list of pending actions for the admin to review. The display shows all pending user actions, listing actions, or reported items. |
| ReviewingReportedItems | The admin reviews the details of a reported item. The display shows details of the reported item for review. |
| TakingAction | The admin takes action on a pending listing or reported item. The display shows a message indicating that an action is being processed. |
| TechnicalError | A technical error occurs while processing the action. The display shows an error message. |
| UserActionConfirmed | The user’s action has been confirmed. The display shows a confirmation message. |
| PostActionConfirmed | The post’s action has been confirmed. The display shows a confirmation message for the post-related action. |

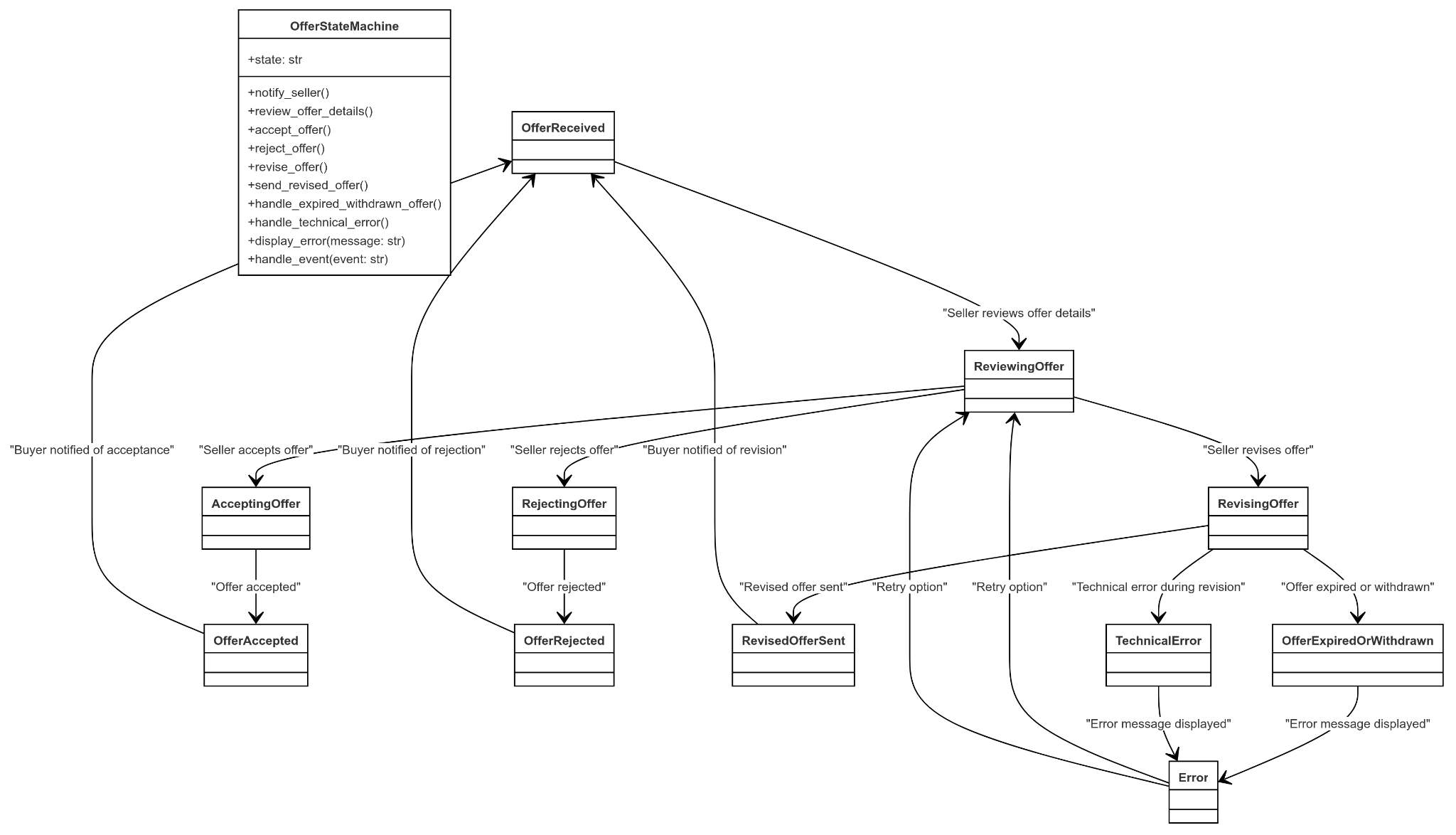
| **Stimulus/Message** | **Description** |
| --- | --- |
| Waiting | The admin is idle or reviewing pending actions. |
| NavigatingToPendingActions | The admin selects to review the pending actions or reported items. |
| ViewingPendingActions | The admin selects a specific item for review. |
| ReviewingReportedItems | The admin selects a reported item to review its details. |
| TakingAction | The admin approves or rejects a specific user action or listing action. |
| TechnicalError | The system encounters a technical issue during the action process. |
| UserActionConfirmed | The admin confirms the user's action. |
| PostActionConfirmed | The admin confirms the action related to a listing or post. |

## Diagram

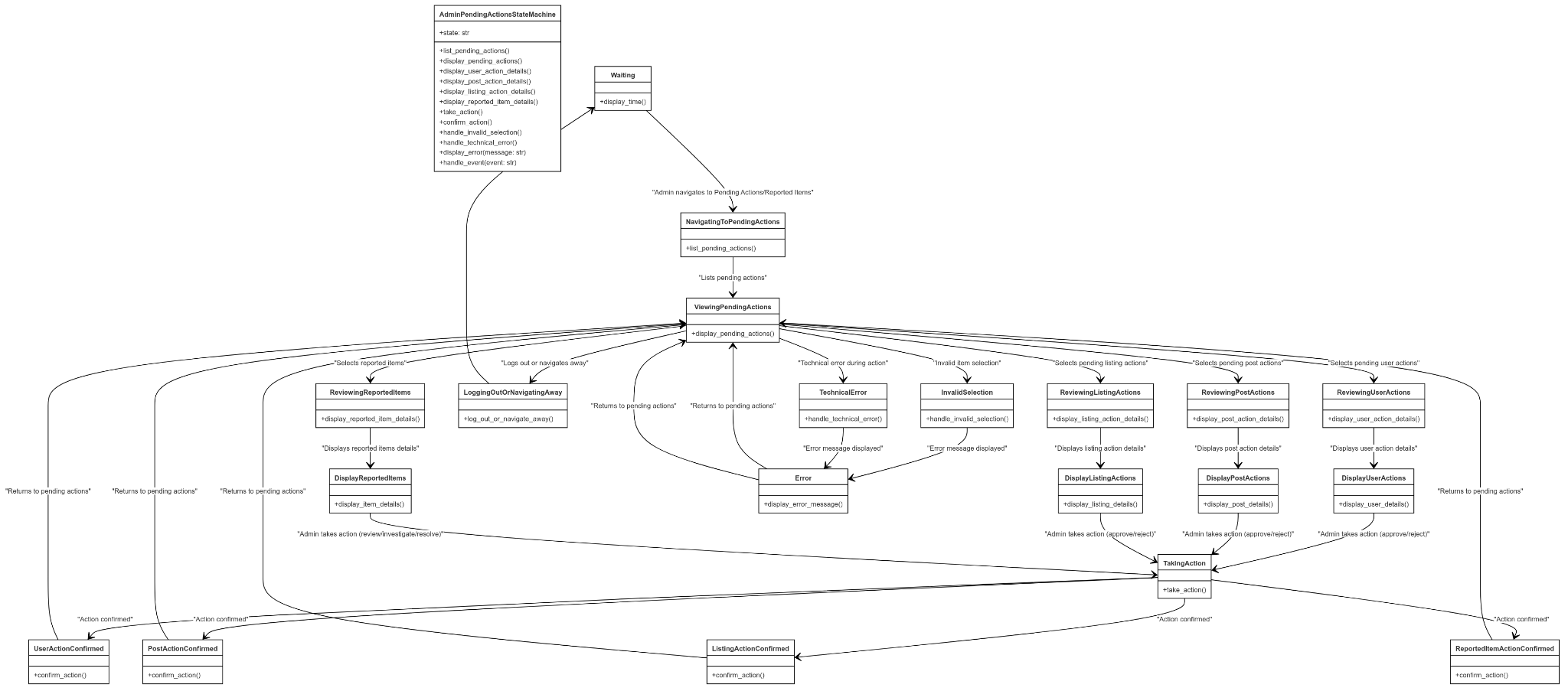
**6.2.1 Create/Edit Listing State Machine**

****

**6.2.2 Offer State Machine**

****

**6.2.3 Admin Pending Actions State Machine**



# Non-functional Requirements / Quality Attributes

| **Sr#** | **Requirements** |
| --- | --- |
| 1 | The system should not utilize more than 1 GB of memory at any time during its execution. |
| 2 | The system should not fail more than 3 times every 24 hours. In case of a failure, the system should restore to normal operations within 5 minutes of a failure. |
| 3 | The response time for search queries should be less than 2 seconds for 95% of the requests. |
| 4 | The user interface should be responsive and provide a consistent experience across different devices, including desktops, tablets, and smartphones. |
| 5 | The system should be accessible 24/7 with an uptime of 99.9% over any given month. |
| 6 | The system should process all buyer-seller communication and transactions with a latency of no more than 500 milliseconds to ensure smooth real-time interactions. |
| 7 | The system should provide user authentication and authorization mechanisms to ensure that users can only access their own data and features. |
| 8 | The system should be compatible with the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge. |
| 9 | The system should be designed to handle up to 1 million records in the database without significant performance degradation. |

# Who Did What?

| **Name of the Team Member** | **Tasks done** |
| --- | --- |
| Saad Ilyas | Use cases description and diagrams, Non functional requirements, Sequence diagrams |
| Abdul Ahad Bin Ali | Use Cases (Descriptions & Diagram), Class Diagram (Diagram & Description) |
| Hasan Malik | Use Cases (Descriptions & Diagram), Sequence Diagrams |
| Umer Jamil | Use cases, use case(description and diagram, sequence diagram, class diagram |
| Aniqa Aqeel | State Diagram, Use Cases (Descriptions & Diagram), Sequence Diagrams |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

| **Section** **Title** | **Reviewer Name(s)** |
| --- | --- |
| Introduction | Aniqa |
| Class diagram | Hasan Malik |
| Use case description | Saad Ilyas |
| Sequence Diagram | Abdul Ahad Bin Ali |
| Use Cases | Abdul Ahad Bin Ali |
| Use Cases | Umer Jamil |
| Class Diagram | Umer Jamil |
| State Diagram | Umer Jamil |