KING SAUD UNIVERSITY

COLLEGE OF COMPUTER AND INFORMATION SCIENCES

DEPARTMENT OF INFORMATION SYSTEM

GRADUATION PROJECT 1

Project title : FixIN

NAME	EMAIL	ID
Talal Saleh Alharbi	439101900@student.ksu.edu.sa	439101900
Khalid Abdullah Alajlan	437105372@student.ksu.edu.sa	437105372
Muath Abdulaziz Alshathri	439102630@student.ksu.edu.sa	439102630
Mohammed omar aljarallah	439101755@student.ksu.edu.sa	439101755

Table of Contents

Table of Figures

Table of Tables

1. Project Abstract:

In this project focuses on a web application named “FixIN” that helps individuals who seek assistance from shops in regard to their smart device repair needs.
The purpose of this web application will allow users to search and connect with shops who request to fix and the user’s smart device issue. This helps save time and effort spent looking for local shops who can fix the issue.

2. Introduction:

The main purpose of this project is to design and develop a web application that’ll allow smart device users to send in requests for repair to all smart device shops that are willing to fix them, the request will contain a description of the problem and will be categorized for the shops to see and to respond to the user’s request with a price and a preliminary solution for the issue that the user is currently facing.
The user begins by selecting their device’s model and manufacturer, and then the user is prompted to type in the issue that the device faces in detail and will be required to attach images or videos of said issue. Then, the user will submit the request and will await the response for repair by the shops.
The shop then receives a notification of new requests. Then they will be able to check out the request listings for the categories that they specialize in. Afterwards, the shop picks a request and starts a chat, in which both the shop and the customer go and discuss the issue in detail. The shop may offer a preliminary solution and a starting price for the fix. The user then can negotiate with the shop.

3. Objectives:

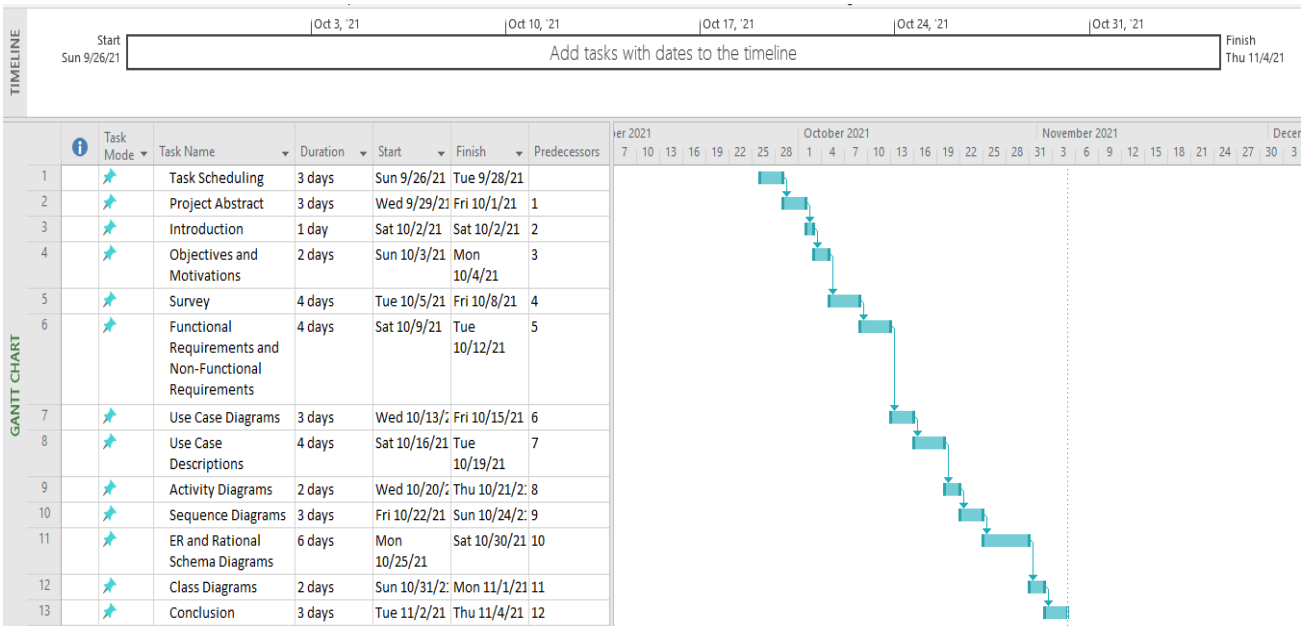
To help the user save time and effort and allow the user to get the best price possible by negotiating with the shops. Users can also receive multiple solution offers from different shops, giving the user the chance to pick the solution and the shop the suits them best.

4. Motivation:

Our main motivation is to help reduce the time spent looking and negotiating with shops in-person, as it can sometimes lead the user to be frustrated with not finding the right price or solution after spending so much time and effort. We also realized that no similar idea exists yet, so we try to use this opportunity to help as many people as we can.

5. Project Time Scheduling:

Managing the time is the most important skill for the team, it's a scale if the team capable to achieve the goal in certain time or not.



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Task Scheduling	3 days	Sun 9/26/21	Tue 9/28/21	
2		Project Abstract	3 days	Wed 9/29/21	Fri 10/1/21	1
3		Introduction	1 day	Sat 10/2/21	Sat 10/2/21	2
4		Objectives and Motivations	2 days	Sun 10/3/21	Mon 10/4/21	3
5		Survey	4 days	Tue 10/5/21	Fri 10/8/21	4
6		Functional Requirements and Non-Functional Requirements	4 days	Sat 10/9/21	Tue 10/12/21	5
7		Use Case Diagrams	3 days	Wed 10/13/2	Fri 10/15/21	6
8		Use Case Descriptions	4 days	Sat 10/16/21	Tue 10/19/21	7
9		Activity Diagrams	2 days	Wed 10/20/2	Thu 10/21/2	8
10		Sequence Diagrams	3 days	Fri 10/22/21	Sun 10/24/2	9
11		ER and Rational Schema Diagrams	6 days	Mon 10/25/21	Sat 10/30/21	10
12		Class Diagrams	2 days	Sun 10/31/2	Mon 11/1/21	11
13		Conclusion	3 days	Tue 11/2/21	Thu 11/4/21	12

FIGURE 1 TIME SCHEDULING

6. System development methodology:

We used in this project waterfall development methodology, which is sequential development approach, in which development is seen flowing downwards (like a waterfall) through several phases.

7. Survey:

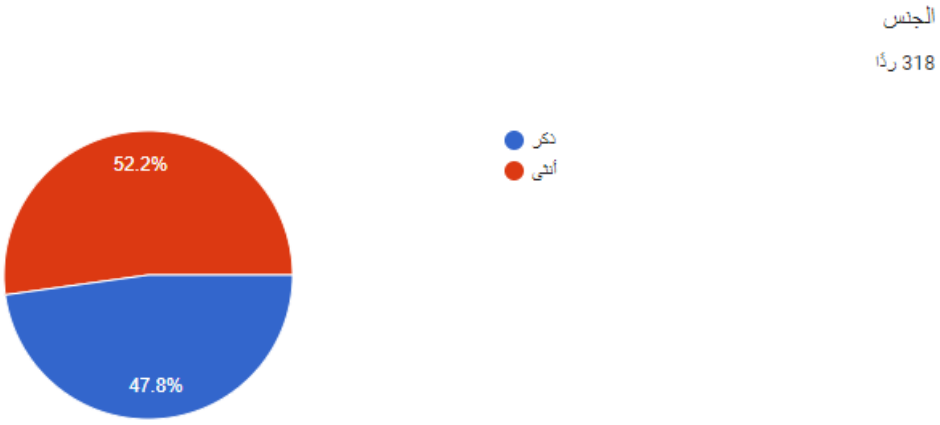


FIGURE 2 SURVEY (1 OF 7)

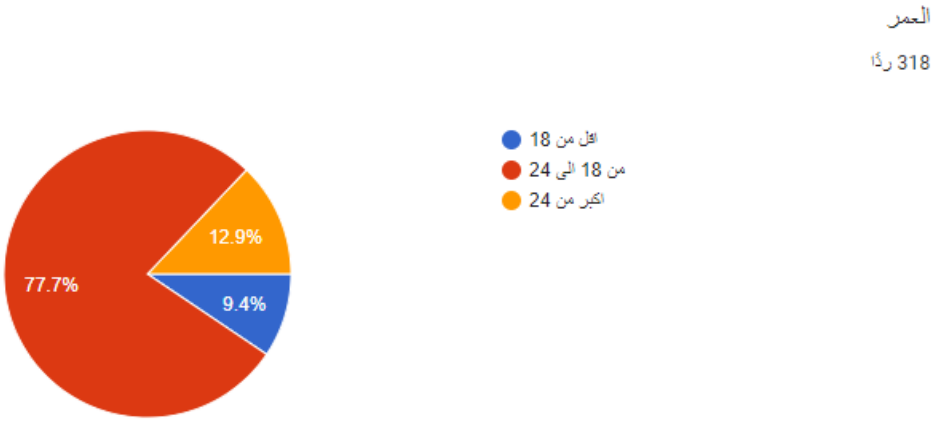


FIGURE 3 SURVEY (2 OF 7)

For the first two questions, we intended to collect their ages and sex as to help understand the demographic interested in this project.

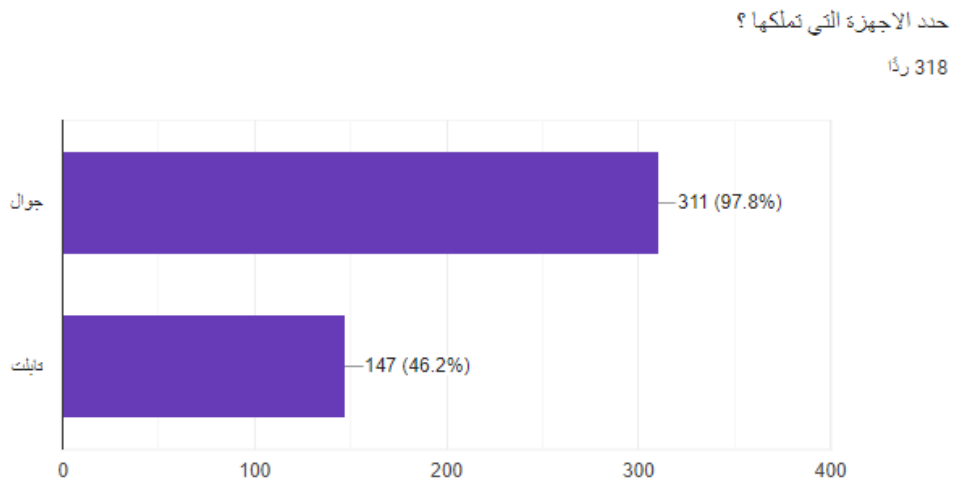


FIGURE 4 SURVEY (3 OF 7)

This question helps us show the type of device they usually use, as to help us accommodate them in the web application and set up services for both types of devices.

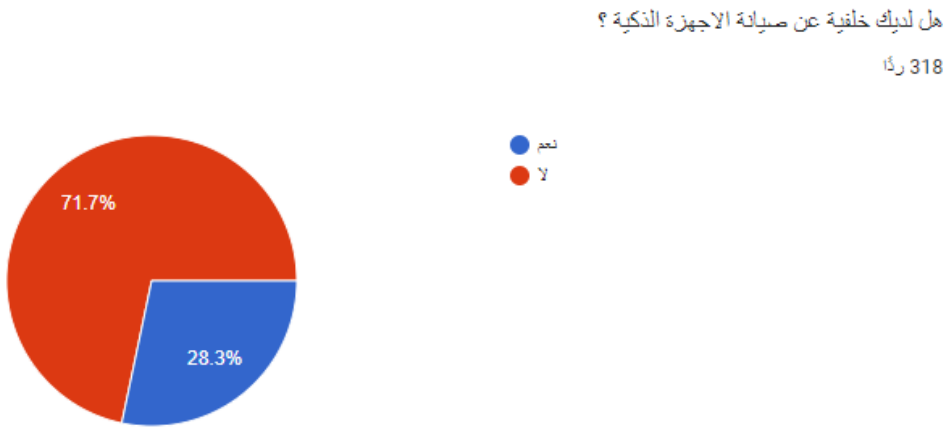


FIGURE 5 SURVEY (4 OF 7)

This question touches on whether the user has ever tried fixing a device—professionally or not—letting us decide on the kind of terminology used in the application, technical or general.

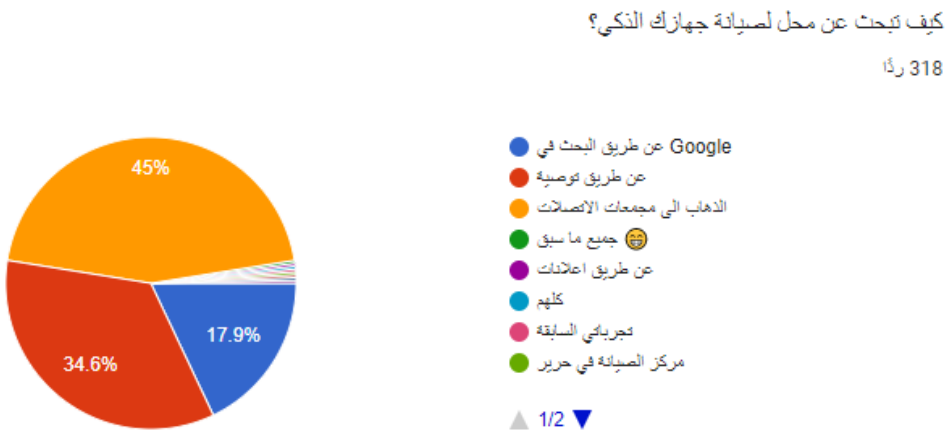


FIGURE 6 SURVEY (5 OF 7)

In this question, we wanted to learn more about how the user got to find shops to fix their devices. Seeing as the majority went to go look for shops physically, we were confident that the web application will help them in cutting that time short.

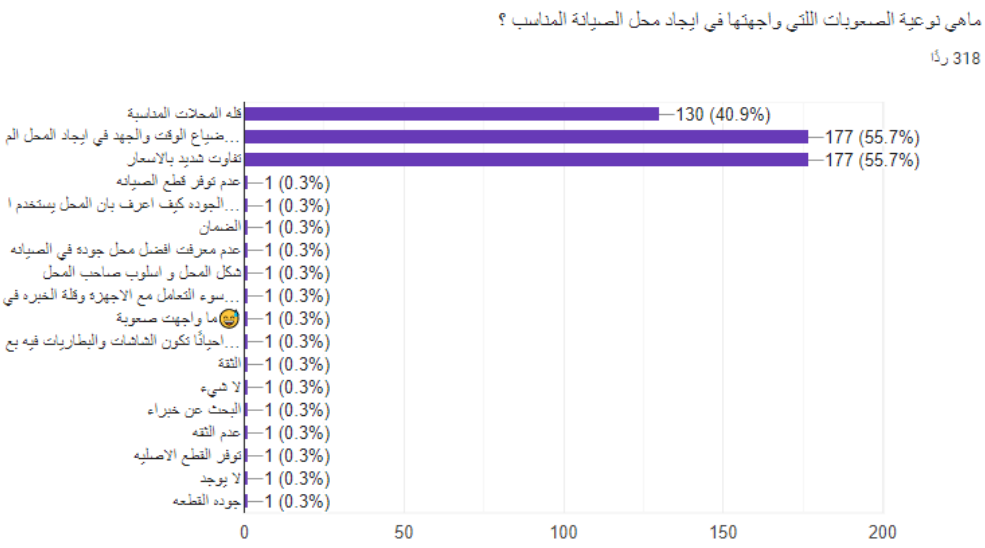


FIGURE 7 SURVEY (6 OF 7)

In this question, we wanted to figure out what problems users faced when trying to find a shop and negotiate. This helps us improve our web application by working on our search and negotiation functions.

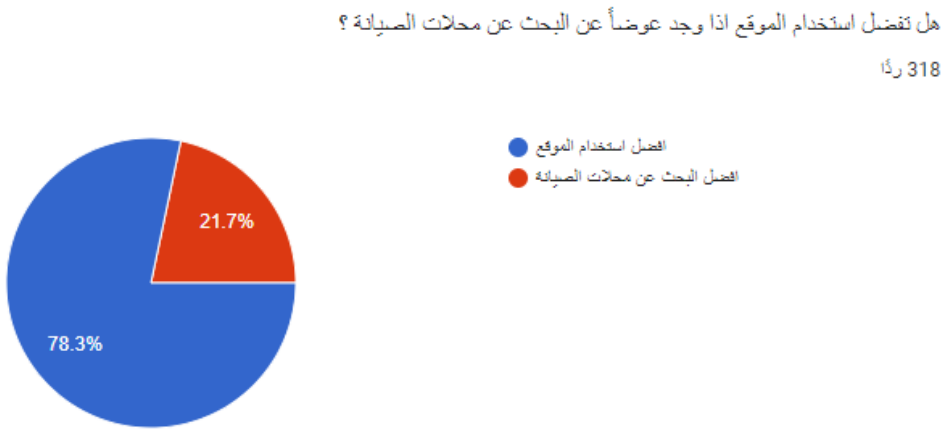


FIGURE 8 SURVEY (7 OF 7)

In this final question, we wanted to gauge interest in our project. Seeing as an overwhelming majority prefers this idea over other methods, we became confident in our work.

8. Positional Users:

Customers : Who wants to repair their devices.
Shops : Who can help Customers to repair their devices.

9. Functional Requirements and Non-functional Requirements:

9.1 Functional Requirements:

- **For Customer:**

- The customer will be able to create an account.
- The customer will be able to login and logout.
- The customer will be able to edit their account.
- The customer will be able to create an issue
- The customer will be able to add details to their issue (such as description and category).
- The customer will be able to pay a verification fee.
- The customer will be able to delete a submitted issue
- The customer will be able to review shop requests.
- The customer will be able to select a request.
- The customer will be able to start a negotiation with the shop.
- The customer will be able to approve or decline a request.
- The customer will be able to rate a shop.

- **For Shop:**

- The shop will be able to create an account.
- The shop will be able to login and logout.
- The shop will be able to edit their account.
- The shop will be able to browse submitted issues.
- The shop will be able to send a request with the solution for the issue.
- The shop will be able to negotiate with a customer.

- **For Admin:**

- The admin will be able to login and logout.
- The admin will be able to display a list of registered accounts.
- The admin will be able to send messages to an account.
- The admin will be able to search for an account.
- The admin will be able to block or unblock an account.
- The admin will be able to display logged negotiation sessions (chat history).

9.2 Non-functional Requirements:

- **Security:** Means that the system is safeguarded against deliberate and intrusive faults from internal and external sources.
- **Performance:** The system should provide high interaction with users and find shops easily and accurately.
- **Usability:** Which is the ease of which the user can learn, operate, prepare inputs, an interpret outputs through interaction with a system.
- **Reliability:** How well the software system consistently performs the specified functions without failure.
- **Scalability:** The system should be able to increase it storage capacity to meet demand.
- **Flexibility:** The system should be able to be used frequently, and it should be able to handle any new updates or any modifications.
- **Maintainability:** The system should be able to be maintained occasionally.

10. System analysis and design:

10.1 Use Case Diagrams:

Use Case is a way used in system development to organize and identify the system requirements/functions. Any outside entity is represented as an actor. Each actor is connected to a one or more use case; each use case represents an activity that the actor can perform. The following Diagrams represent all use case Diagrams in our project for each different actor:

For Customer:

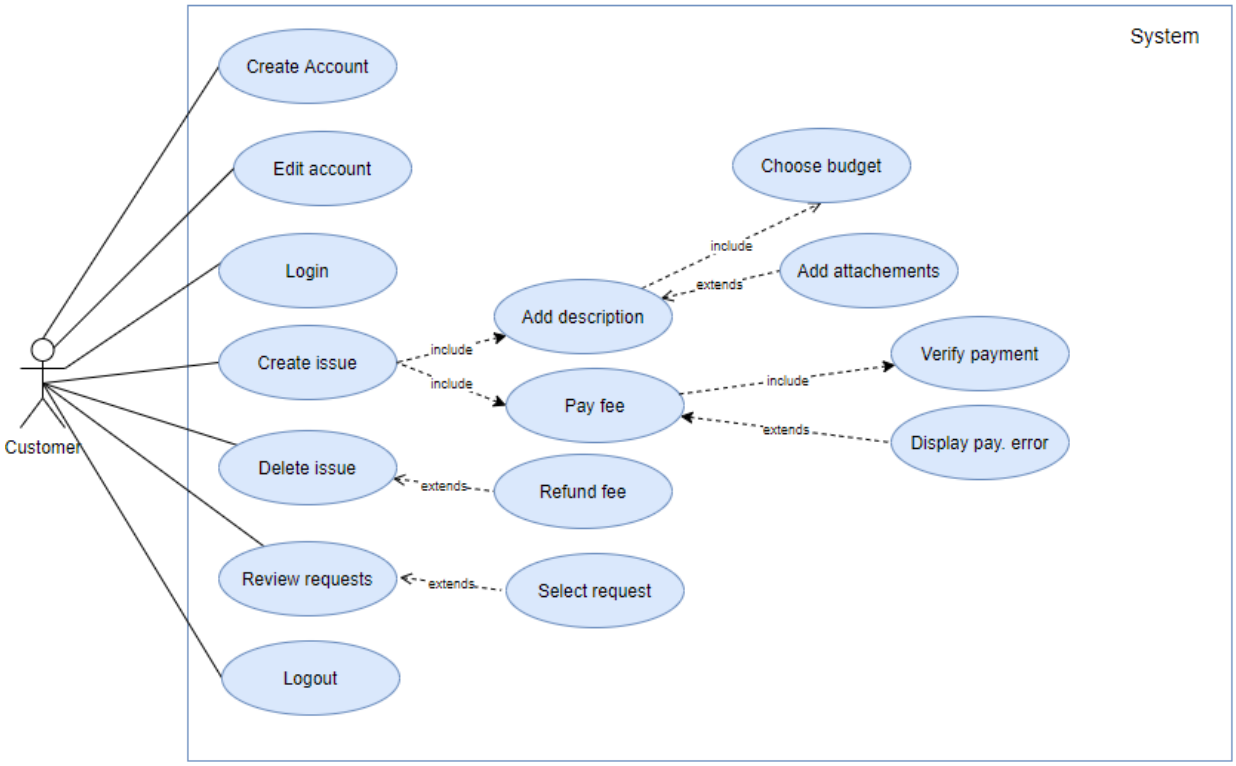


FIGURE 9 CUSTOMER USE CASE

For Shops:

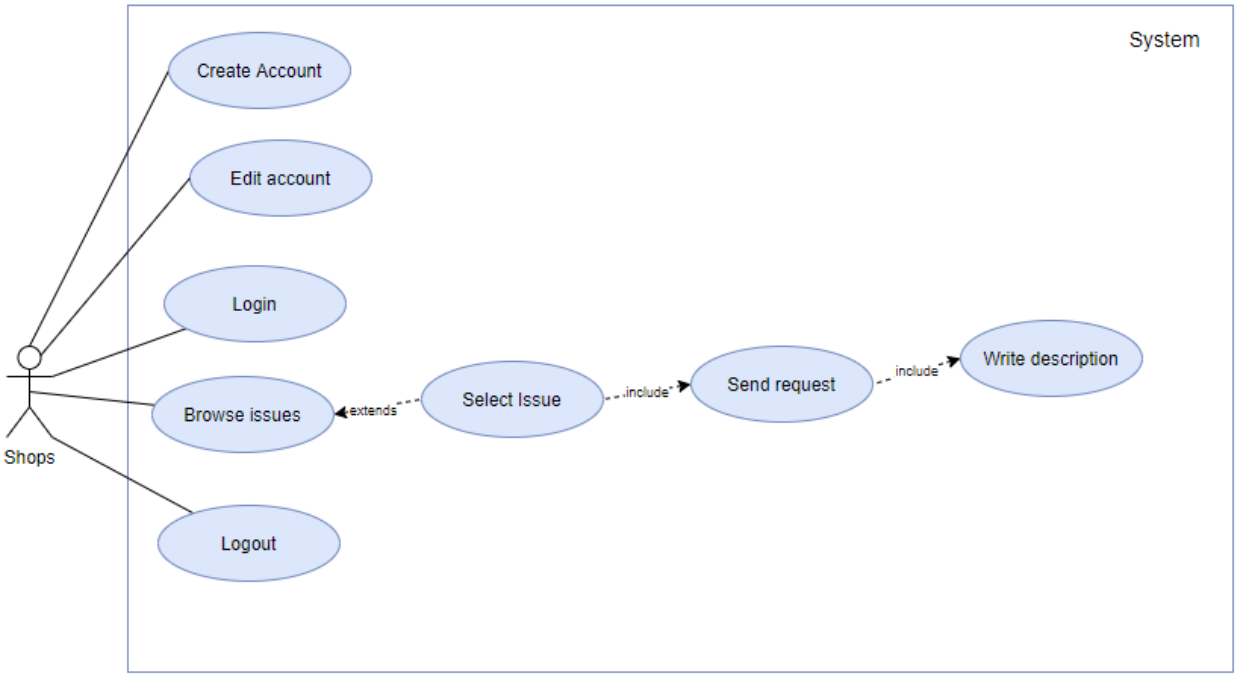


FIGURE 10 SHOPS USE CASE

For Customer and Shops:

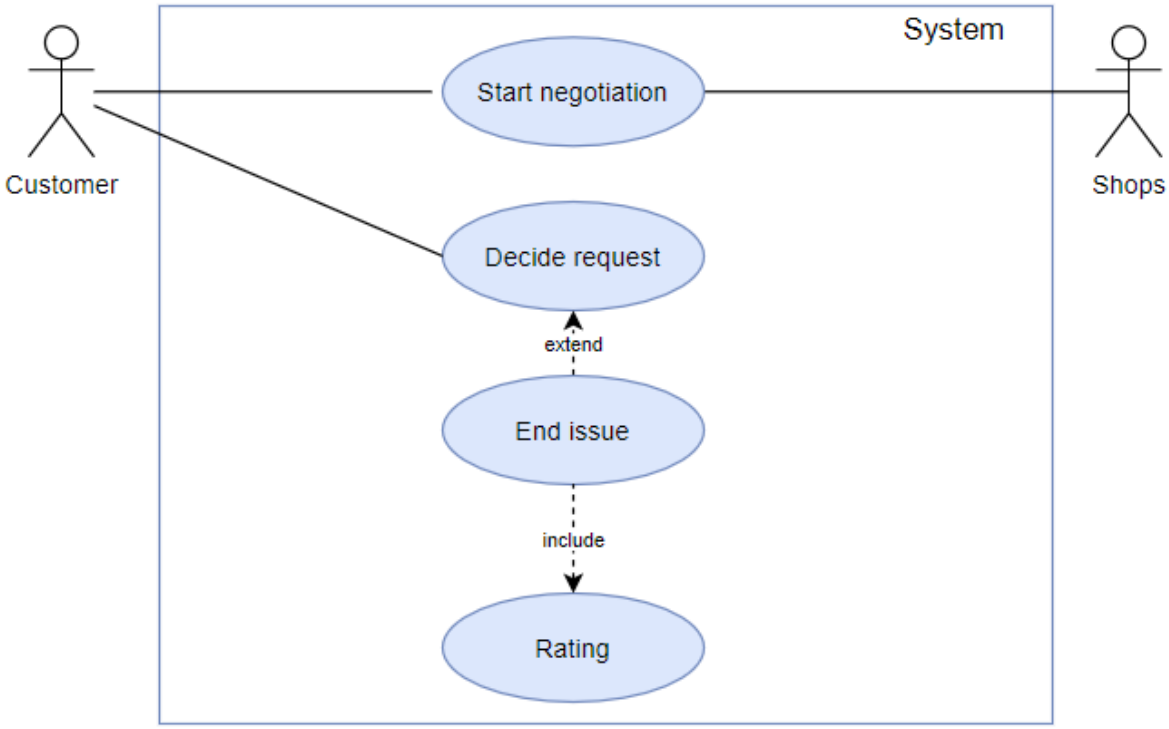


FIGURE 11 CUSTOMER AND SHOPS USE CASE

For Admin:

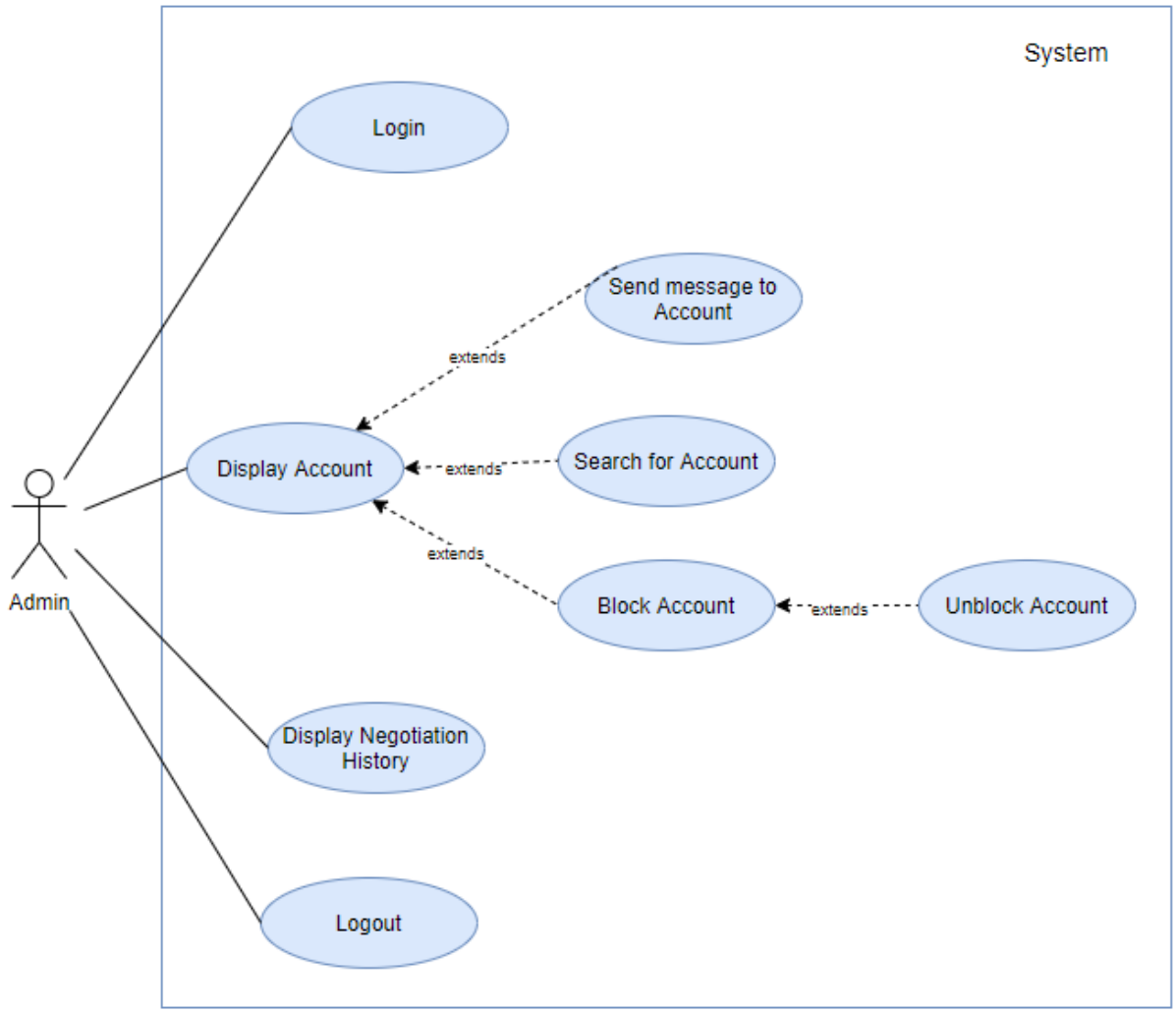


FIGURE 12 ADMIN USE CASE

10.2 Use Case Descriptions:

Use Case Name:	Creating issues.	
Scenario:	The customer will Create issue of their problem.	
Triggering Event:	The customer faces an issue with their device and wants to enlist help.	
Brief Description:	The customer will write the description of the issue, pay fee and send the repair request to the shops.	
Actors:	1- Customer.	
Related Use Cases:		
Stakeholders:	1- Customer. 2- Shop.	
Preconditions:	Customer must have an account in the system.	
Post conditions:	The issue is posted successfully.	
Flow of Activities:	Actor	System
	1. The customer will post request That include a description. 2. customer pays fee.	1.1 System display payment menu. 2.1 System verifies payment. 2.2 System posts request.
Exception Conditions:	1.1 Post is empty. 2.2 Customer enter invalid payment.	

TABLE 1 "CREATING ISSUES" USE CASE DESCRIPTION

Use Case Name:	Reviewing requests.	
Scenario:	The customer will review requests sent by the shops and then select the appropriate request.	
Triggering Event:	The customer receives repair requests by the shops.	
Brief Description:	The customer will get requests from shops and review them and select the best offer that suit him.	
Actors:	1- Customer.	
Related Use Cases:	Creating issue.	
Stakeholders:	1- Customer. 2- Shop.	
Preconditions:	Issue created.	
Post conditions:	The customer will find the shop that will help him to find solution for the issue.	
Flow of Activities:	Actor	System
	1. The customer will review requests that comes from shops. 2. The customer will select the request.	1.1 System will display the descriptions that sent by every request from shops. 2.1 System will connect the customer with shop. 2.2 System starts chat session.
Exception Conditions:	1.1 No shop has made a request yet.	

TABLE 2 "REVIEWING REQUESTS" USE CASE DESCRIPTION

Use Case Name:	Start negotiation.	
Scenario:	The customer and the shop will start negotiation about the repair.	
Triggering Event:	The system starts a chat session with customer and shop.	
Brief Description:	After selecting the appropriate shop, the system will start a chat session with both customer and shop to start negotiation about the price and how to fix the customer device.	
Actors:	1- Customer 2- Shop	
Related Use Cases:	Reviewing requests.	
Stakeholders:	1- Customer. 2- Shop.	
Preconditions:	Customer selected the shop.	
Post conditions:	The issue will be closed based on customer entry.	
Flow of Activities:	Actor	System
	1. Customer and shop will start negotiation. 2. Customer will decide to repair from this shop or no. 3. Customer will fill the rating form.	1.1 System logs chats into database. 2.1 System approve customer entry. 2.2 System will end the issue and display rating form. 3.1 System will save the rating form in the database.
Exception Conditions:	3.1 form is not filled.	

TABLE 3 "START NEGOTIATION" USE CASE DESCRIPTION

Use Case Name:	Browsing issues.	
Scenario:	The shop will browse the request of customer.	
Triggering Event:	The shop wants to check out the issues posted by customer.	
Brief Description:	The shop will send a request to customer that include description of the solution.	
Actors:	1- Shop.	
Related Use Cases:	Creating Issues.	
Stakeholders:	1- Shop. 2- Customer.	
Preconditions:	Existing requests from the customers.	
Post conditions:	The customer will receive solutions from the shops.	
Flow of Activities:	Actor	System
	1. Shop requests issue listings. 2. Shop will select one of the issues from the list. 3. Shop sends a brief description of the solution.	1.1 System will display to the shop a list of issues currently available. 2.1 System will connect the shop with customer. 2.2 System starts a chat session. 3.1 System will send the solution to the customer
Exception Conditions:	1.1 No issues available	

TABLE 4 "BROWSING ISSUES" USE CASE DESCRIPTION

Use Case Name:	Creating an account.	
Scenario:	Actor registers into the system for access.	
Triggering Event:	When the actor wants to obtain access to the web application.	
Brief Description:	The actor registers into the system by using a basic information such as (First name, last name, age, phone number, e-mail and password).	
Actors:	1- Customer. 2- Shop.	
Related Use Cases:	Logging in.	
Stakeholders:	1- Customer. 2- Shop.	
Preconditions:	The phone number or email do not exist in the database.	
Post conditions:	Account created.	
Flow of Activities:	Actor	System
	1. The actor enters their info. 2. Actor verifies ownership of details via SMS.	1.1 System will compare details in the database to avoid duplicates. 2.1 System registers the actor into the accounts' database.
Exception Conditions:	1.1 Same details already exist in the database	

TABLE 5 "CREATING AN ACCOUNT" USE CASE DESCRIPTION

Use Case Name:	Logging in.	
Scenario:	Actor logs into the web application.	
Triggering Event:	Actor needs the services of the web application.	
Brief Description:	The actor will enter their registered details to authenticate and gain access into the system.	
Actors:	1- Customer. 2- Admin. 3-Shop.	
Related Use Cases:	Creating an account, Logging out.	
Stakeholders:	1- Customer. 2- Admin. 3- Shop.	
Preconditions:	Actor must have successfully registered an account before.	
Post conditions:	The actor gains access to the system.	
Flow of Activities:	Actor	System
	1. Actor enters in their account credentials.	1.1 System compares the entered details with the accounts' database. 1.2 System allows access for the actor.
Exception Conditions:	1.1 The login information is wrong (Entering wrong or false credentials).	

TABLE 6 "LOGGING IN" USE CASE DESCRIPTION

Use Case Name:	Logging out.	
Scenario:	Actor logs out and ends the session.	
Triggering Event:	The actor wants to finish their session.	
Brief Description:	The actor confirms the log out prompt, thus ending the session and revoking further access until next login.	
Actors:	1- Customer. 2- Admin. 3- Shop.	
Related Use Cases:	Logging In.	
Stakeholders:	1- Customer. 2- Admin. 3- Shop.	
Preconditions:	Must be already logged in with a valid account.	
Post conditions:	The actor is logged out.	
Flow of Activities:	Actor	System
	1. Actor requests logging out. 2. Confirm log out.	1.1 System prompts for customer's confirmation. 2.1 System ends the actor's session, logs them out.
Exception Conditions:		

TABLE 7 "LOGGING OUT" USE CASE DESCRIPTION

Use Case Name:	Displaying Accounts.	
Scenario:	Admin views account and their details.	
Triggering Event:	The admin needs to view accounts and their details or block their accounts or send message to accounts.	
Brief Description:	The admin gains access to a dashboard that allows them to view the accounts of customers and shops and their details, while also having admin controls over the accounts.	
Actors:	1- Admin.	
Related Use Cases:		
Stakeholders:	1- Customer. 2- Admin. 3- Shop.	
Preconditions:	At least one account must exist in the system.	
Post conditions:	Accounts and their details and admin actions displayed.	
Flow of Activities:	Actor	System
	1. Admin requests to view account.	1.1 System displays all available info related to existing account. 1.2 System provides access to administrative controls.
Exception Conditions:	1.1 No accounts exist in the system.	

TABLE 8 "DISPLAYING ACCOUNTS" USE CASE DESCRIPTION

Use Case Name:	Displaying Negotiations' History.	
Scenario:	Admin views negotiation's chat logs.	
Triggering Event:	The admin needs to view and moderate chat logs between the customers and the shops.	
Brief Description:	The admin gains access to a dashboard that allows them to view chat logs and admin controls.	
Actors:	1- Admin.	
Related Use Cases:	Reviewing Requests.	
Stakeholders:	1- Customer. 2- Admin. 3- Shop.	
Preconditions:	At least one negotiation must have occurred in the system.	
Post conditions:	Negotiation logs displayed.	
Flow of Activities:	Actor	System
	1. Admin requests to view chat logs.	1.1 System displays all available info related to existing negotiations. 1.2 System provides access to administrative controls.
Exception Conditions:	1.1 No negotiations exist in the system.	

TABLE 9 "DISPLAYING NEGOTIATIONS' HISTORY" USE CASE DESCRIPTION

Use Case Name:	Editing Account.	
Scenario:	The actor manages and edits their account.	
Triggering Event:	When the actor wants to manage its account.	
Brief Description:	The actor can manage its account to be able to change the account details (such as email, username, and password).	
Actors:	1- Customer. 2- Shop.	
Related Use Cases:	Creating Accounts.	
Stakeholders:	1- Admin. 2- Customers. 3- Shop.	
Preconditions:	1 -The actor must have a registered account. 2 -The old password should be confirmed in the process of update the account.	
Post conditions:	Account updated.	
Flow of Activities:	Actor	System
	1. Actor requests to the system to manage its account. 2. Actor changes details	1.1 System prompts the customer to confirm old password. 2.1 System registers new details over the old ones.
Exception Conditions:	1.1 Old password is wrong. 2.1 New details already exist in the system.	

TABLE 10 "EDITING ACCOUNT" USE CASE DESCRIPTION

Use Case Name:	Delete issue	
Scenario:	The customer can delete the issue after posting it.	
Triggering Event:	When the Customer wants to delete the issue.	
Brief Description:	The customer can delete the issue if he doesn't want to repair his device anymore.	
Actors:	Customer.	
Related Use Cases:	Create issue.	
Stakeholders:	1- Admin. 2- Customer.	
Preconditions:	The customer must have an issue.	
Post conditions:	The issue is deleted successfully.	
Flow of Activities:	Actor	System
	1. Customer requests to the system to delete his issue.	1.1 System check for the availability of the issue. 1.2 System delete the issue. 1.3 System refund the fee.
Exception Conditions:	1.1 No issues exist in the system.	

TABLE 11 "DELETE ISSUE" USE CASE DESCRIPTION

10.3 Activity Diagram:

The Activity Diagram are very important, because it shows the process of each case. The following Diagrams represent most important Activity Diagrams for each different actor:

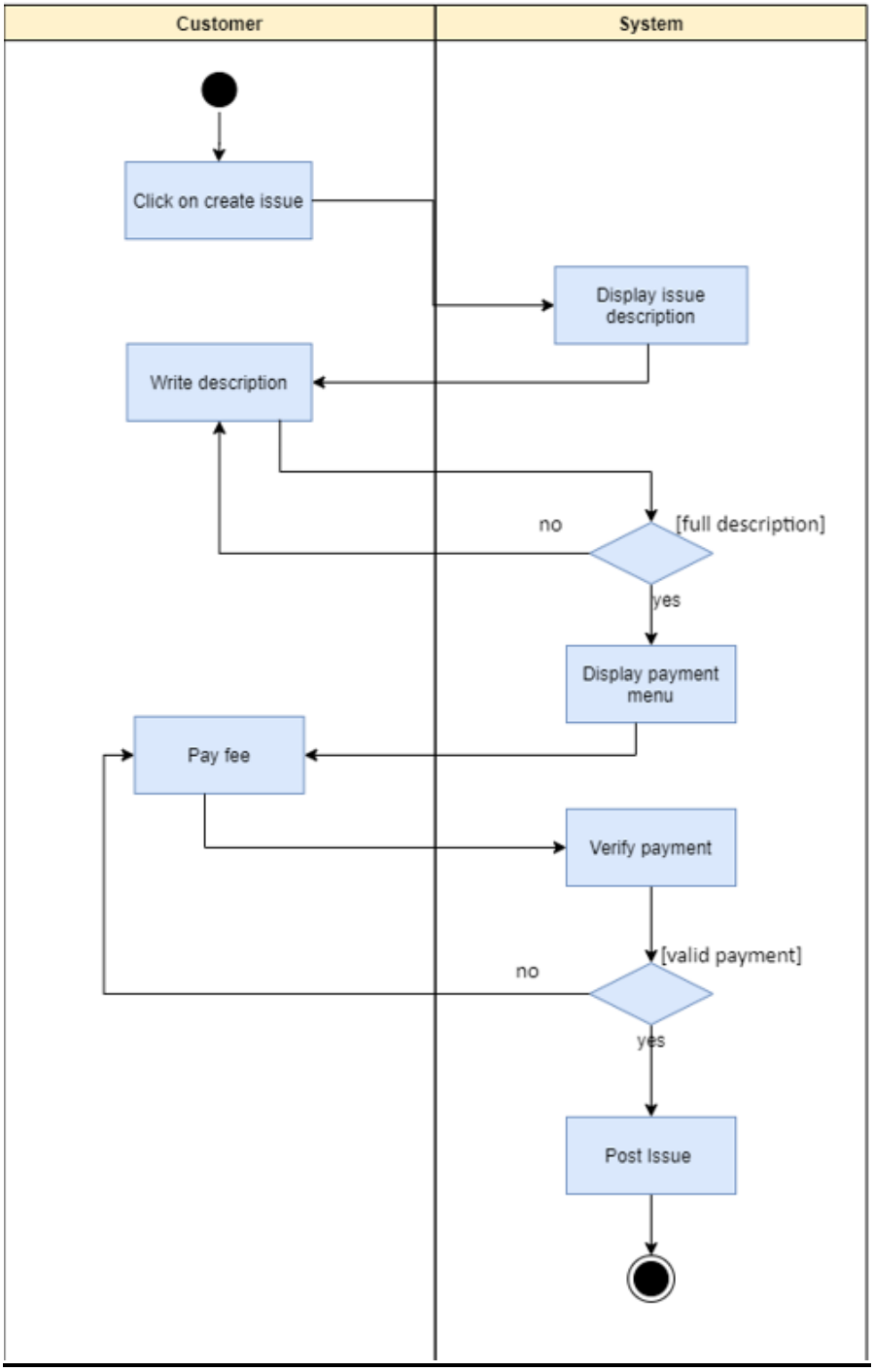


FIGURE 13 "CREATING ISSUES" ACTIVITY DIAGRAM

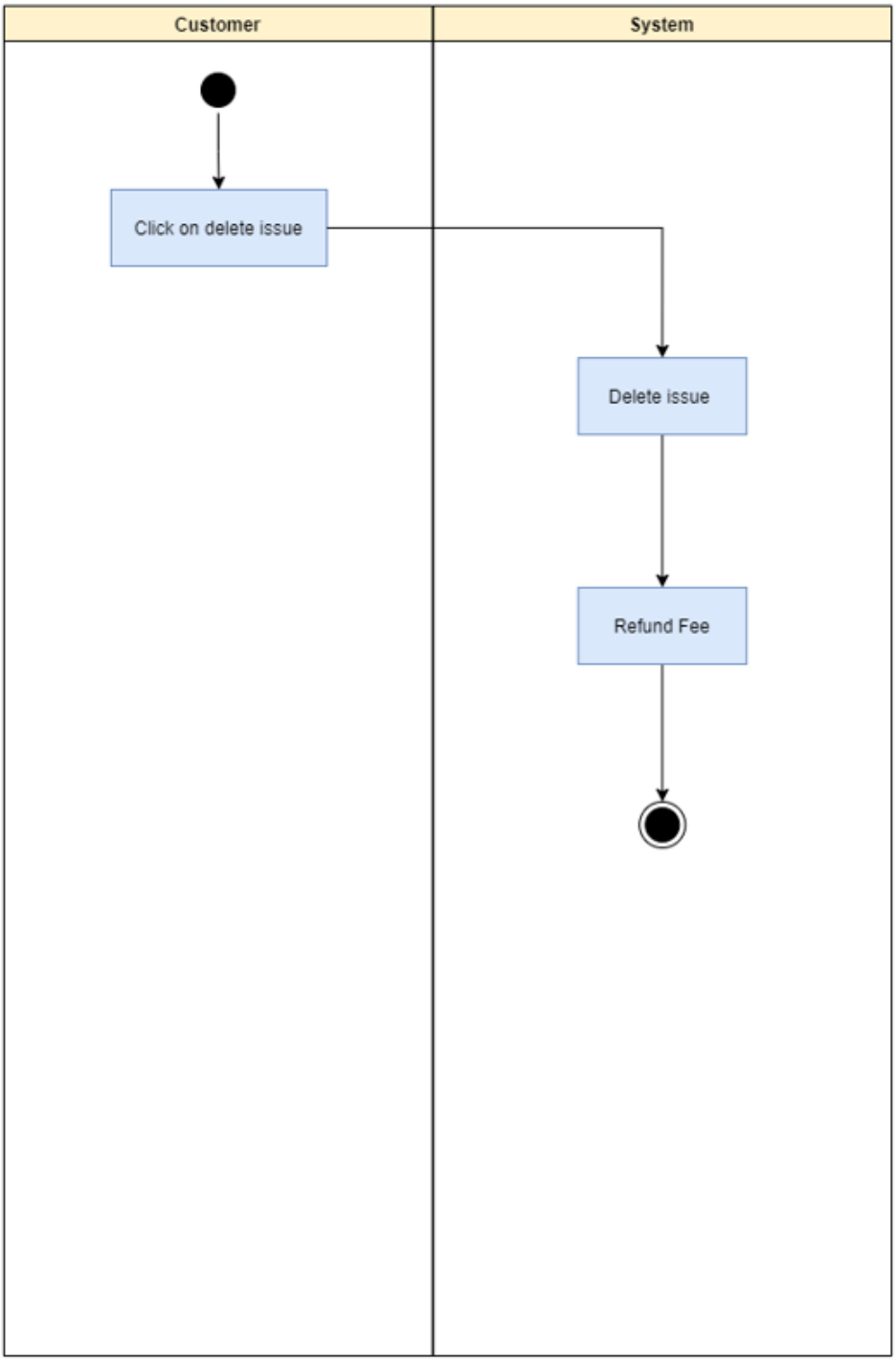


FIGURE 14 "DELETE ISSUE" ACTIVITY DIAGRAM

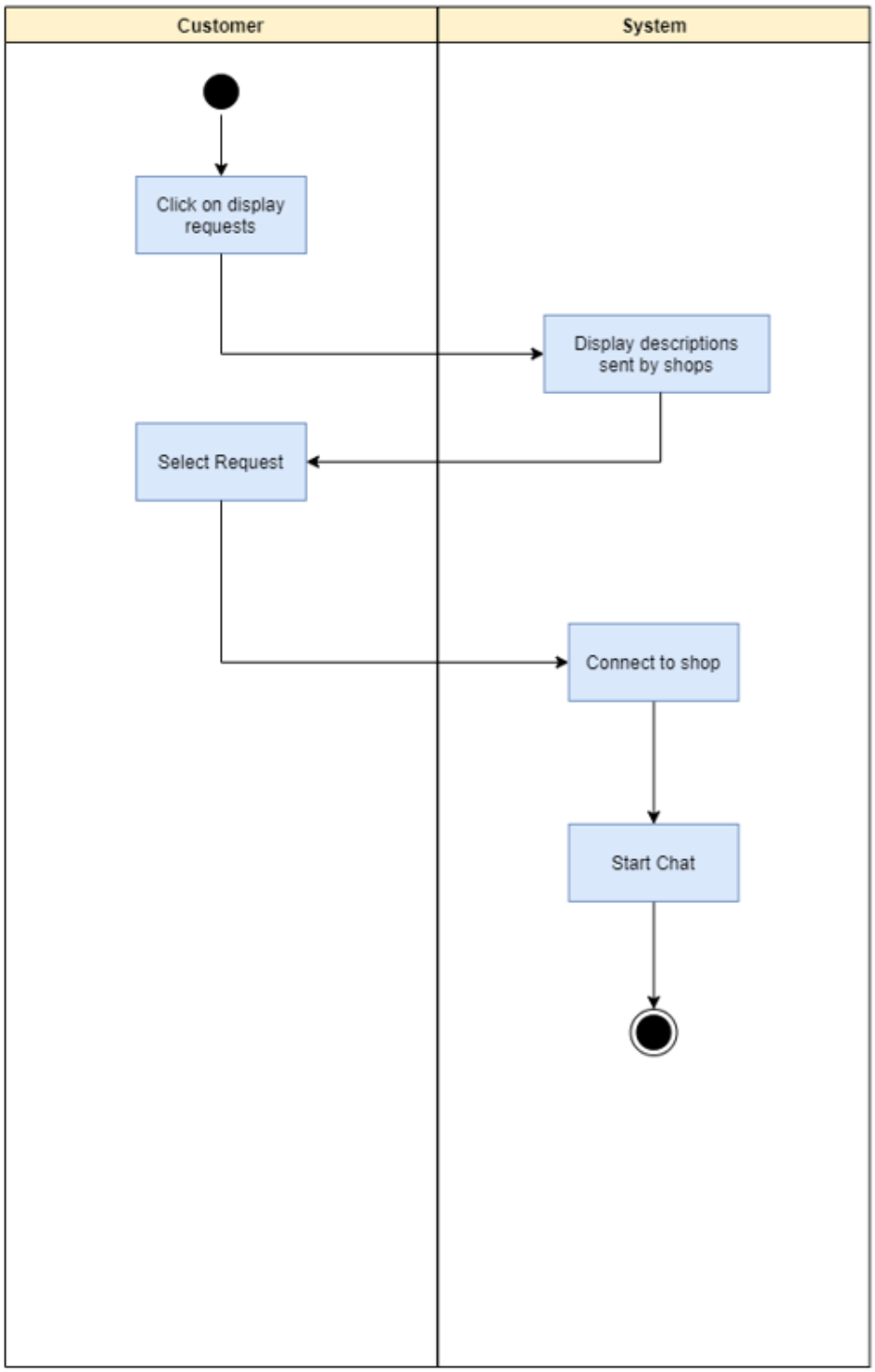


FIGURE 15 "REVIEW REQUESTS" ACTIVITY DIAGRAM

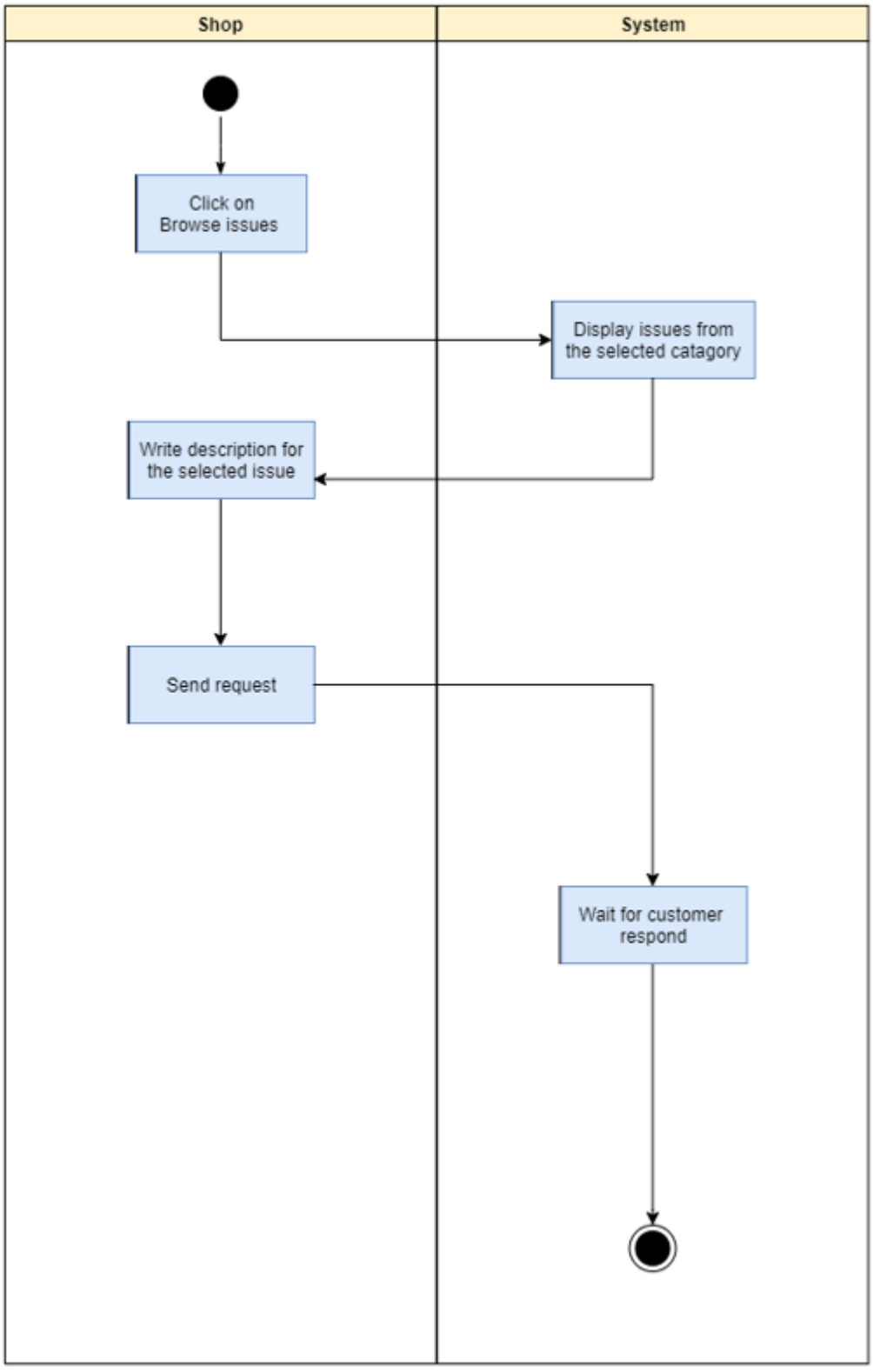


FIGURE 16 "BROWSE ISSUES" ACTIVITY DIAGRAM

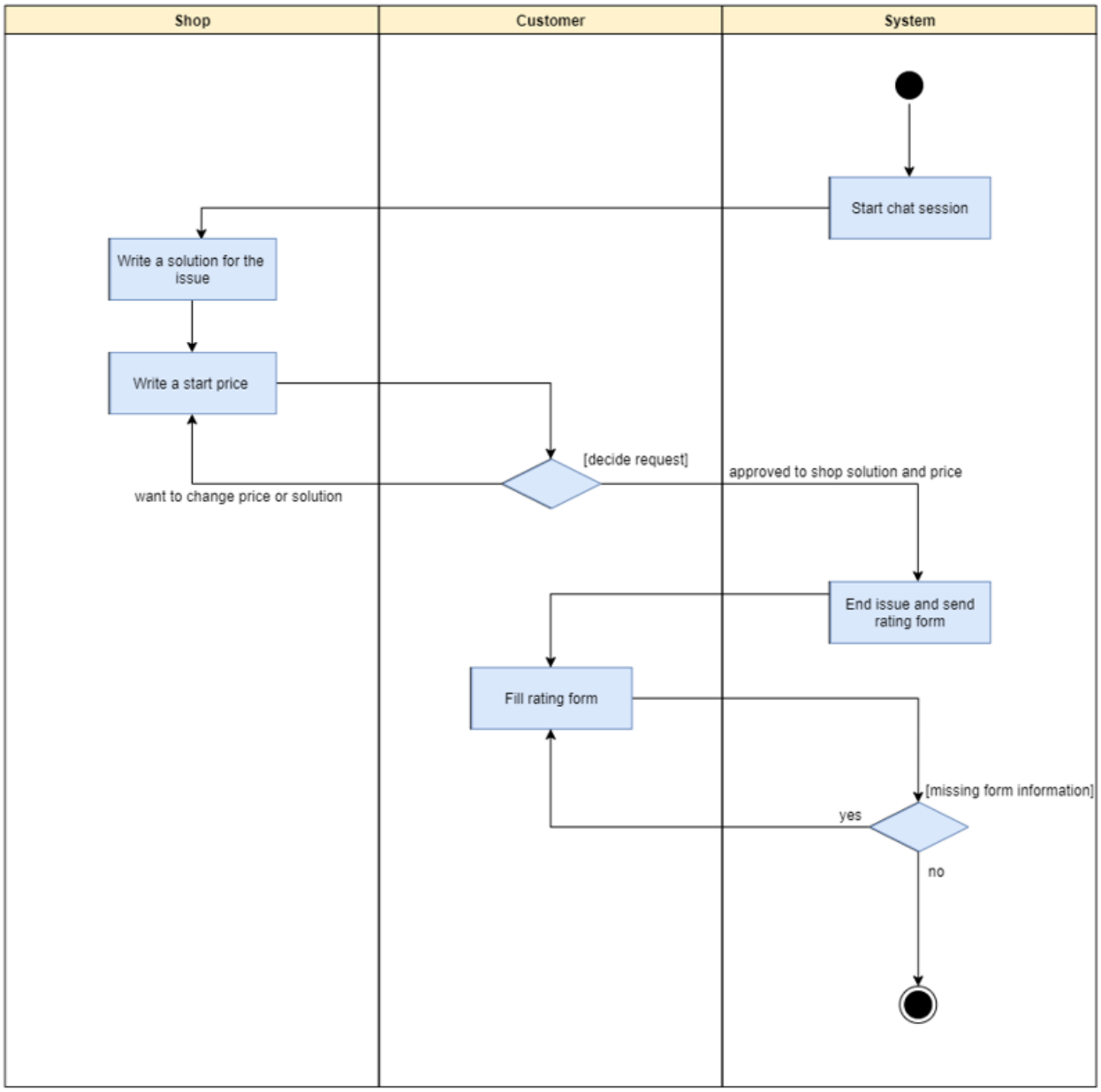


FIGURE 17 "START NEGOTIATION" ACTIVITY DIAGRAM

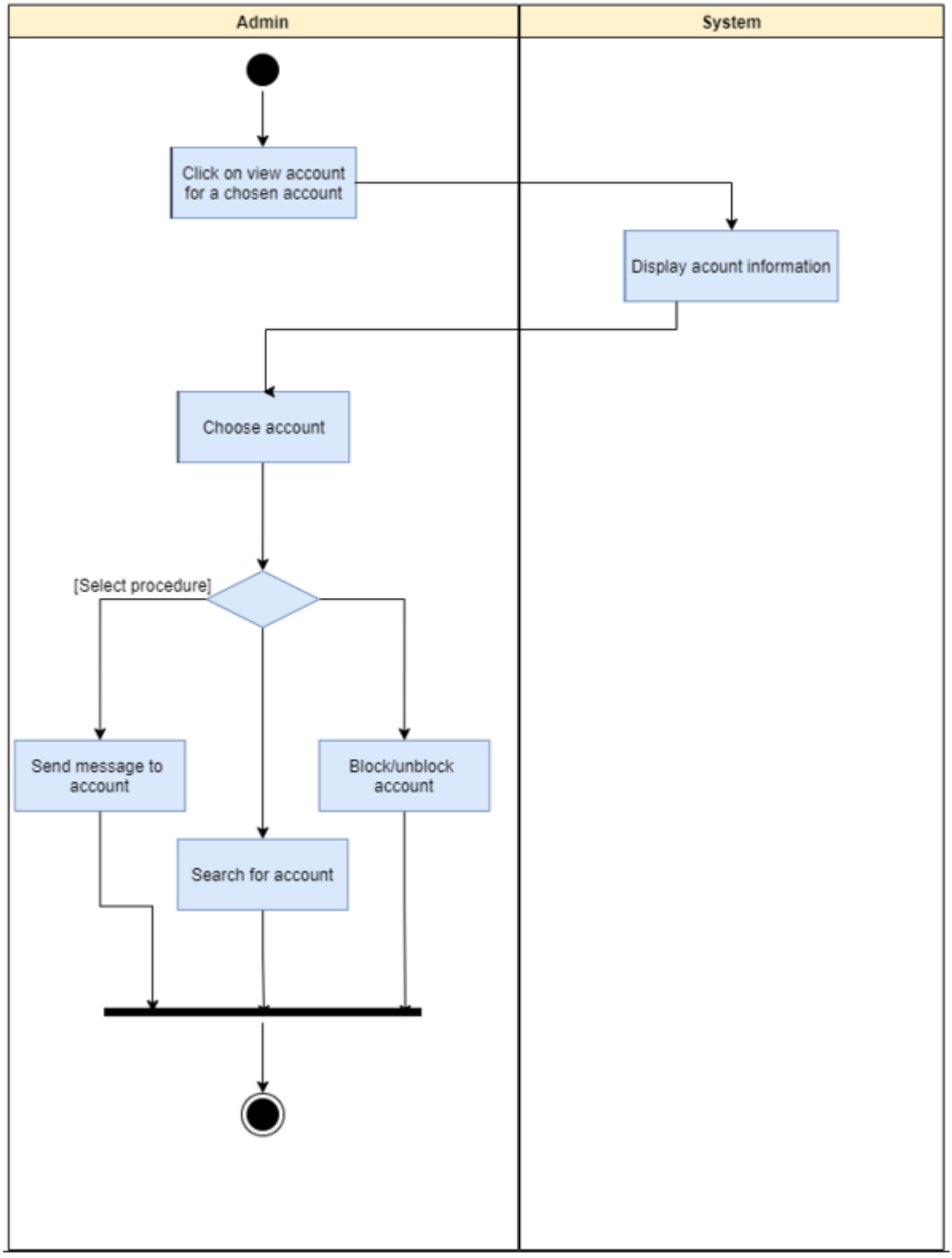


FIGURE 18 "VIEW ACCOUNT" ACTIVITY DIAGRAM

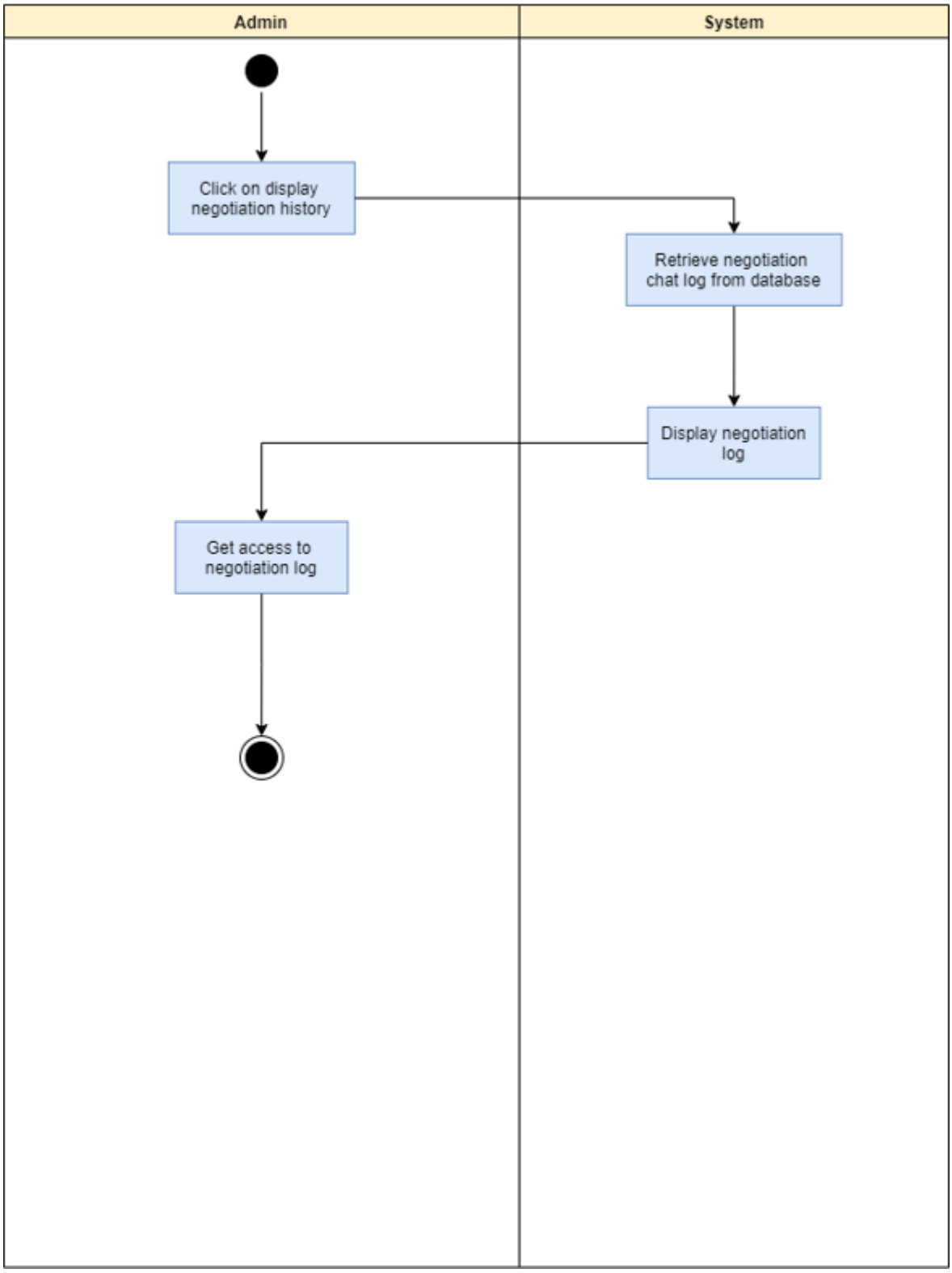


FIGURE 19 "VIEW CHATLOG" ACTIVITY DIAGRAM

10.4 Entity Relationship Diagram (ER Diagram):

The Entity Relationship model is used by software engineers to describe the database. With the ER Diagram we can describe each relation (table) in the database, and the relations are shown, if it is a mandatory or an optional relation.

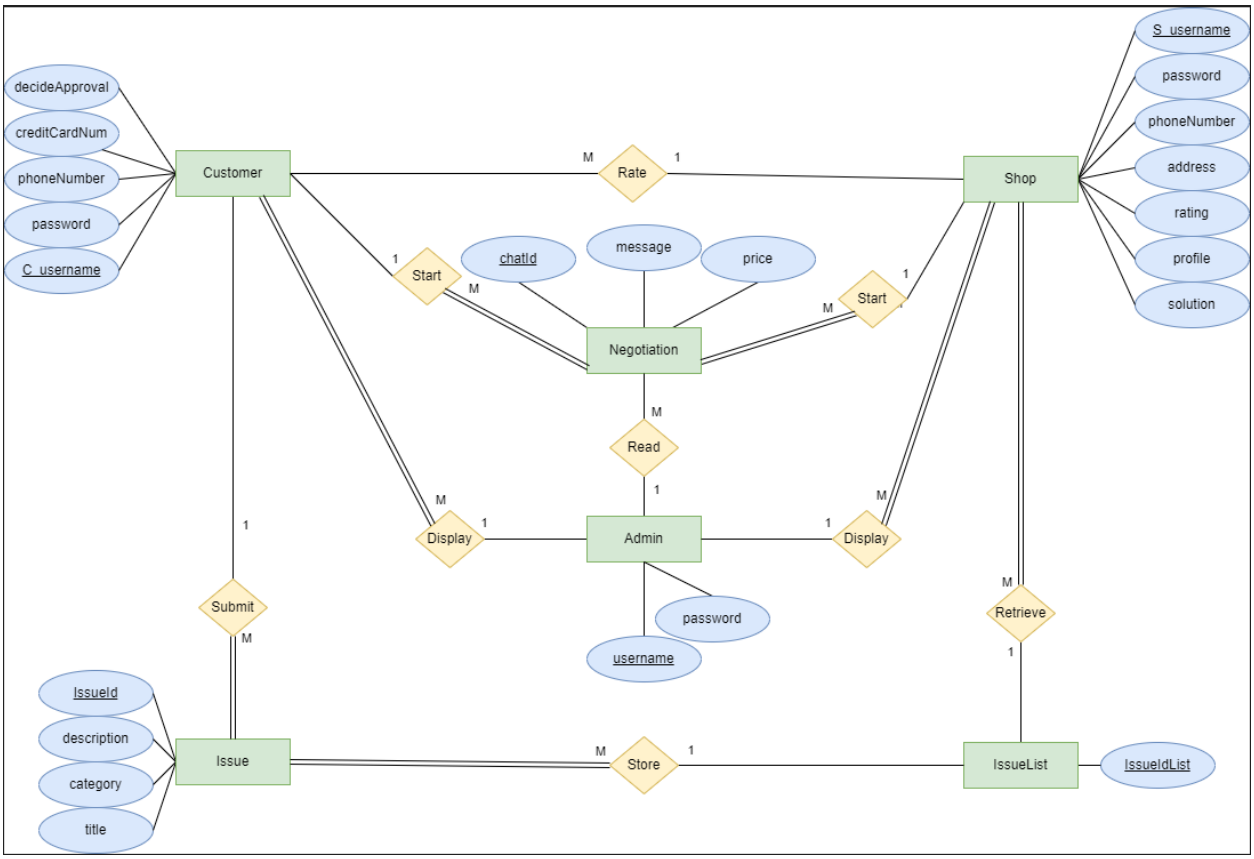


FIGURE 20 ER DIAGRAM

10.5 Rational Schema:

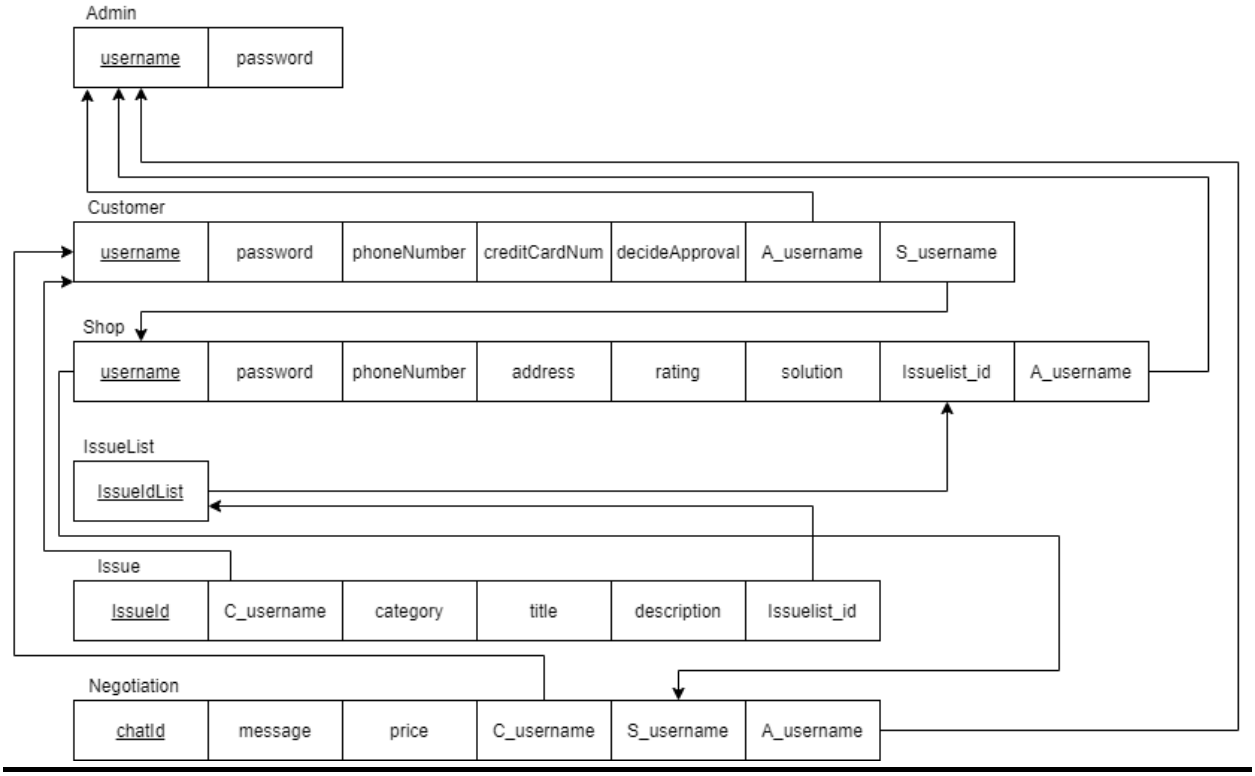


FIGURE 21 RATIONAL SCHEMA

10.6 Sequence Diagram:

A Sequence Diagram shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are sometimes called event diagrams or event scenarios.

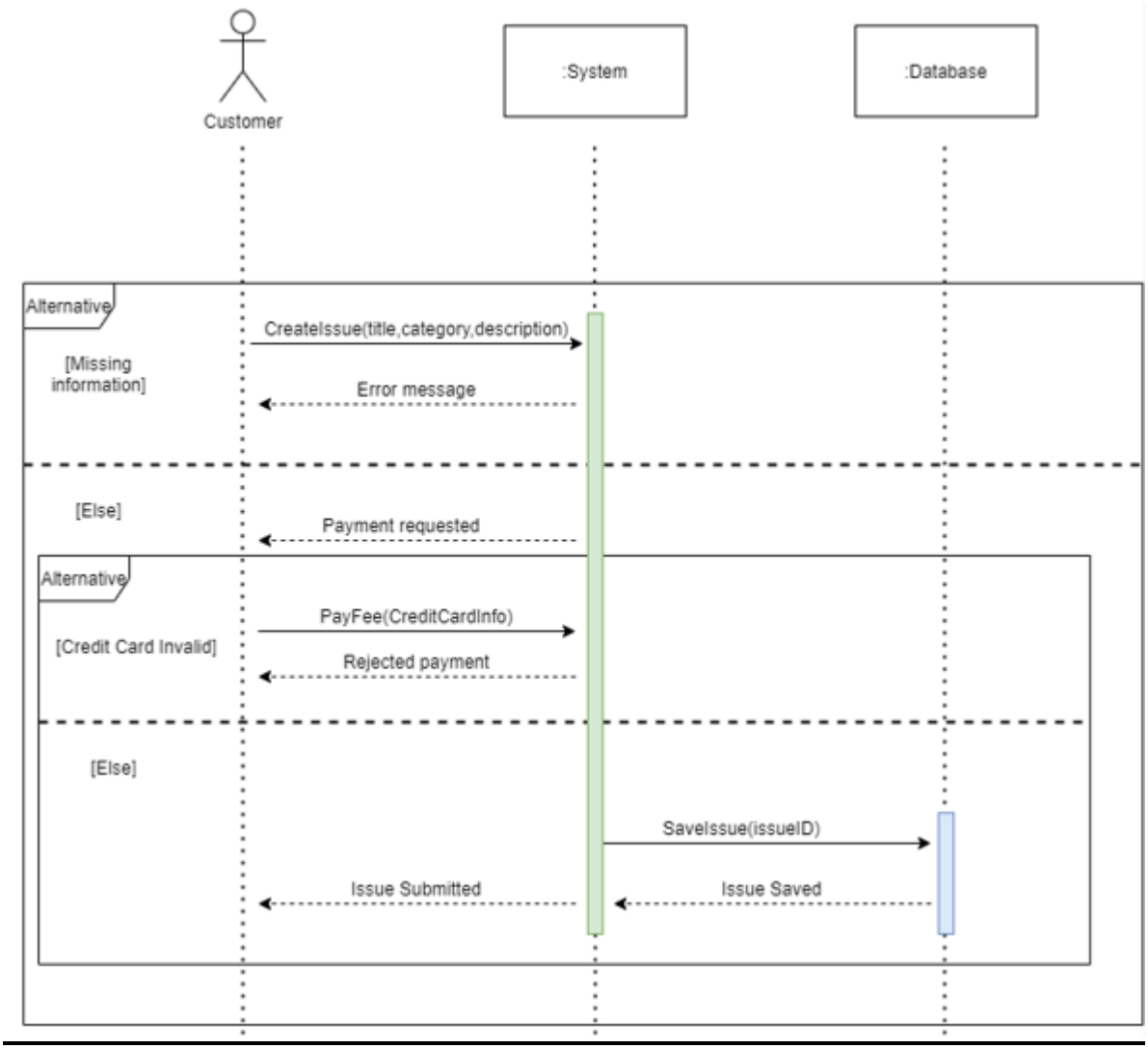


FIGURE 22 "CREATING ISSUE" SEQUENCE DIAGRAM

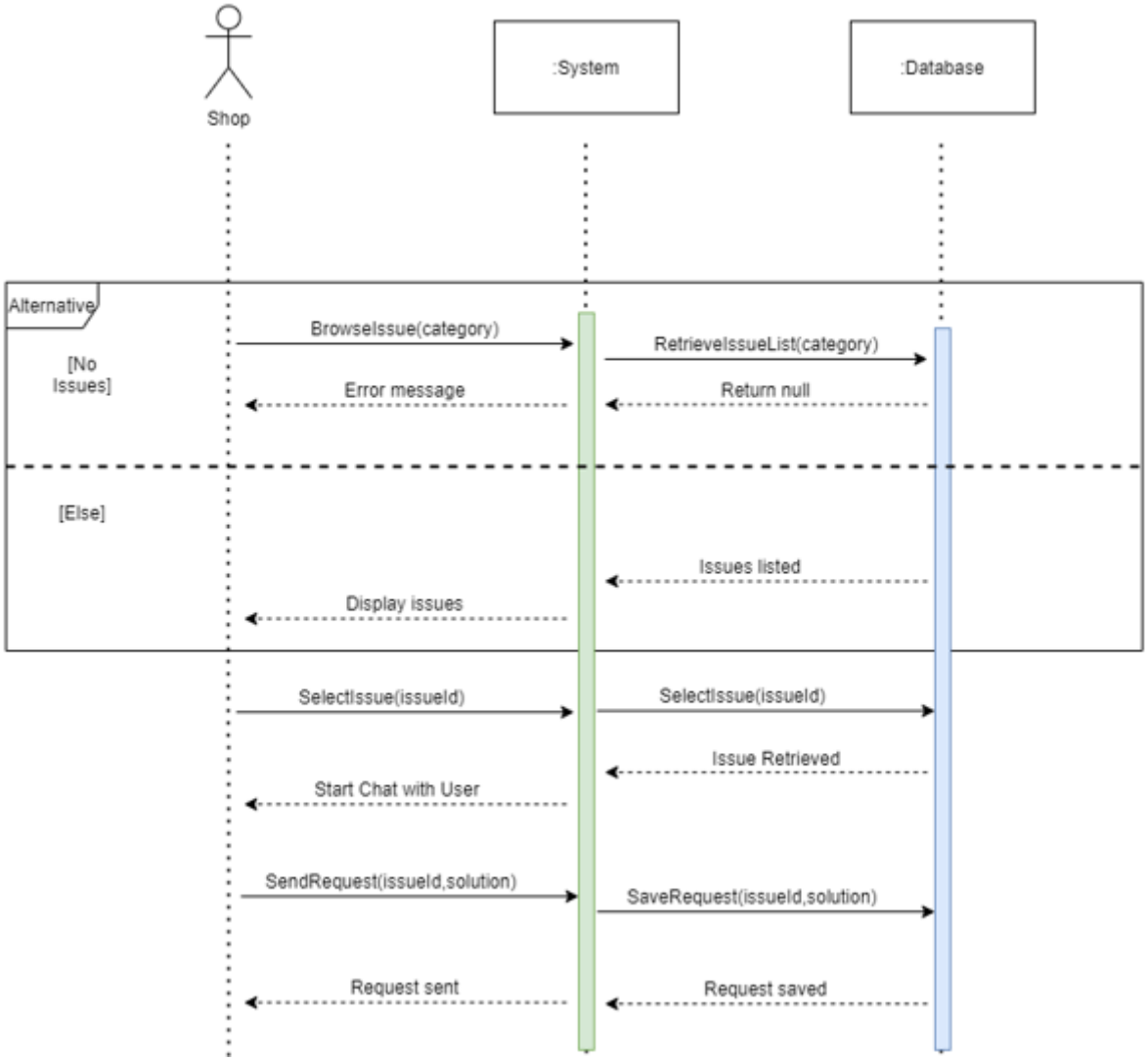


FIGURE 23 "BROWSE ISSUES" SEQUENCE DIAGRAM

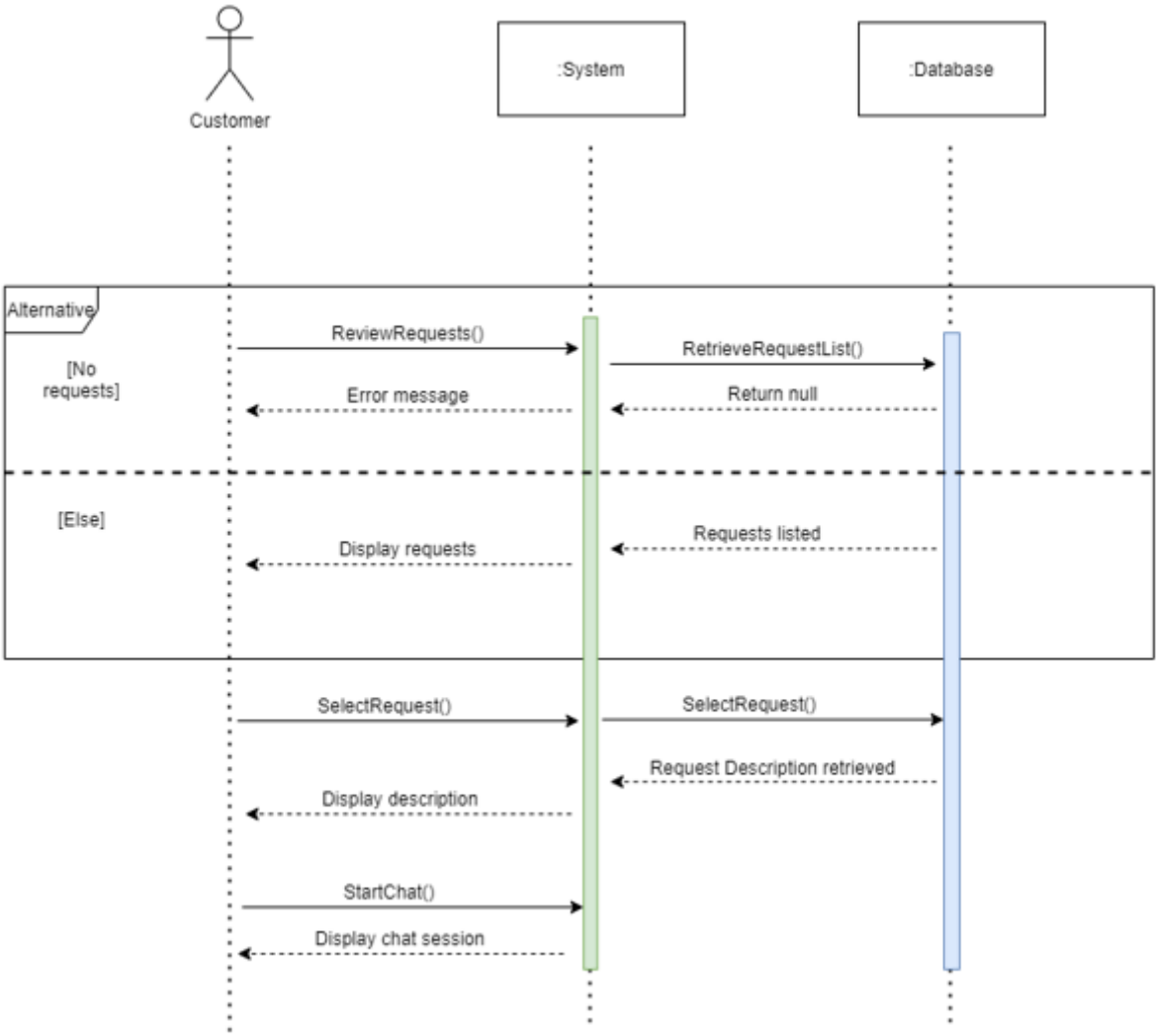


FIGURE 24 "REVIEW REQUESTS" SEQUENCE DIAGRAM

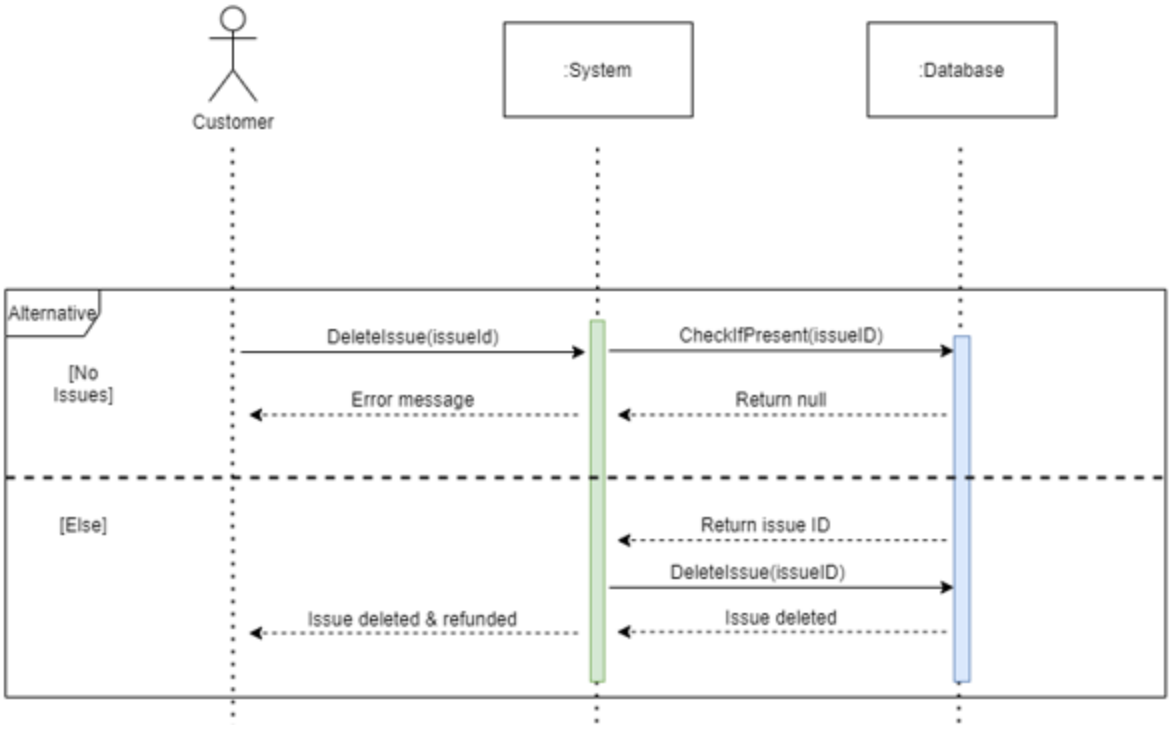


FIGURE 25 "DELETE ISSUE" SEQUENCE DIAGRAM

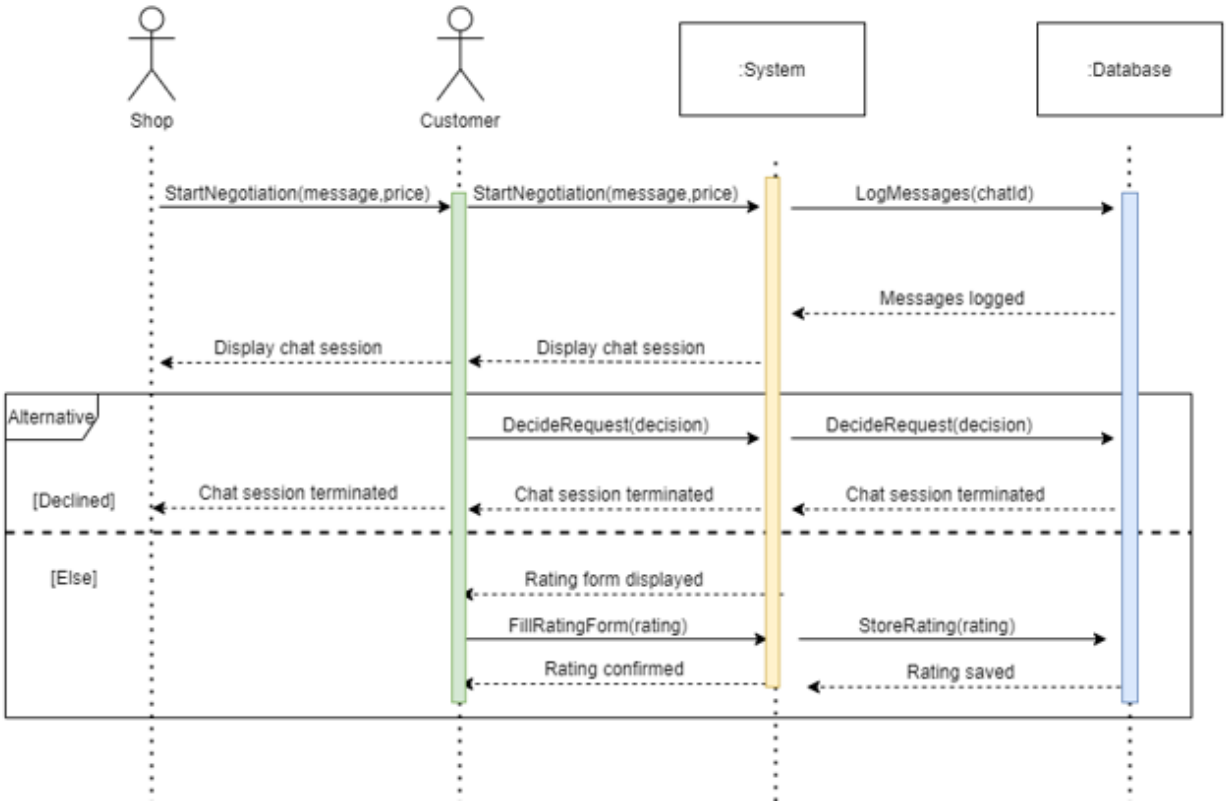


FIGURE 26 "START NEGOTIATION" SEQUENCE DIAGRAM

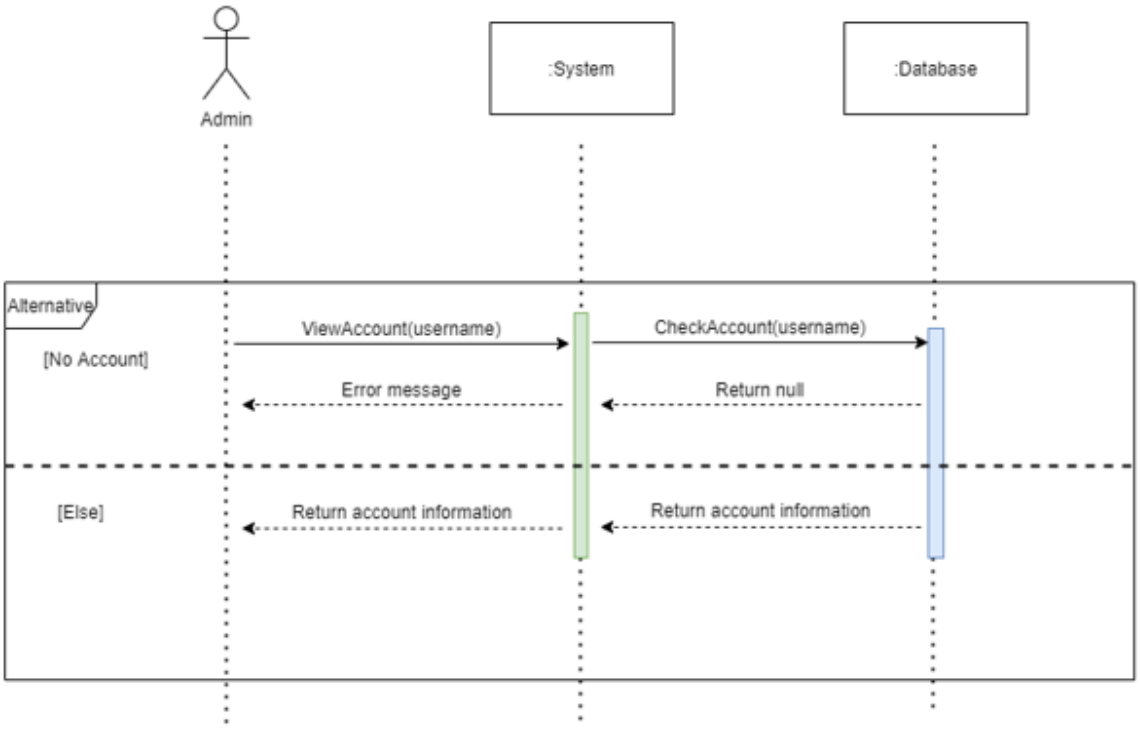


FIGURE 27 "VIEW ACCOUNT" SEQUENCE DIAGRAM

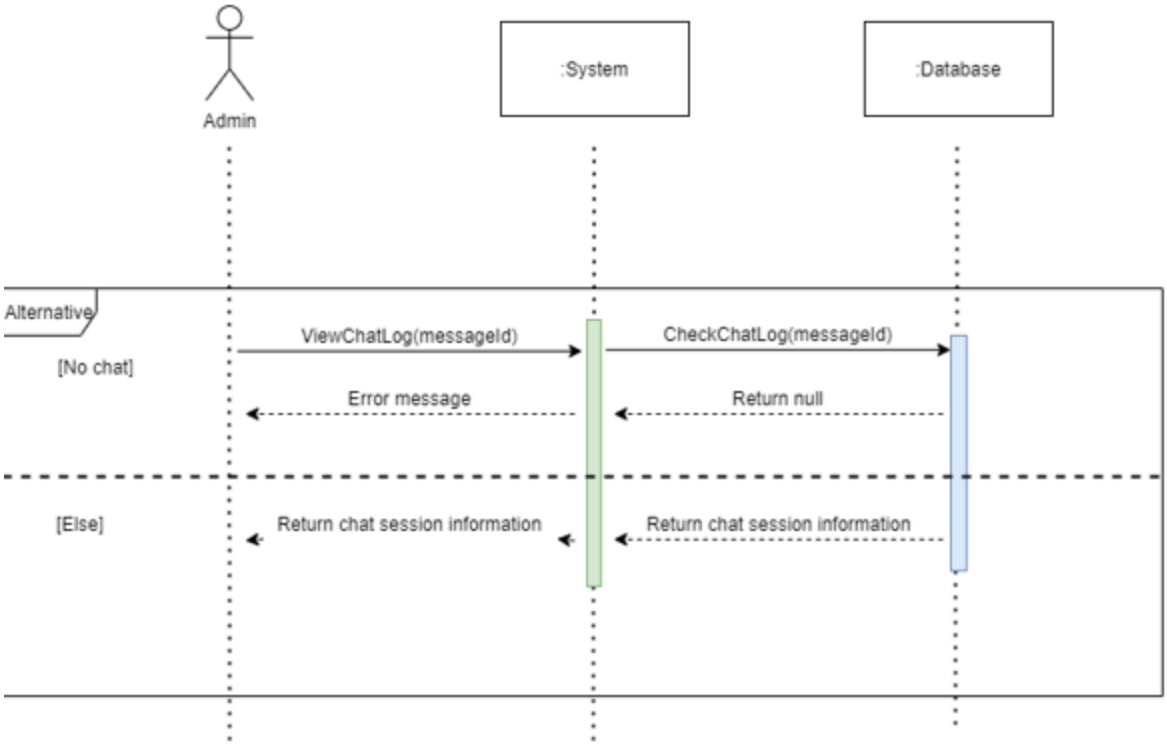


FIGURE 28 "VIEW CHAT LOG" SEQUENCE DIAGRAM

10.7 Class Diagram:

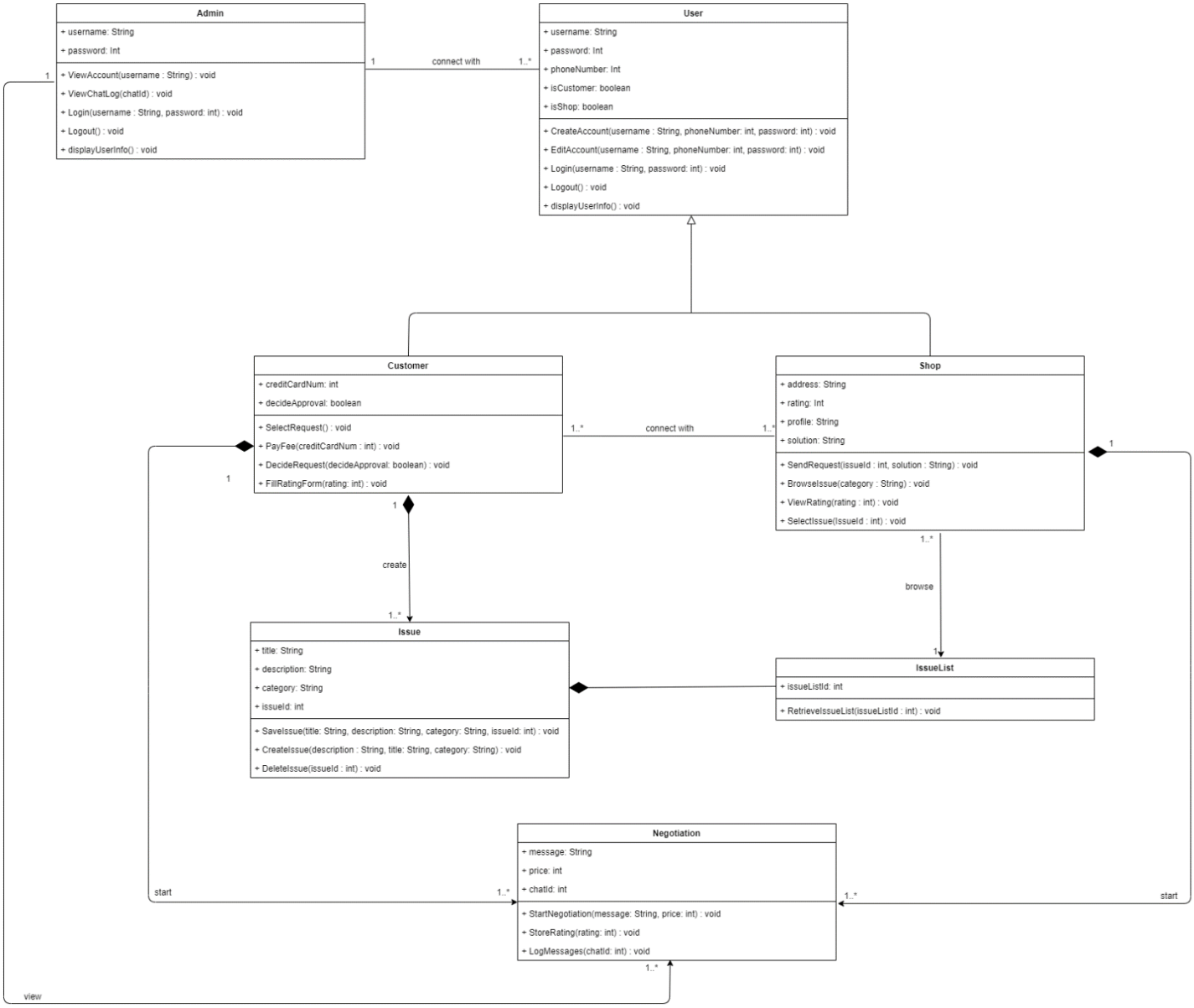


FIGURE 29 CLASS DIAGRAM

11. Conclusion:

- ## Social Impact

This web application will help both shops who are looking for customers and customers who are looking to fix their phones save both their efforts and time by allowing easy contact and arrangement between them.

- ## Ethical Impact

The web application will help customers gain more confidence in whatever shop they pick to fix their phone, since we allow viewing of multiple shops and their ratings.

- ## Legal Impact

The admins have the ability to review chat logs, in case there was a legal concern, thus aiding the legal process.

12. Reference List:

- Video Conferencing, Cloud Phone, Webinars, Chat, Virtual Events | Zoom
- Project Management Software | Microsoft Project
- https://www.draw.io/