

Iteration 2:

Table of Contents:

ADD Step 1: Review Inputs

- 1. Step 2: Establish Iteration Goal by Selecting Drivers**
- 2. Step 3: Choose One or More Elements of the System to Refine**
- 3. Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers**
- 4. Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces**
- 5. Step 6: Sketch views and Record Design Decisions**
- 6. Step 7: Perform Analysis of Current Design and Review Iteration Goal**

Design Process

The following section follows the ADD process as well as looking back onto iteration 2. We will look at the inputs to allow us to be able to choose the drivers in the design decisions that we make in this process. We will then begin the second iteration for steps 2 to 7.

ADD Step 1: Review Inputs

In this step of the ADD method, we will review the inputs and show which requirements will be the drivers, as shown below.

Category	Details		
Design Purpose	The purpose is to produce a detailed design to allow the construction of the application in the most efficient manner		
Primary Functional requirements	The primary use cases we will be using will be the following: UC-1: Monitoring the Network Status UC-2: Detecting Faults in the Network UC-8: Displaying information UC-10: User Authentication		
Quality Attribute Scenarios			
	ID	Importance	Implementation Difficulty
	QA-1	High	High
	QA-3	High	Medium
	QA-4	High	Low
	QA-5	Medium	High
	QA-6	Medium	High

Step 2: Establish Iteration Goal by Selecting Drivers

The goal of this iteration is to address the concern of identifying structures and to support the primary functions:

- UC-1: Monitor App Status
- UC-2: User Authentication
- UC-13: Register

Step 3: Choose One or More Elements of the System to Refine

The elements of the system that will be refined are display order history, we will be refining this element to display order history from a longer period of time.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

Design Decisions and Locations	Rationale and Assumptions
Create a Domain Model for the application	We must identify the major entities in the domain along with their relationships. In order to do this we must create a domain model for the system.
Identify Domain Objects that map to functional requirements	We do this to each distinct functional element so we can understand the modules in different layers rather than have them all be coagulated.
Map the system use cases to domain objects	We use this to identify major entities so it can be analyzed and deciphered with whatever use cases the system has.
Use Spring framework and Hibernate	Spring because of how widely it is used, we can use it in our SmartShop for the application development. Hibernate also works very well with Spring. The development team also agrees that Spring is a good framework to work with.

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decisions and Locations	Rationale
Create only an initial domain model	The primary use cases need to be outlined and defined but only in an initial domain model.
Map the system use case	We need to identify the objects that are all in our system use cases.
Associate frameworks with a module in the data layer	As spoken about earlier, we will be using the Spring and Hibernate framework hand in hand and in doing so we will be able to discuss previous modules.
Decompose the domain objects across the layers to identify layer-specific modules with an explicit interface	This will ensure that all the modules are being supported and all the functionalities will be identified. Making sure that it follows CRN-3, where all the concerns are that the server load will be divided of busy to non busy across different time zones

Step 6: Sketch views and Record Design Decisions

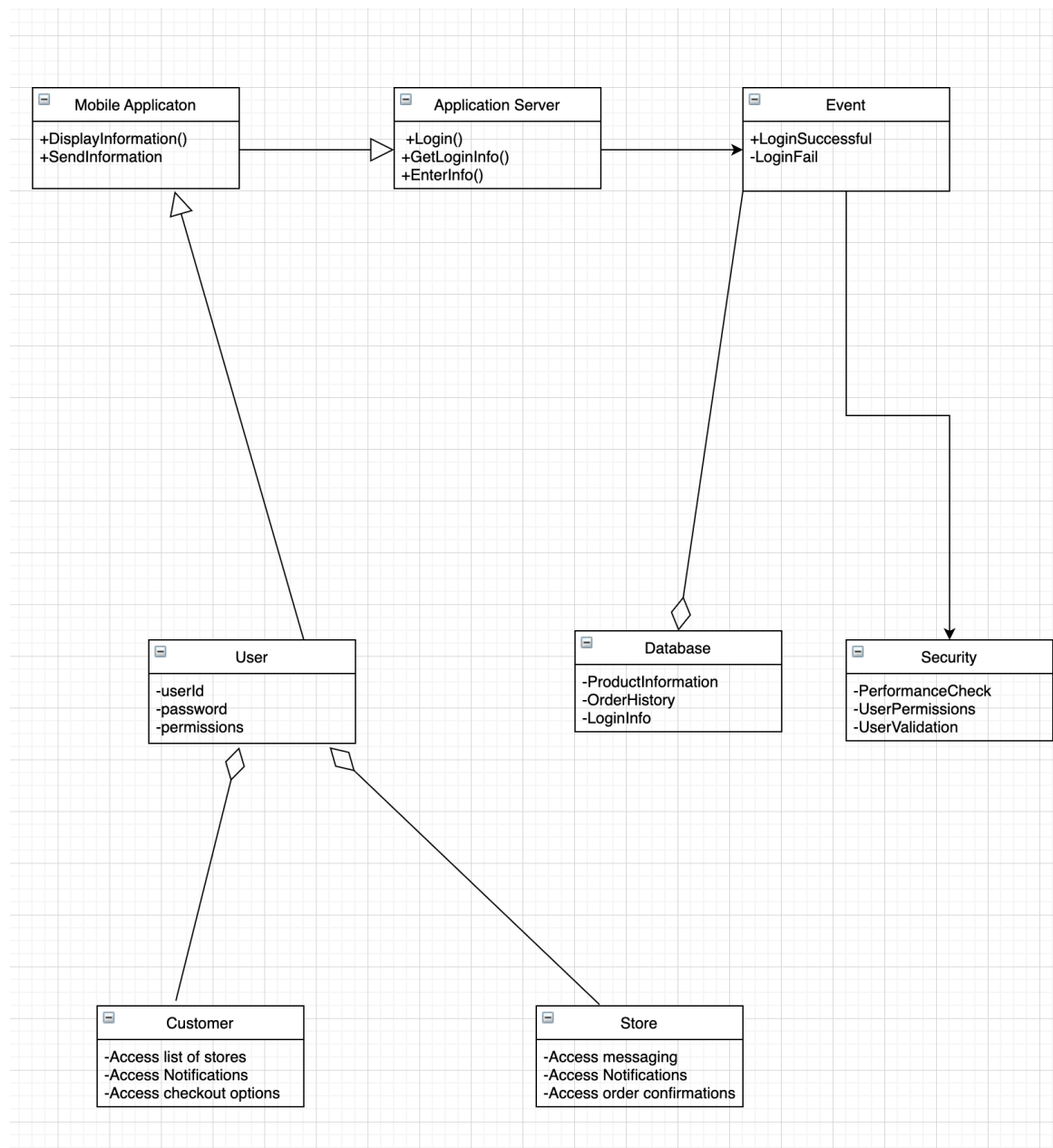


Figure 1: Initial Domain Model

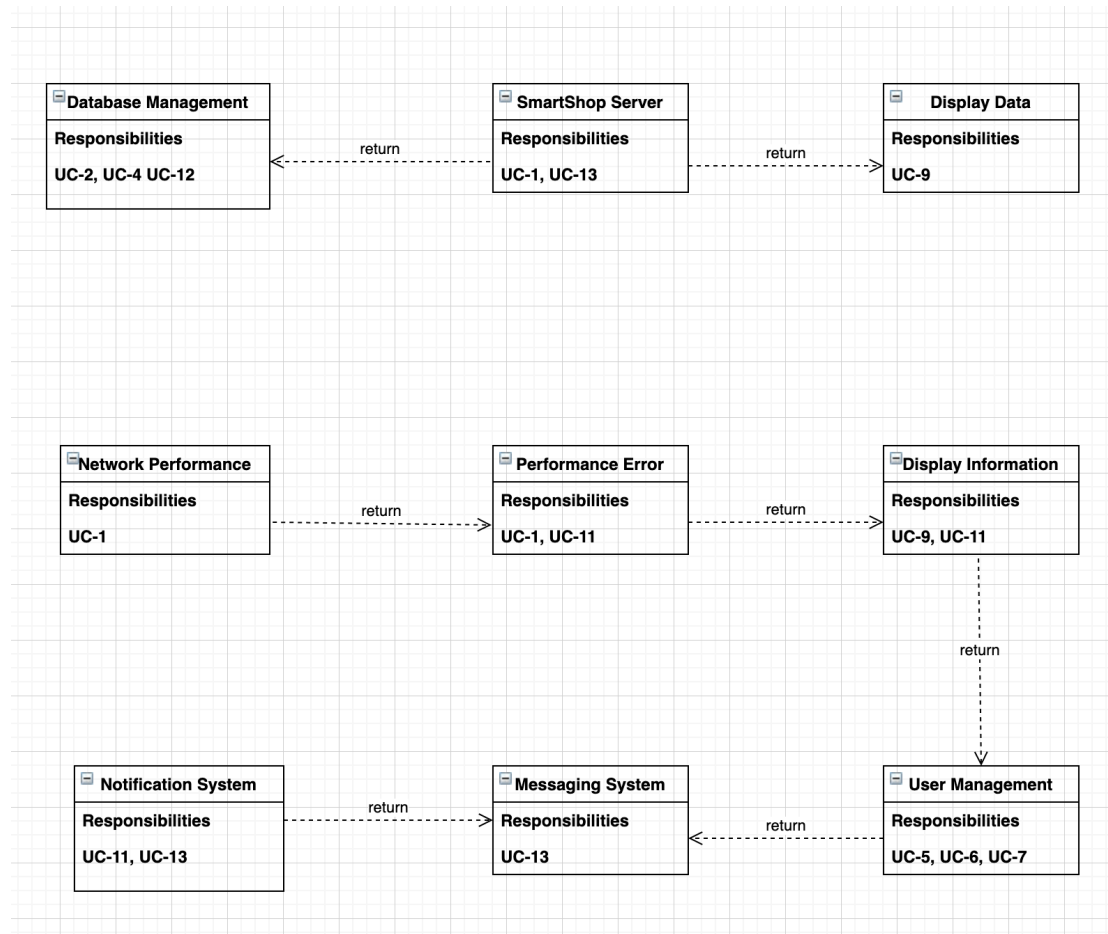


Figure 2: Domain Objects Associated with Use Cases

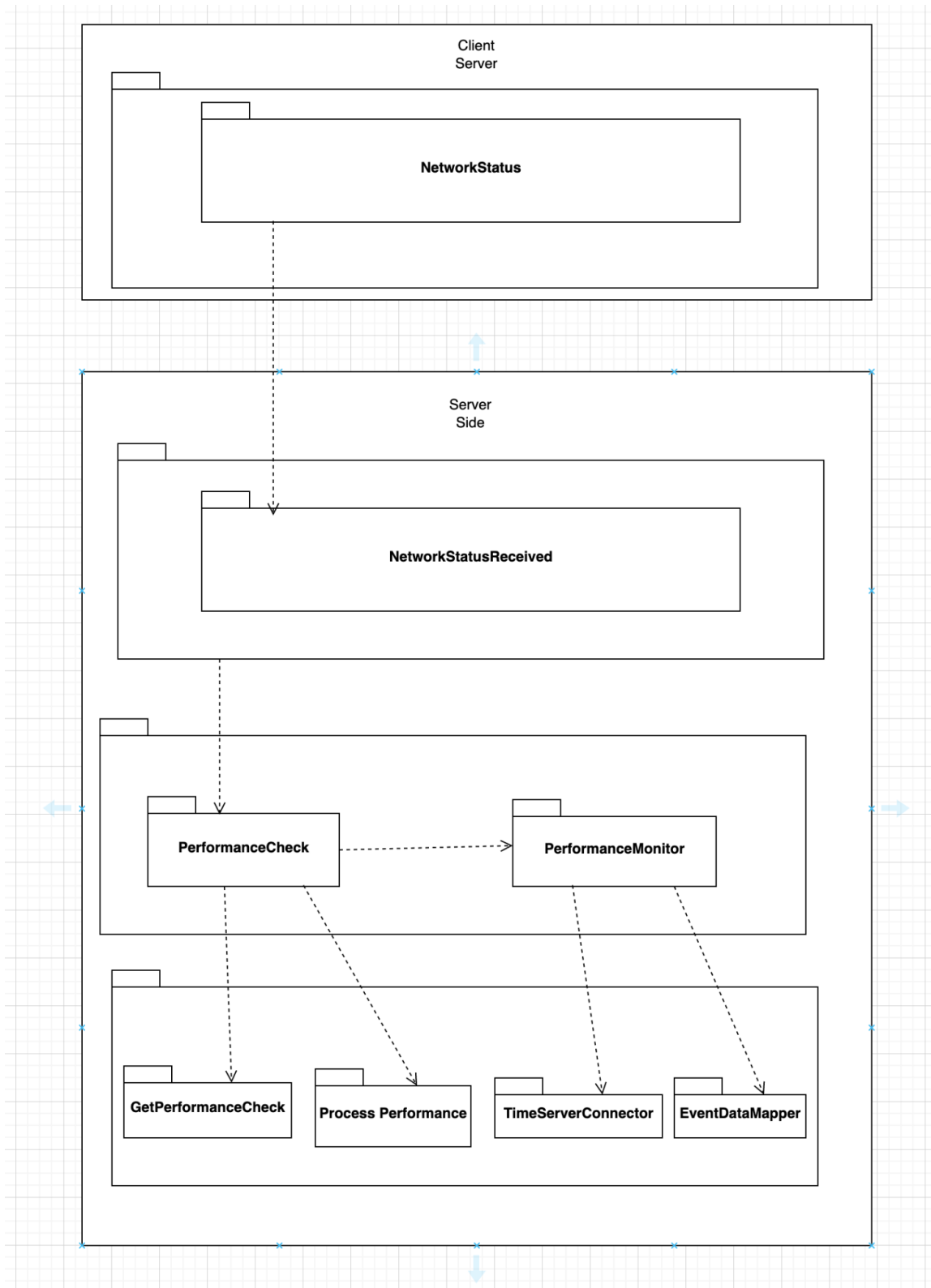


Figure 3: Extended Module View

The responsibilities of the the above figure are in the table below:

Element	Responsibility
Network Status	Gets network status from users via the mobile application.
Network Status Received	Receives the requested data from the user.
Performance Check	Checks performance from the user.
Performance Monitor	Monitors performance across all users.
Time Server Connector	Responsible for persistence operations related to the time server.
Event Data Mapper	Responsible for persistence operations related to the events.

USE CASE 1: Monitor App Status

The following figure shows us the initial sequence diagram for UC-1. It shows how the user will interact and access the app and the app will work in turn its performance and efficiency to the users needs. When launched, a loading screen will occur, and then connect the user to the application which holds the database for all the stores that the user will browse through.

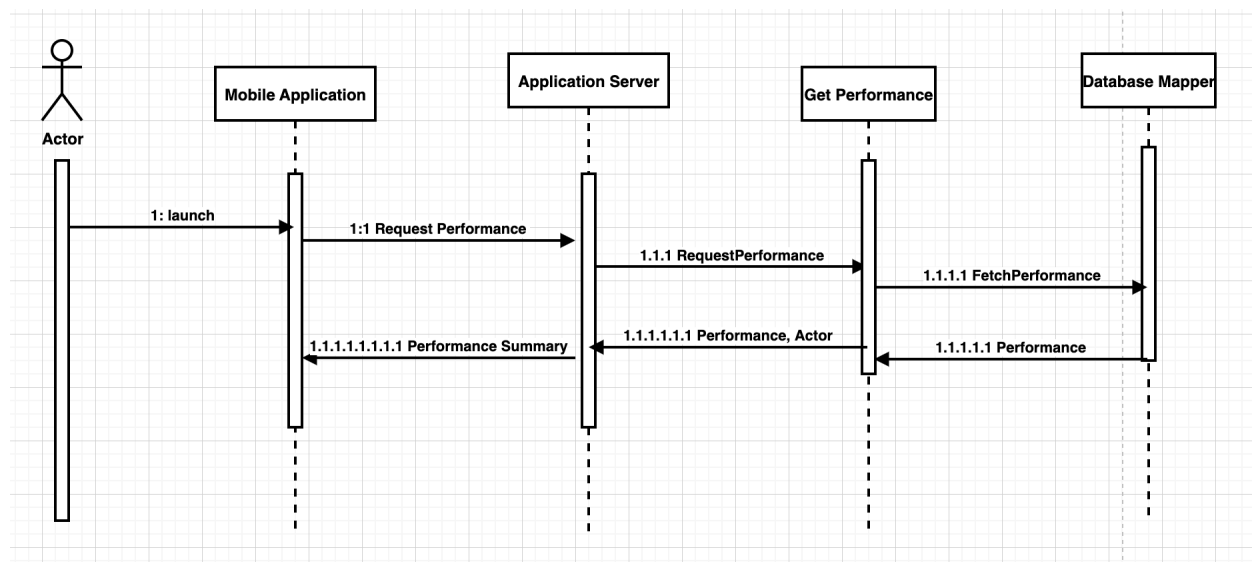


Figure 4: Sequence Diagram for UC-1

Element	Responsibility
Mobile Application	Application that is being accessed by users.
Request performance	Send a server request while the user has launched the application.
Request server performance	Send a request to the network regarding the server.
Fetch performance	Receive the fulfilled performance requests.
Performance Summary	Get a summary of all performance inquiries.

USE CASE 2: User Authentication

The next figure shows the initial sequence for UC-2. It shows the user authentication and how the user would access the application via logging in. This example shows the application being accessed by a mobile device which connects to the server which holds the database with their personal information. The data is then sent back to the actor.

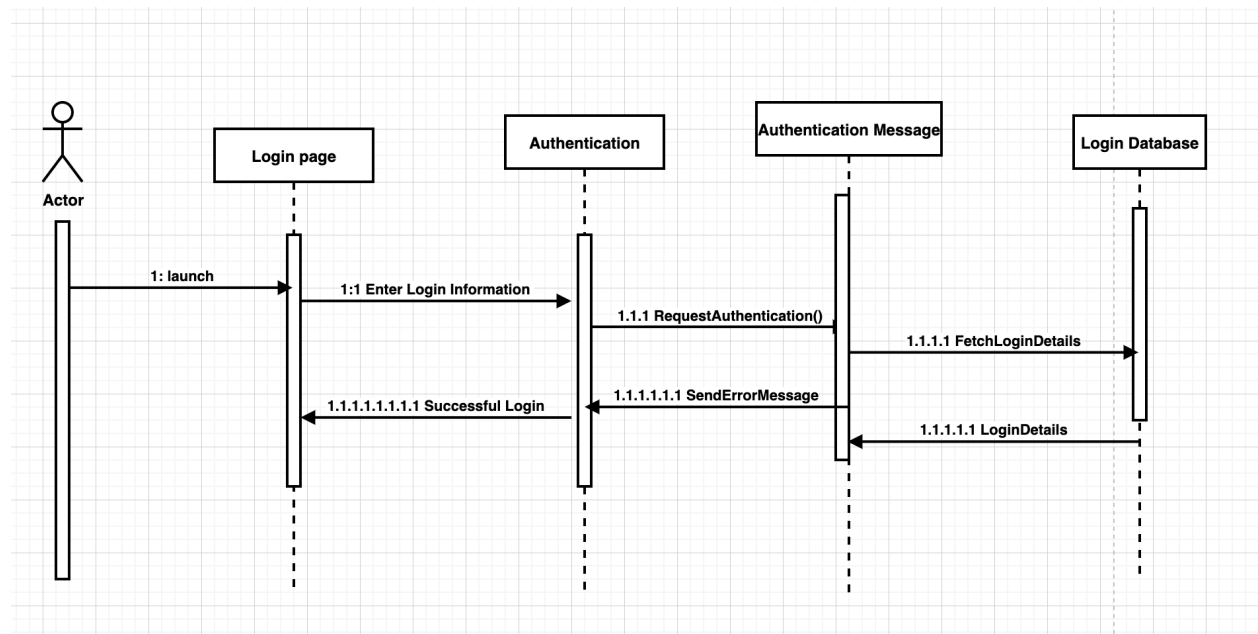


Figure 5: Sequence Diagram for UC-2

Element	Responsibility
Login page	Users can send login info by putting in credentials.
Authentication	Users will put personal credentials and the system will check for validity of credentials.
Authentication message	If user can log in, will return successful login, and if can't, will return error.
Login Database	Will have login details stored if user needs it changed.

USE CASE 13: Register

The next figure shows the initial sequence for UC-13. It shows the register processing the payment and how the users final decision will take place after ordering. This example shows the application being accessed in the application, and then processing the order. The data is then finally sent back to the actor.

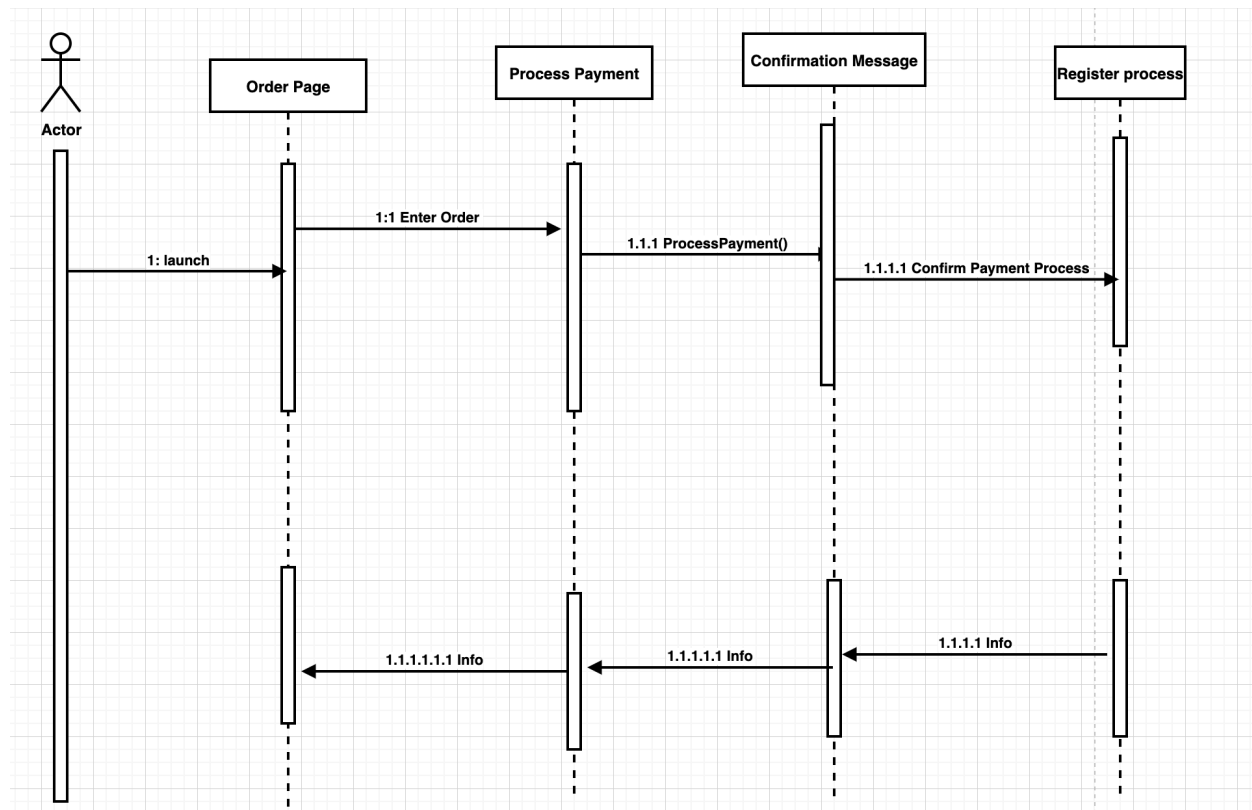


Figure 6: Sequence Diagram for UC-13

Element	Responsibility
Order Page	User will enter order based on items in database.
Process Payment	Will have payment processed through the server and application and will request network validation while logged into the application.
Confirmation Process	Request recent order info as well as displays most recent order.
Register Process	Request order history, and order tracking and order message.

Step 7: Perform Analysis of Current Design and Review Iteration Goal

The designs made in this iteration had guided us towards an initial understanding of the efficiency and functionality and how it is used through the system. The primary use cases that were identified and the modules alongside them had proved to be very beneficial. As we had made our sketches and understood the design process, we had a new architectural concern, and will be added to the Kanban board as shown below.

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions made during the iteration
		UC-1	Diagrams have been created to support this use case in this iteration.
		UC-2	Modules across layers and preliminary interfaces to support this use case have been identified.
	UC-4		Use cases have been briefly spoken about, but not the primary focus in this iteration.
		UC-13	Modules across layers and preliminary interfaces to support this use case have been identified.

		CRN-1	Additional technology is identified compatibility.
		CRN-2	Modules associated with all the use cases are identified to make them accessible to users.
	CRN-3		Additional knowledge regarding this is discussed with the team.
		QA-1	Relates to associated use case (UC-1) which has been described in this iteration. Performance will be monitored.
	QA-2		Relates to associated use cases (UC-1, UC-13) which have been described in this iteration. Customers will have access to a database of their saved information and purchase history.
	QA-3		No relevant design decisions were made in this iteration.
		QA-4	Relates to associated use case (UC-2) which has been described in this iteration. Will ensure the user has permissions to access their login information.
QA-5			Not addressed in this iteration
QA-6			Not addressed in this iteration
QA-7			Not addressed in this iteration
	CON-1		Identified modules relating to the issue have been investigated.
	CON-2		Modules for accessing security information and user information were identified in this iteration via UC-2,UC-1
CON-4			No relevant design decisions have been made so far in this

			iteration.
--	--	--	------------