|  |  |  |
| --- | --- | --- |
| VILNIAUS KOLEGIJA  UNIVERSITY OF APPLIED SCIENCES  FACULTY OF ELECTRONICS AND INFORMATICS  https://screenshotscdn.firefoxusercontent.com/images/eaf3f7f3-2952-4801-af5c-4f20e8ae8b88.png | | |
|  | | |
| **ECOMMERCE WEBSITE PROJECT** | | |
|  | | |
| GROUP WORK  6531BX028 PI18E | | |
| STUDENTS | (SIGNATURE) | EDITA KOMAROVA    DŽIUGAS PEČIULEVIČIUS |
| 2020-11 |
| (SIGNATURE) |
| 2020-11 |
|  |
|  |
|  |
| LECTURER | (SIGNATURE) | JŪRATE VAIČIULYTĖ | |
|  | 2020-11 |  |
|  |  |  |
| 2020 | | |

Table of Contents

[1 SOFTWARE ARCHITECTURE DIAGRAM 3](#_Toc57376500)

[2 SOFTWARE DESIGN MODEL 7](#_Toc57376501)

[2.1 Functional requirements traceability table 7](#_Toc57376502)

[2.2 Entity-relationship diagram 8](#_Toc57376503)

[2.3 Structure diagram 8](#_Toc57376504)

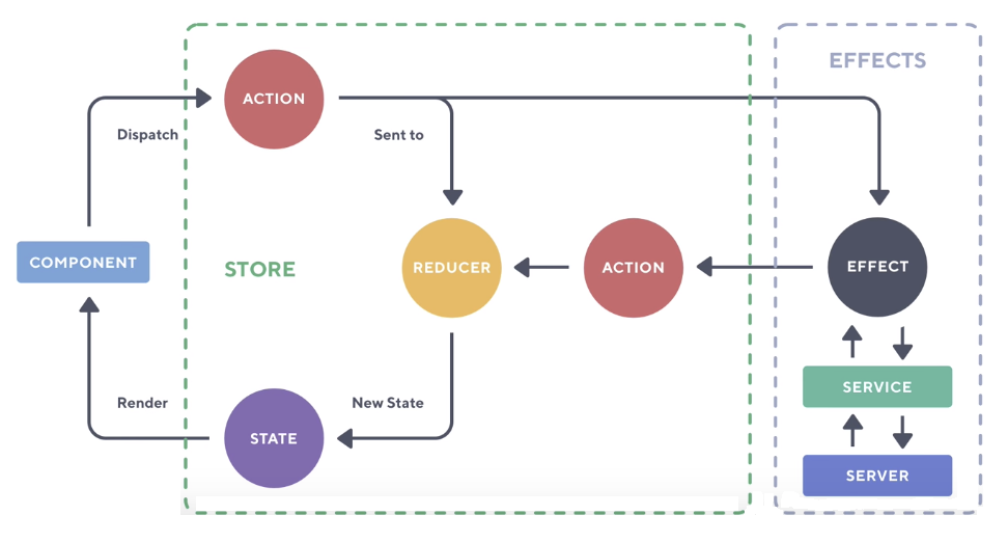
[2.4 System behavior diagrams 9](#_Toc57376505)

[3 UPDATED PROJECT SCHEDULE 16](#_Toc57376506)

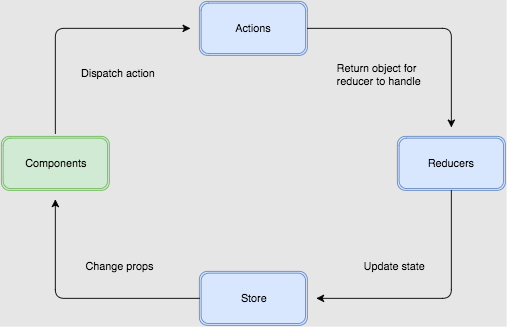
# SOFTWARE ARCHITECTURE DIAGRAM

Software architecture diagram and its description. Basically react is not based on any architecture like MVC. Used Redux state management in this project.

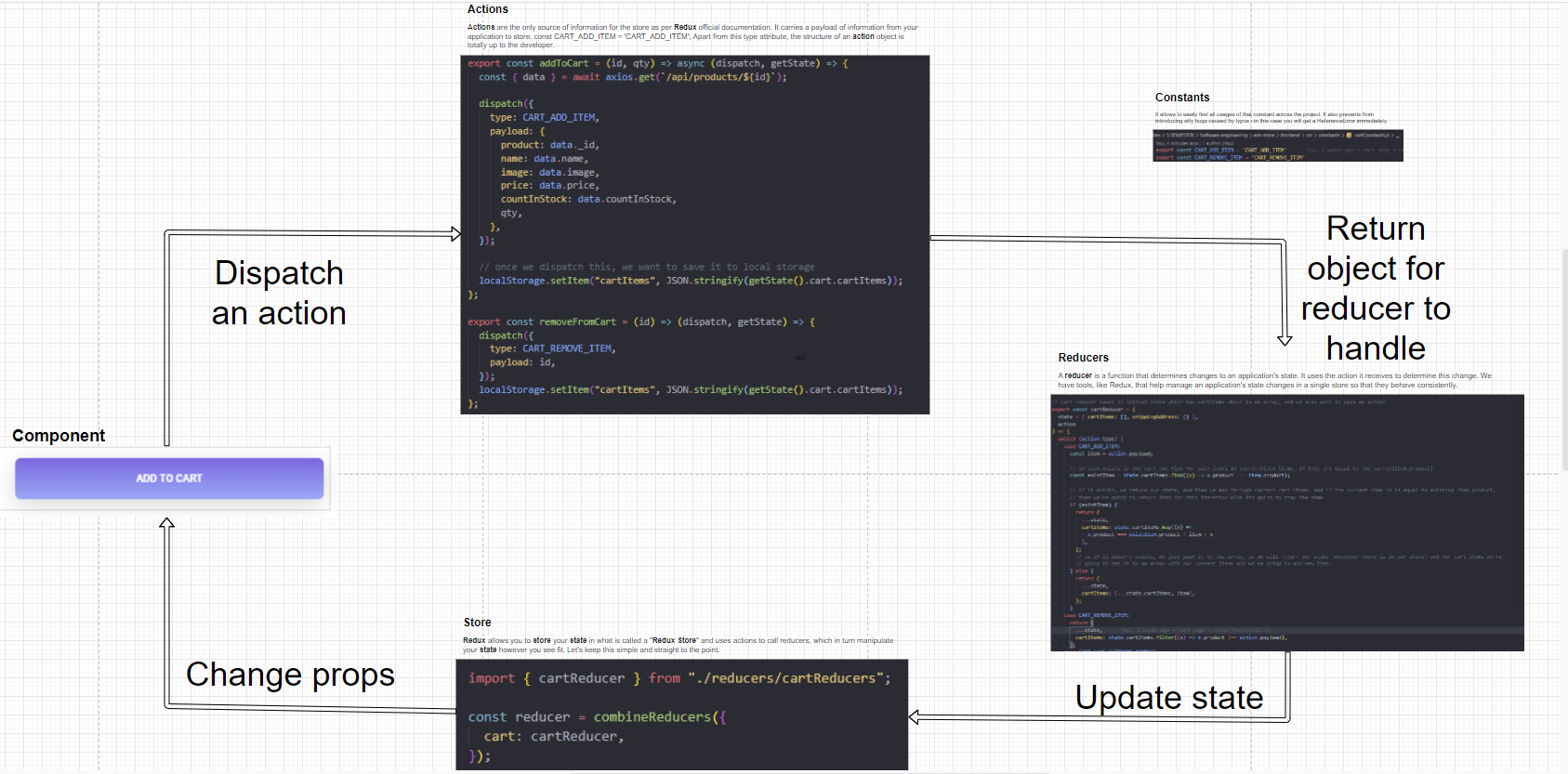
Once we click a button, we trigger (dispatch) an action. An action in redux returns an object and gives it to a reducer to handle it. Those reducers are put into a store in one big combined reducer. And then we re-render our components depending on the state. There are also effects that are used when we have to make requests and fetch information from our backend.



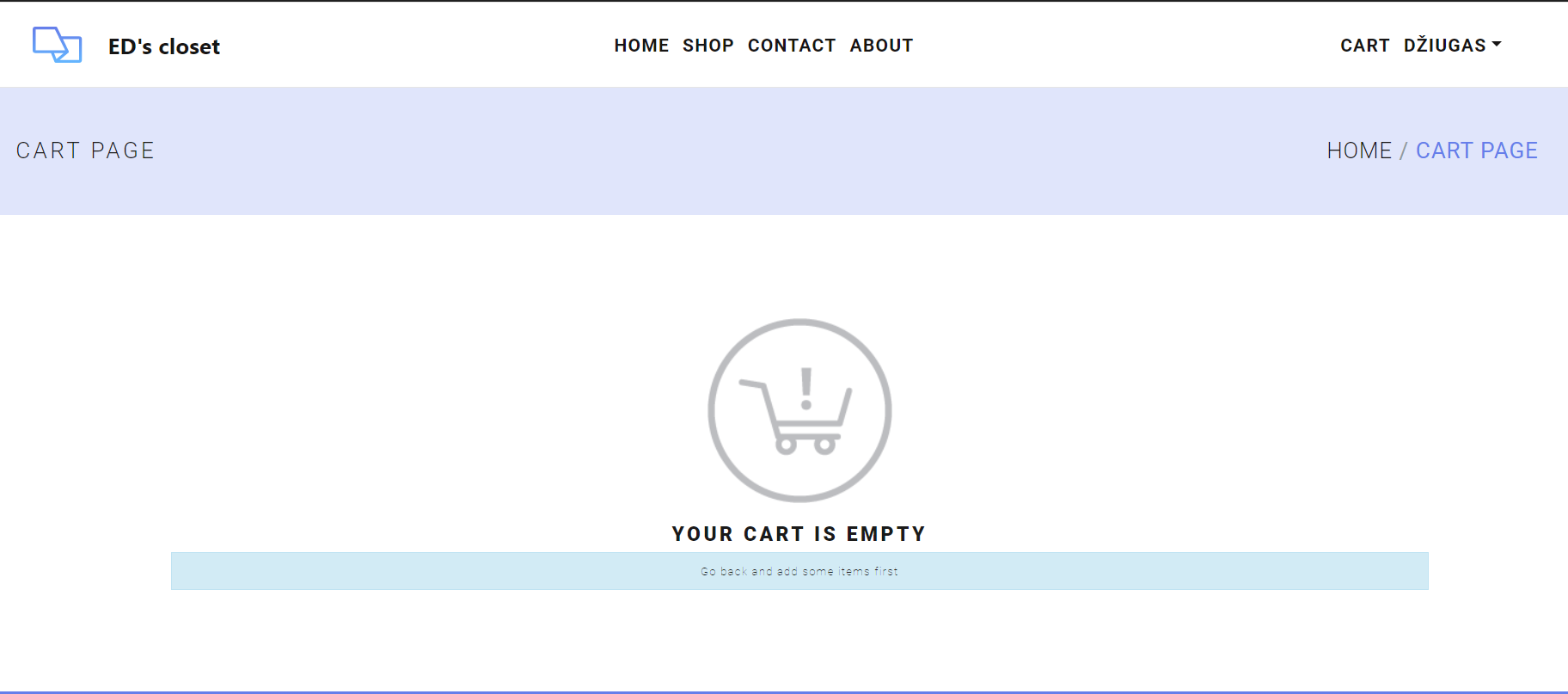
Here’s a simpler graph without any react effects.

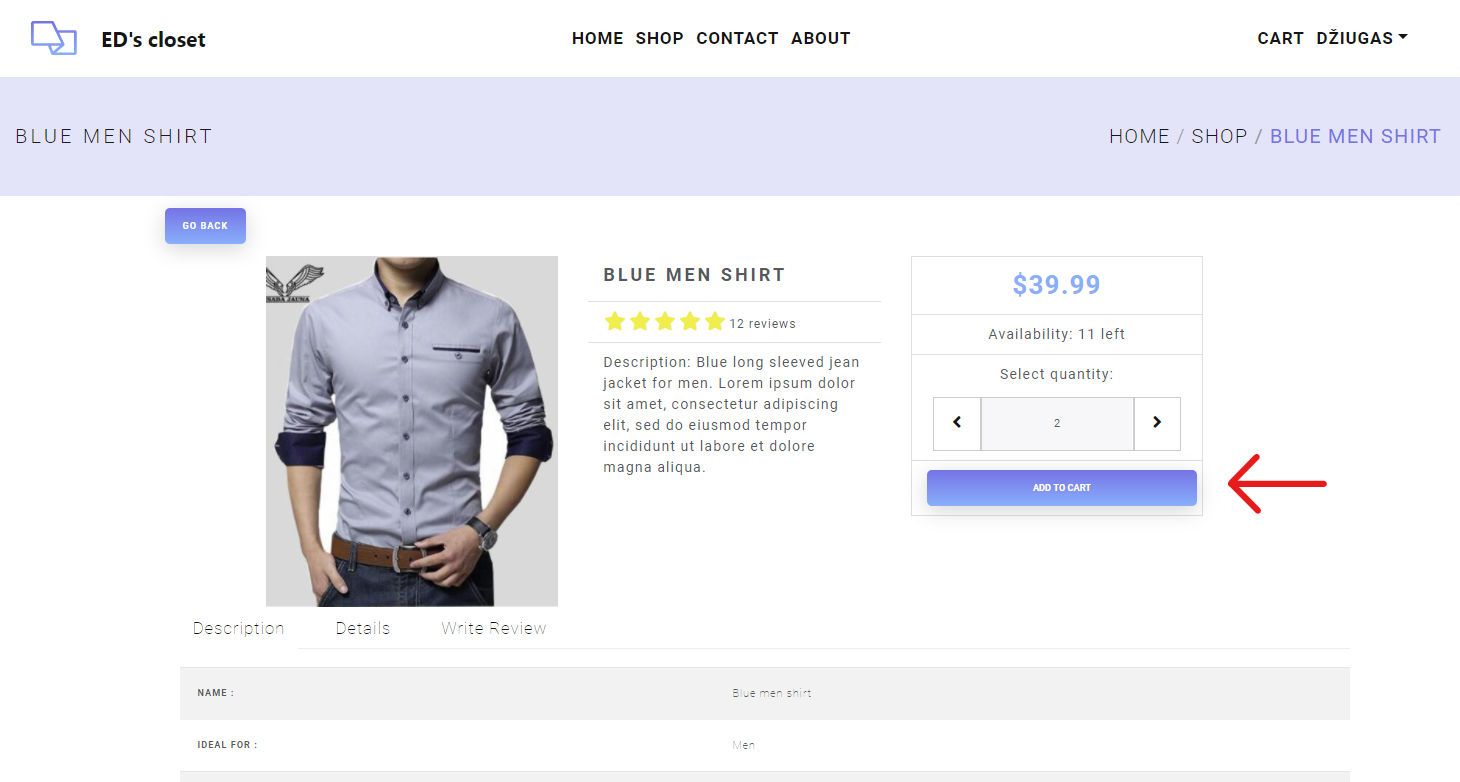


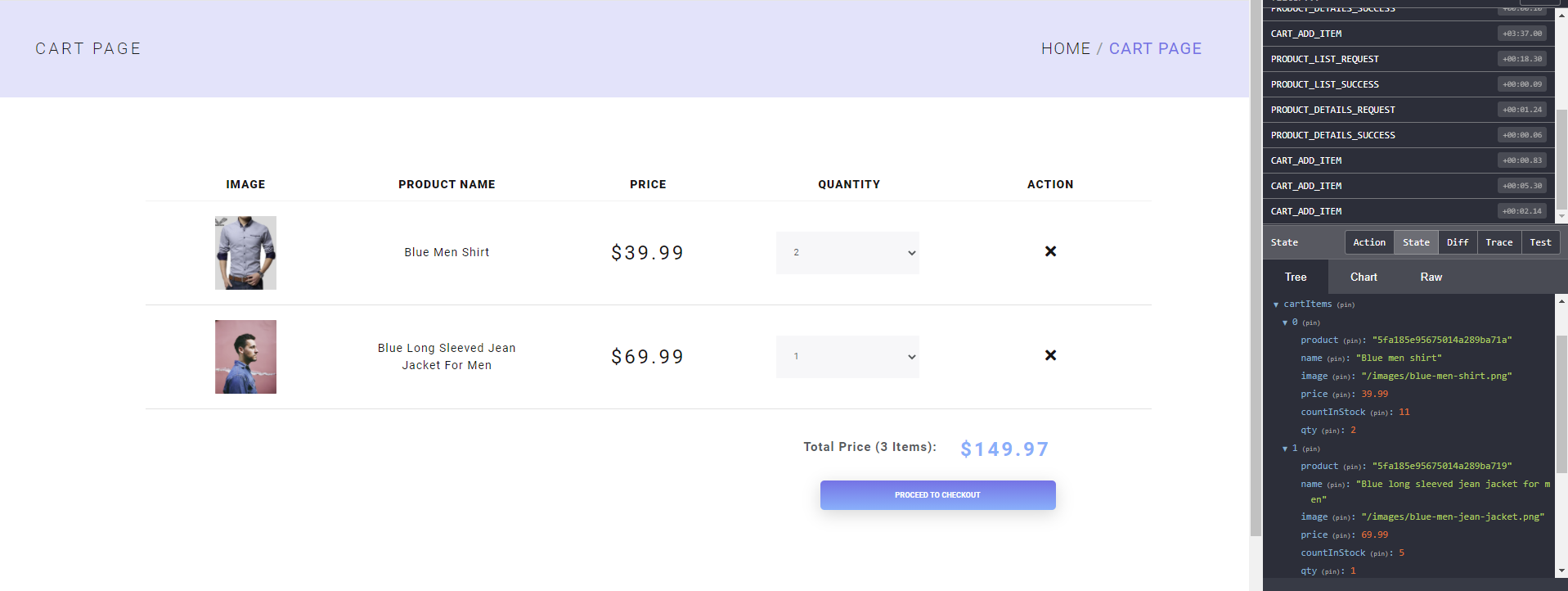
This is how add to cart flow looks like.

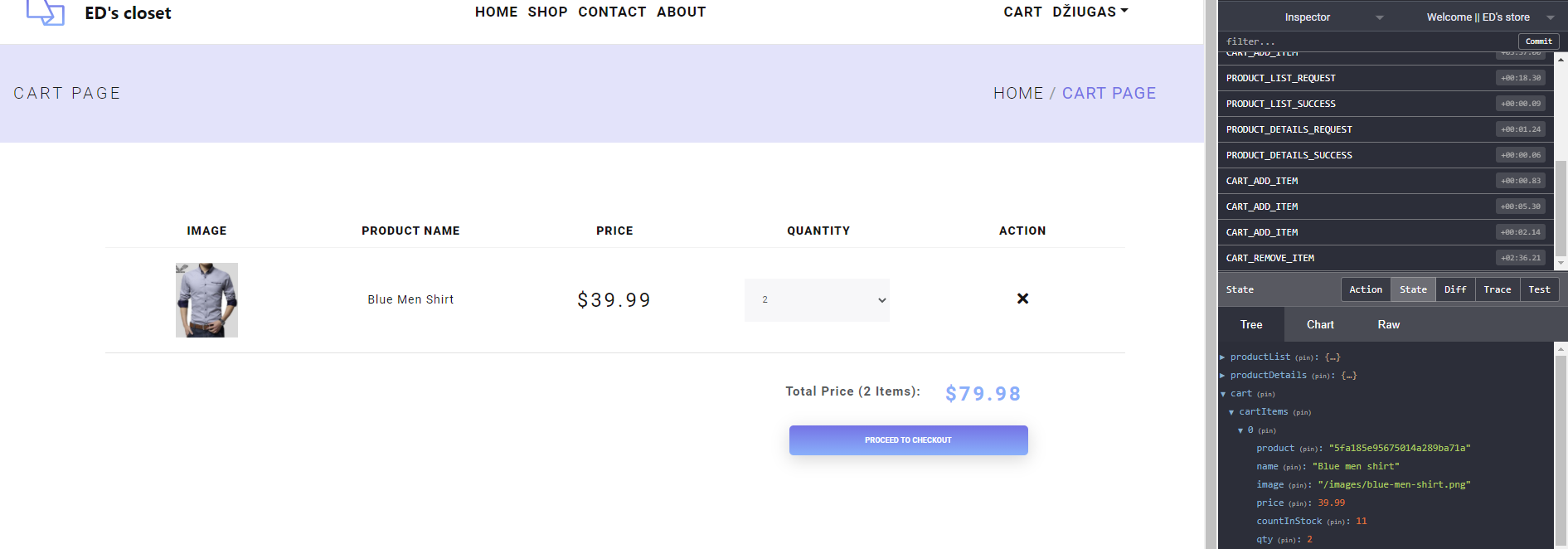


And here how it looks on the user end









# SOFTWARE DESIGN MODEL

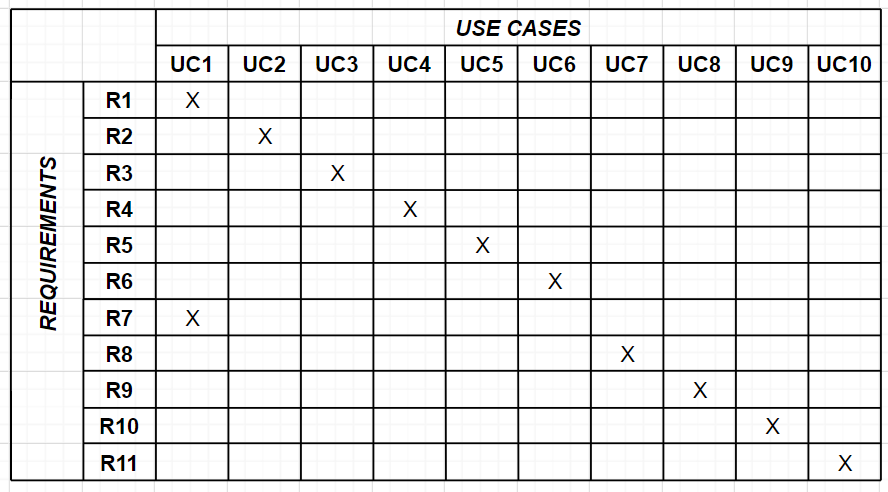
## Functional requirements traceability table

Functional Requirements:

1. The system should allow users to view product collections.
2. The system should allow users to add item to shopping cart.
3. The system should have Sign-in window where user can login.
4. The system should allow users to check out and buy an item.
5. The system should allow users to search for the product.
6. The system should allow users to read and write reviews.
7. Admin should be able to view products in the system.
8. Admin should be able to view orders in the system.
9. Admin should be able to update the order.
10. Admin should be able to edit/delete products.
11. Admin should be able to add products into the system.

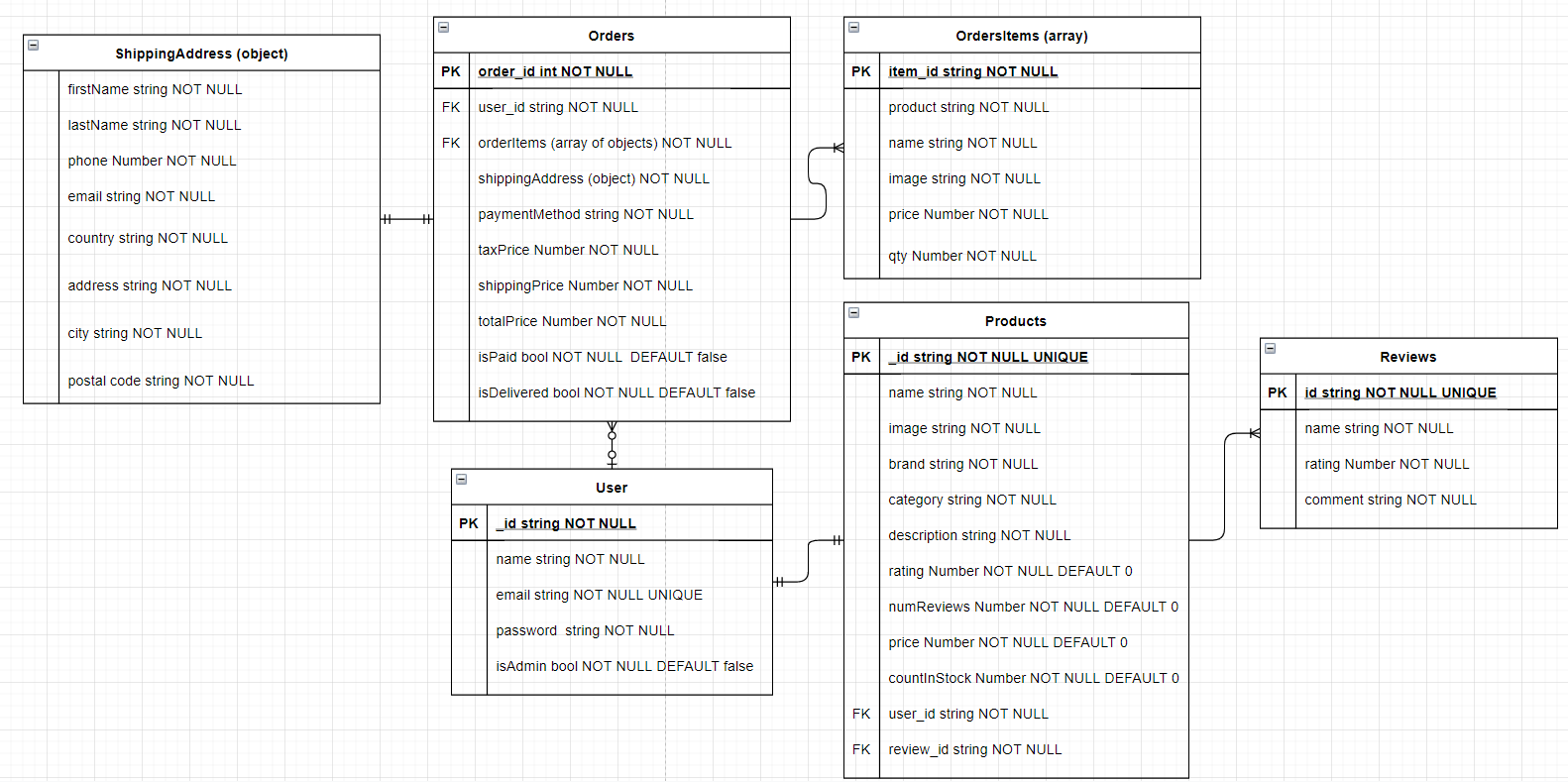
Diagram

Description automatically generated

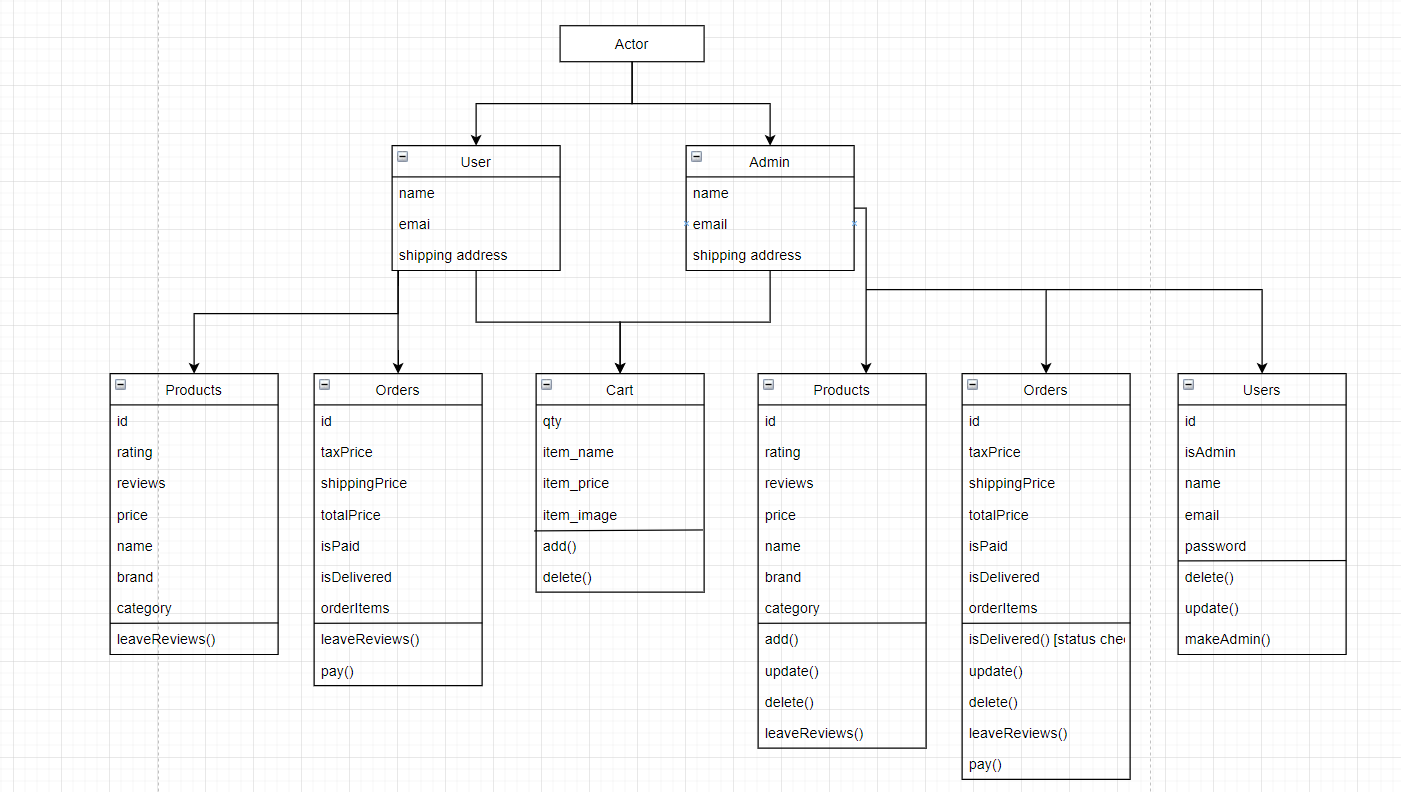


## Entity-relationship diagram

This is a mongoDB NoSQL database.



## Structure diagram



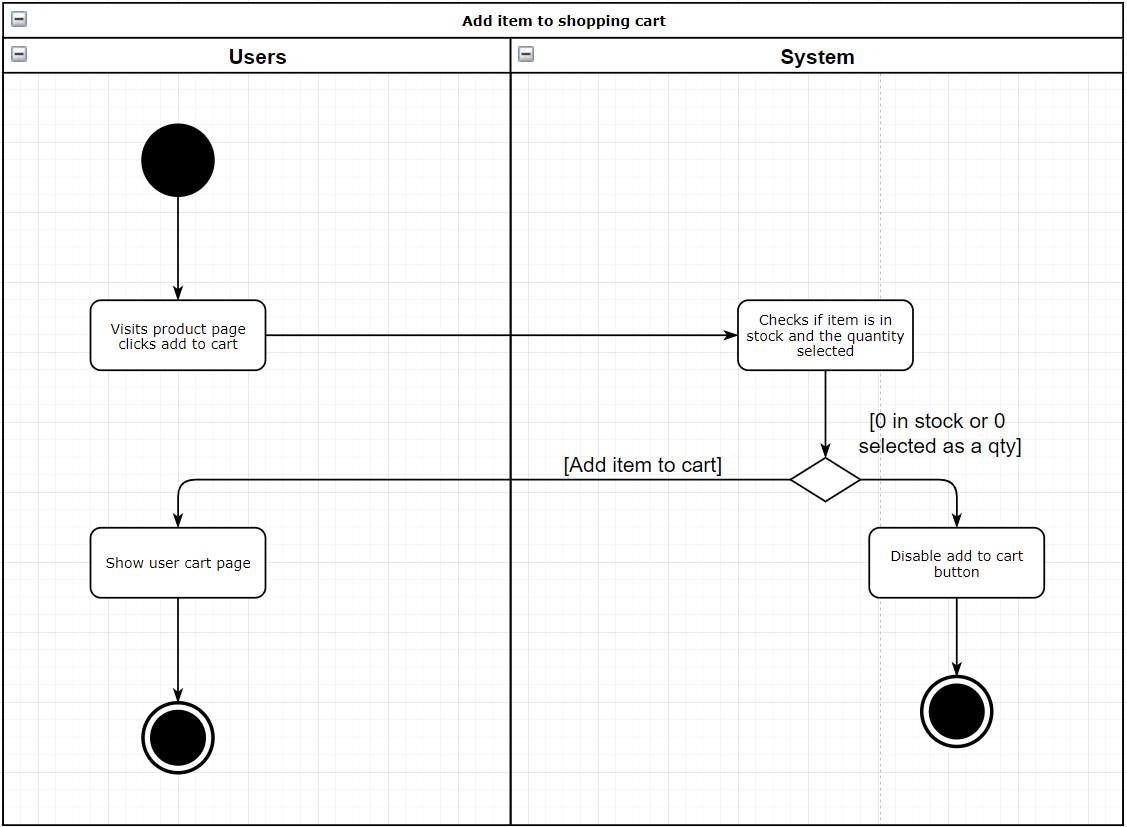
## System behavior diagrams

**activity diagram for each use-case and state diagram for the whole system**

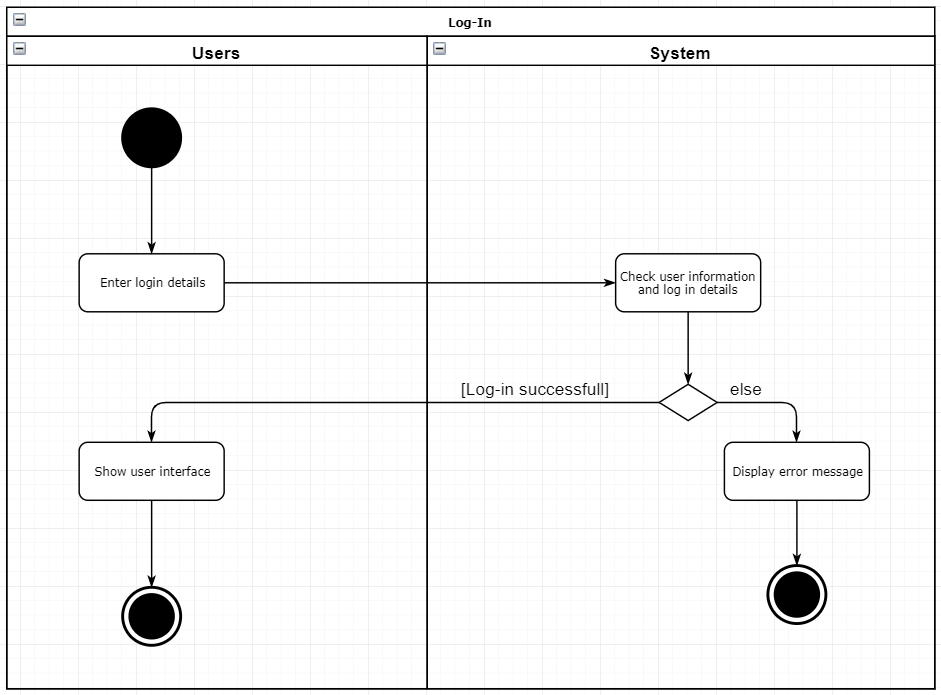
Diagram

Description automatically generated

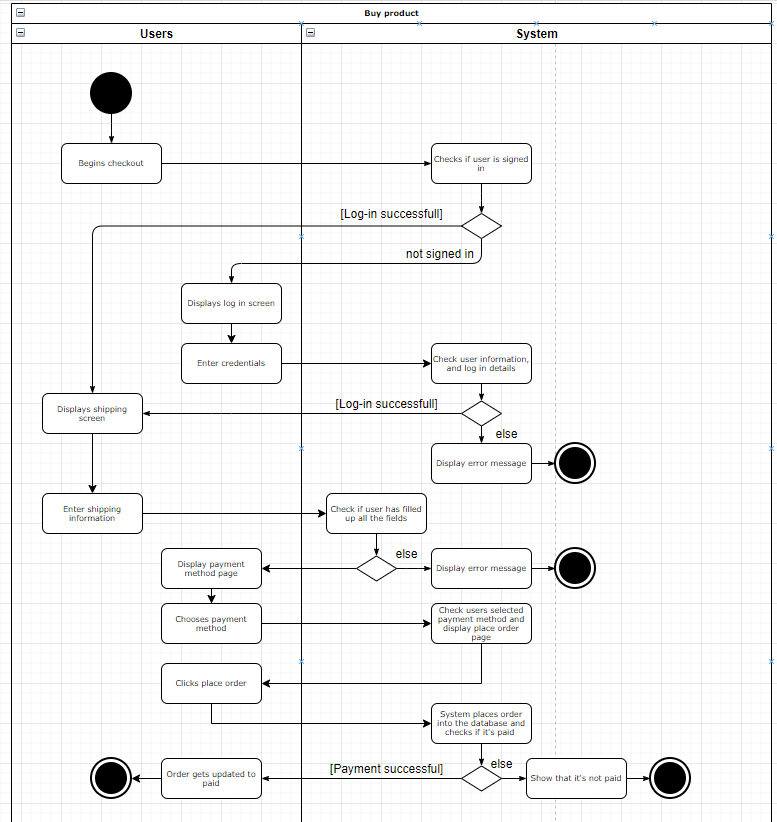
1UC-2 Add item to the shopping cart



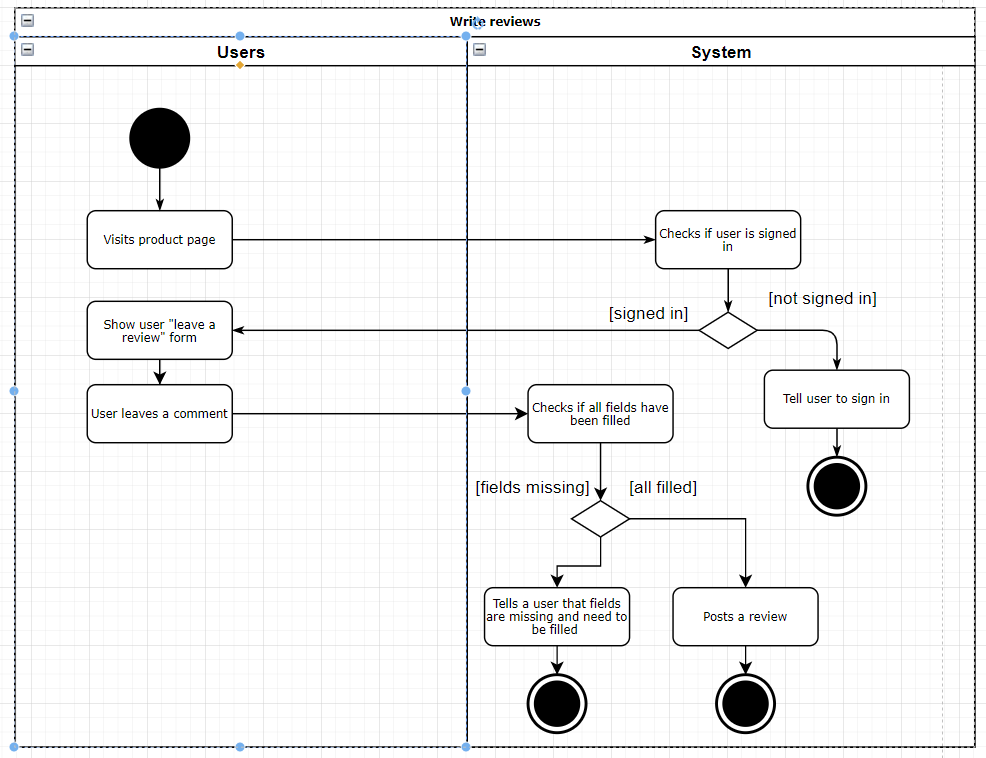
2. UC-3 Login



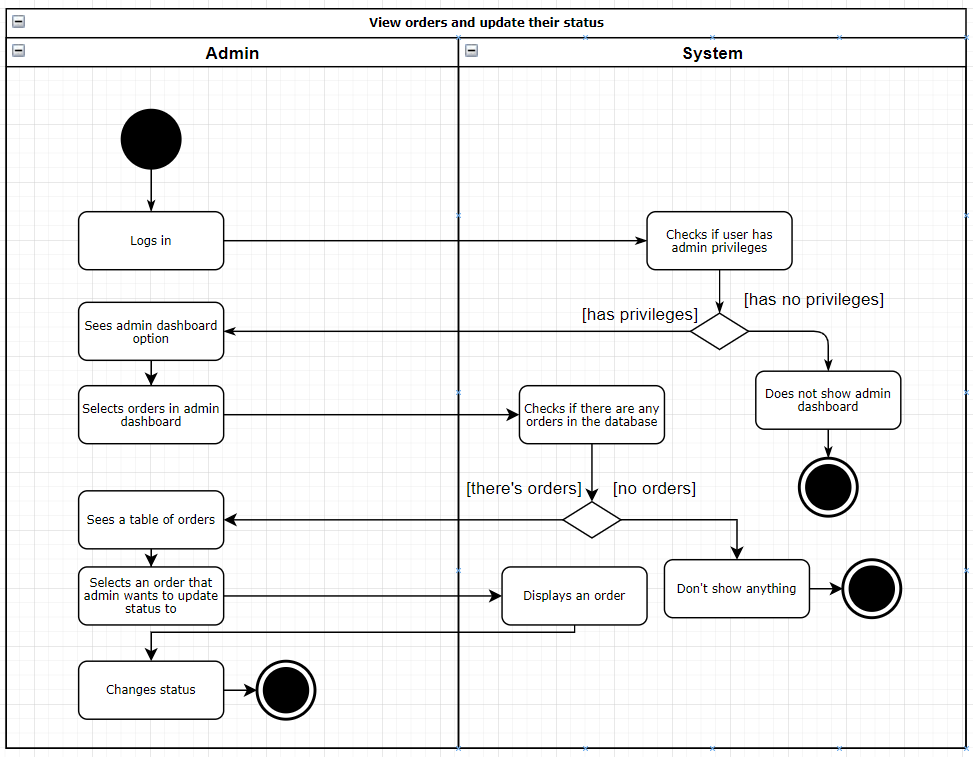
3. UC4 - Buy product



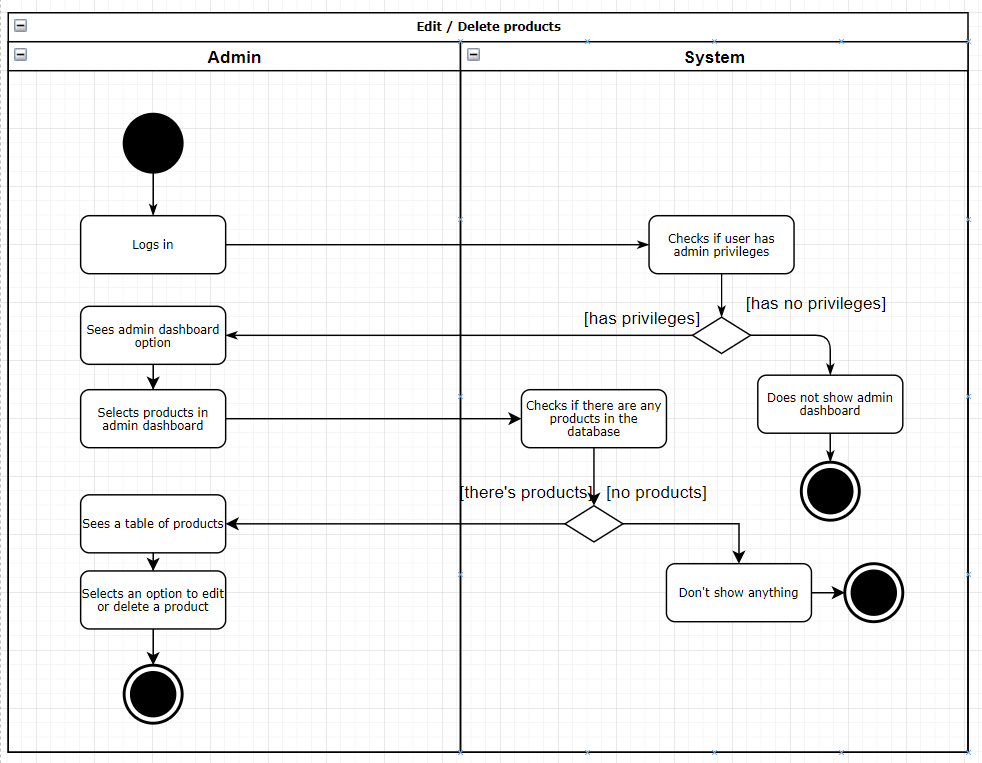
4. UC6 - Write reviews



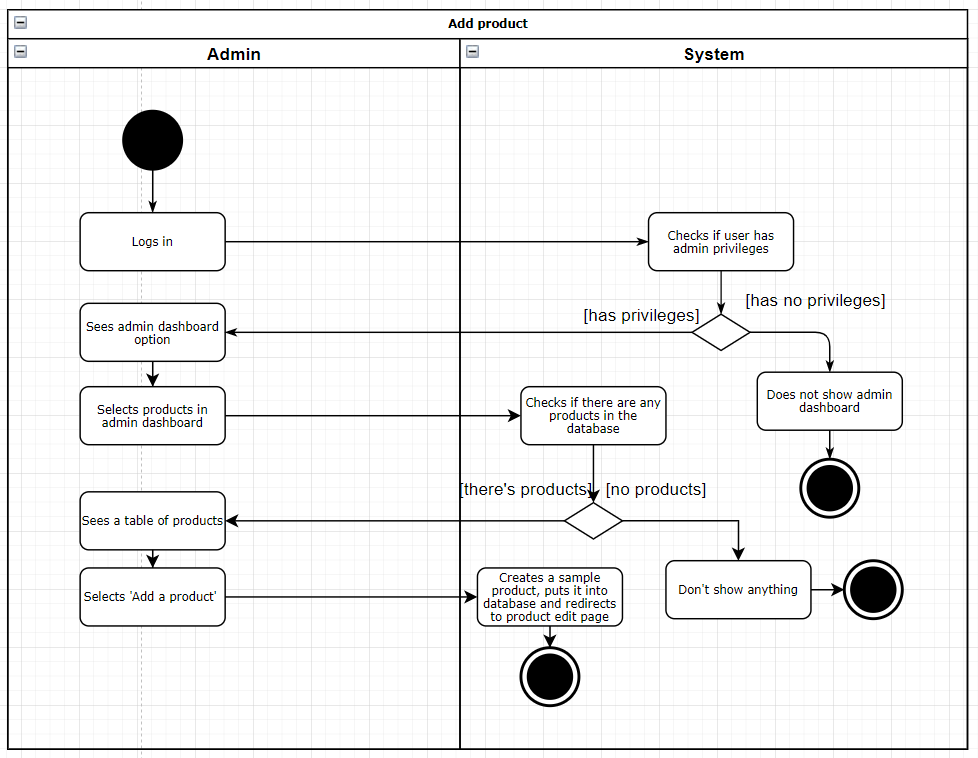
5. UC7& UC8 - view orders & update their status



6. UC9 - Edit / Delete products

**

7. UC10 - Add a product



# UPDATED PROJECT SCHEDULE

Update project schedule to be sure that you will be able to fully develop your system

