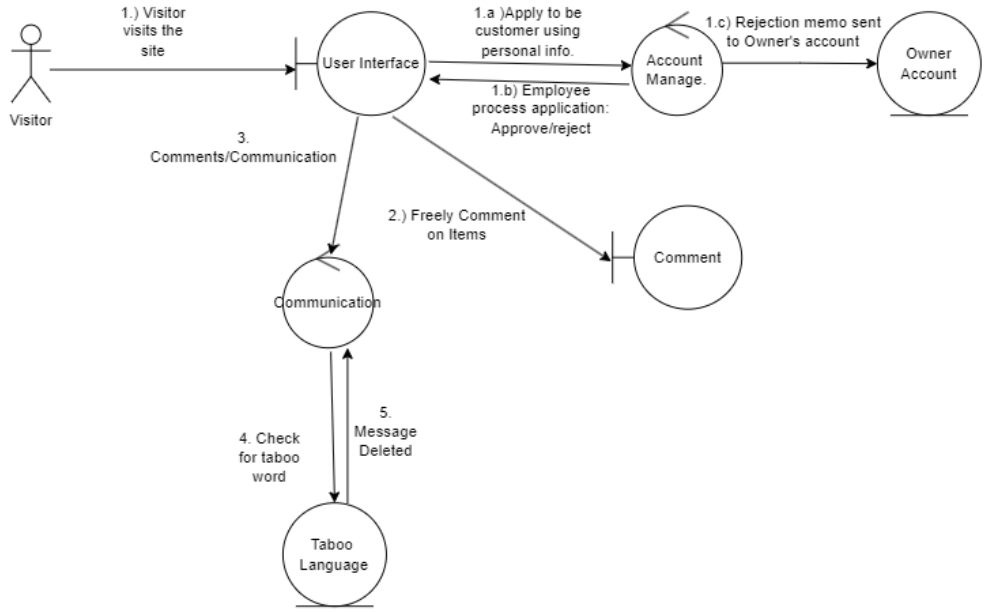
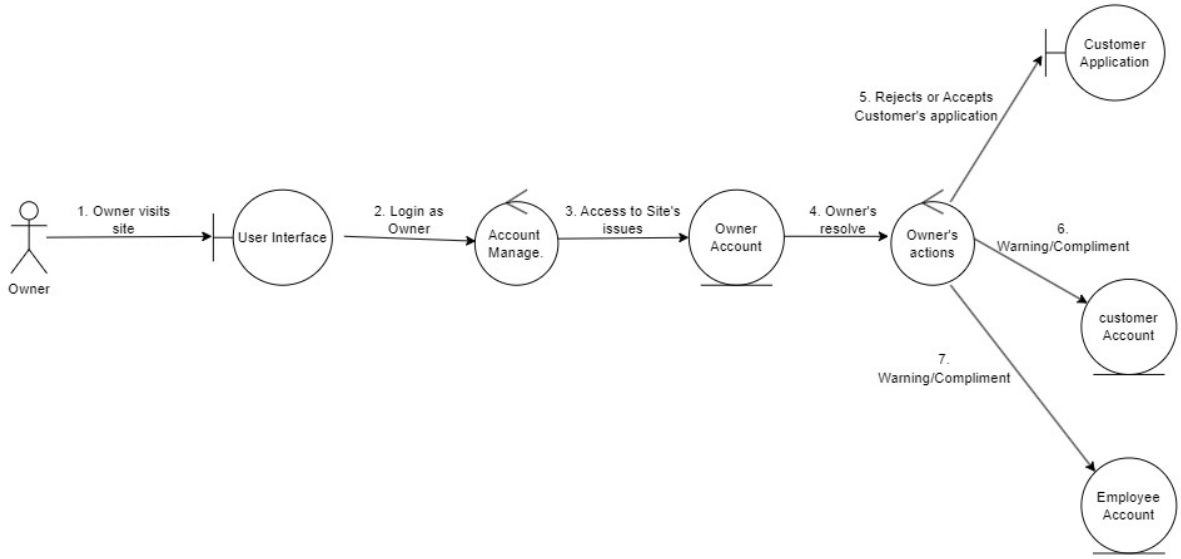
Group Q
Phase II: Design Report

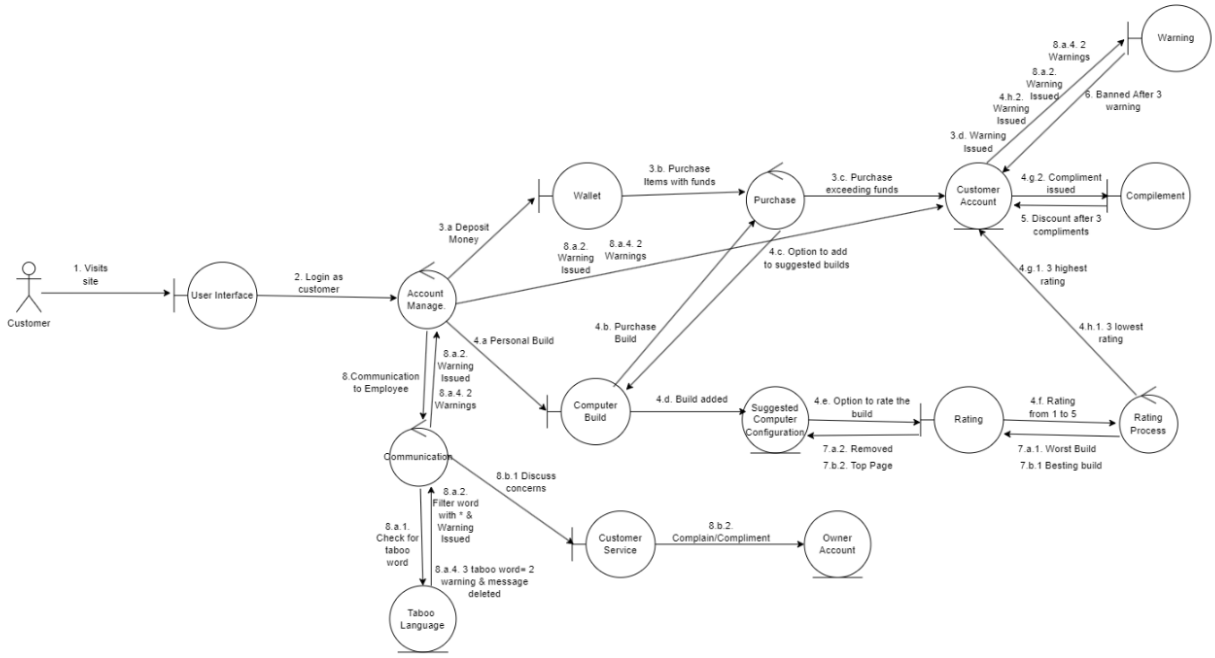
1. Introduction
an overall picture of the system using collaboration class diagram



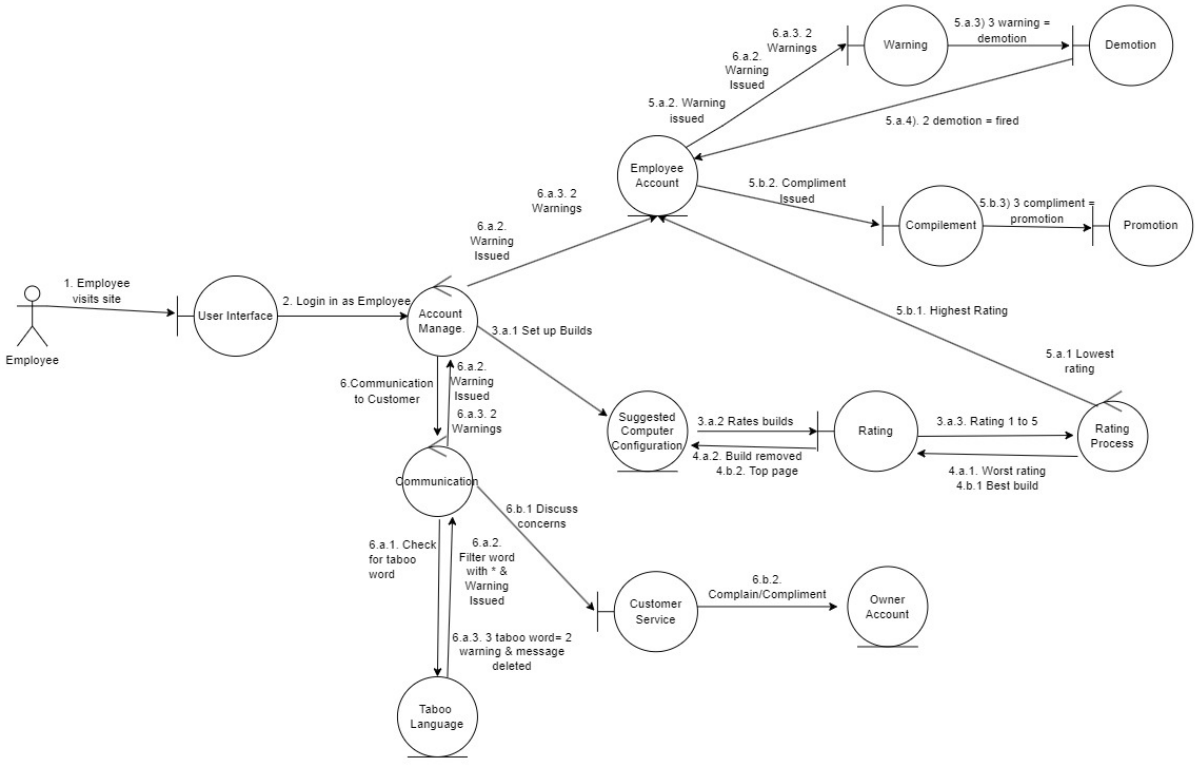
Visitor: Collaboration Diagram



Owner: Collaboration Diagram



Customer: Collaboration Diagram



Employee: Collaboration Diagram

In an overview of the whole system, in order to properly design the collaboration diagram we must first identify all the actors, control classes, input/output classes, and data classes.

Actors: People who will interact with the system

- **Visitors:** Anyone who visits the site, can apply to be a customer and comment on items.
- **Customers**: Anyone is able to apply to be a customer, can purchase items, build personal computers, rate, and talk to employees
- **Employees**: Anyone who works for the store, they are tasked with building suggested builds, assisting customers, and moderation: Banning and discounts to Customers.
- **Store Owner**: The Store owner can process many actions, such as handling customer complaints, warnings, and compliments to customers and employees. Handle management issues such as demotion and promotion.

Control Class: Classes that process/control data

- **Account Management:** Manages the actor's accounts
- **Communication:** Process the comments and communication between visitors, customers, and employees
- **Owner's actions:** The owner process decision relating to warning/compliments and promotion/demotion
- **Purchase:** Processes Purchase by customers
- **Rating processes:** Processes the rating scores of products and gives out warnings/compliments

Input/Output Classes: Represents data that is inputted or outputted by the system

- **User Interface:** The users can interact with the websites and decide to signup, log in, view items, etc.
- **Wallet:** Customers are allowed to deposit money into their account
- **Computer Builds:** Customers are allowed to build their own computer builds
- **Warnings:** Warnings are issued to the customers and employee account
- **Compliment:** Compliments are issued to the customers and employee account
- **Rating:** Rating issued to computer builds
- **Customer Services:** A customer is able to talk to an employee about concerns and an employee is able to discuss the issue with the customer
- **Demotion:** After 3 warnings, an employee can be demoted and after 2 demotions, the employee is fired
- **Promotion:** After 3 compliments the employee is promoted
- **Customer Application:** A visitor can apply to be a customer, they can be either approved or rejected
- **Comment:** All users are about to comment on items

Data Classes: classes represent the data stored in the system

- **Owners Account:** All information relating to Owner, such as handling warnings/compliments, complaints, and customer concerns.
- **Customer Account:** All information relating to the customer's account such as warnings and compliments, and the customer account status such as banned, or discounts.
- **Employee Account:** All information relating to the Employee's account such as warnings and compliments, promotions, and demotions.
- **Taboo Language:** A list of words that are banned
- **Suggested Computer Configure:** Computer builds that are pre-built by the employee and customer-built.

Visitors can freely visit the site, where they can interact with the **User Interface** where they can either choose to sign up and be a customer, their application is processed by **Account Management**, where their application can be accepted or rejected by an employee processing the application if the application is rejected, the owner is sent a memo for further inspection if the applicant disagrees. Further, without any account, the visitor can **comment** on items, but the comments are processed by the **communication control**, which regulates the language used in the communication. After checking with the **Taboo Language**, if it contains taboo words the comment is deleted.

Customers can visit the site and sign in to their accounts, through their **Account Management**, they can deposit money into their **wallet,** and with the funds in their account, when they go to **purchase,** if their purchase price exceeds what they have in their wallet, then they will be warned for reckless behavior. Customers can also build their personal **computer build**, and they go on to purchase that build. If the purchase is successful then they have the option of using their build as a suggested **computer configuration**. Further, customers have the option to rate builds, and the rating of their computer build and others are processed by the **Rating process**, which will either reward the customer for their build by issuing a **compliment** to their **customer account**, if they have 3 compliments, then they will receive a discount. Otherwise, if their computer build has a low rating then a **warning** is issued to their **customer account,** if they have 3 warnings then they will be banned, which is reflected on their **customer account.** The Customer can also communicate with **customer service**, their communication moderation process is the same as the visitors, where using taboo words will be met with warnings and deletion of comments. After the **customer service**, they can send the owner a compliment or complaint about the employee that helped them to the **Owner's Account.**

Employees share similar specifications as visitors and customers, as any communication goes through a moderation process, where they are warned and censored for using taboo words. Similar to the customer's interaction with **customer service**, employees can either complement

or complain about customers to the **owner's account**. Employees have the power to set up **suggested computer builds**. Similarly, their builds can go through the **rating process**, where the best builds are complimented, and that **compliment** and 3 compliment allows for the employee to get a **promotion.** Likewise, bad ratings are warned and 3 warnings result in a **demotion.**

The **Store Owner,** like all other users are able to log in to their account, and have access to the **owner's account**. In their account, they are able to see any pending issues such as warnings and compliments, promotions, and demotions of employees. Through the **owner's actions,** they are about to make the final decision on **customer applications**, and issue warnings/compliments to customers and employees' accounts.

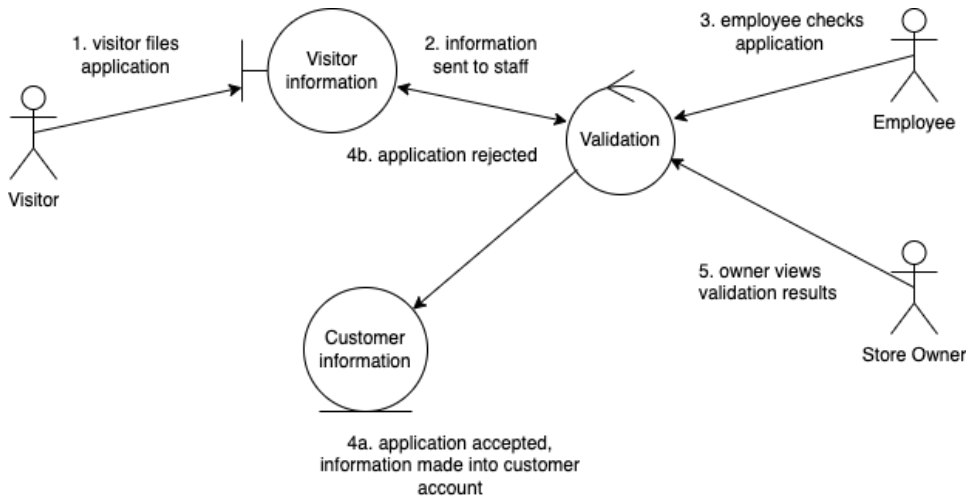
2. All use cases
   - Scenarios for each use case: normal AND exceptional scenarios
   - Collaboration or sequence class diagram for each use case, choose

   3 or more use cases: draw the Petri-nets instead of class diagrams

Customer application

Normal scenarios:
A visitor files an application to be a customer by entering their information and submitting it to the store.
An employee or employees are able to view the visitor's application and can choose to either accept it and promote the visitor to a customer and grant them access to customer actions, or reject the application with reasoning provided.
A store owner is able to view any rejection reasoning made by employees, if necessary.

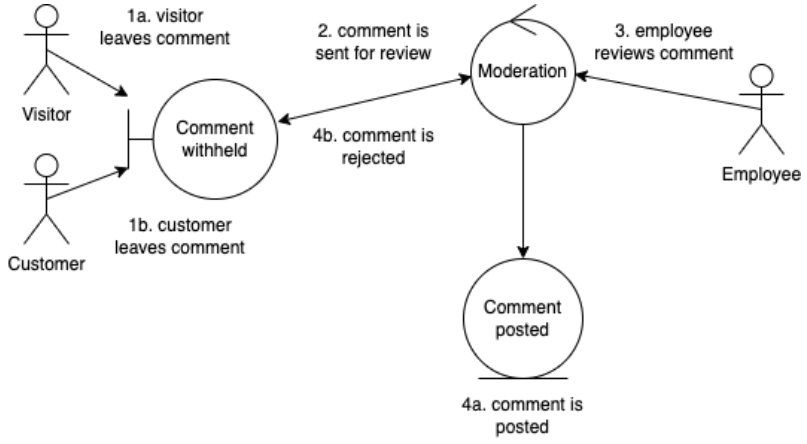
Exceptional scenarios:
The customer application cannot be filed successfully if a visitor does not provide sufficient information.
An application cannot be sent in cases of website or server outages.



Comments

Normal scenarios:
A visitor or customer is able to place a comment on a computer build.

Exceptional scenarios:
A visitor or customer cannot leave any comment that includes explicit or inappropriate language: such comments will not be left on the computer build and the appropriate amount of warnings will be sent to the offender.
A banned customer account cannot leave comments.
Any comment with appropriate behavior cannot be placed in cases of website or server outages.

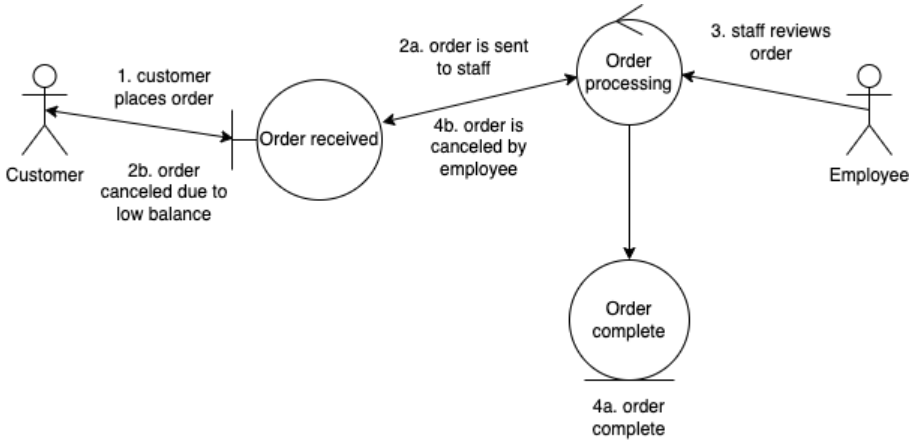


Orders

Normal scenarios:

A customer can place an order for computer parts and/or builds with sufficient information and payment balance.
An employee is able to confirm and fulfill successful incoming orders sent by customers.

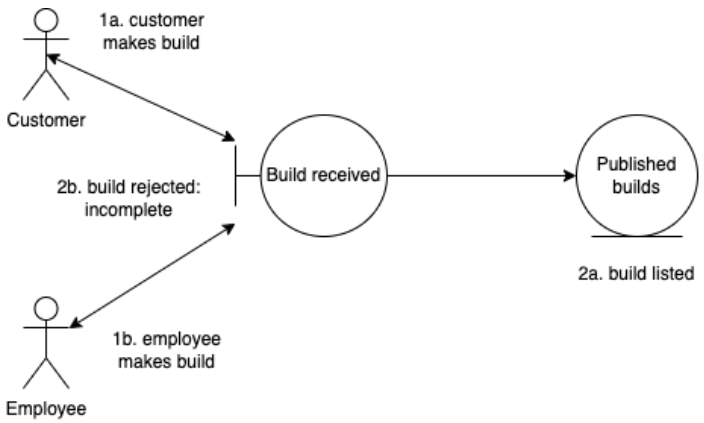
Exceptional scenarios:
A customer's order is rejected if their order cost exceeds their account balance. The order is canceled and the customer is sent a warning.
A banned customer account cannot place orders.
An employee can also cancel an order for any reason, particularly if the store has run out of stock of a certain item.
Any order cannot be placed in cases of website or server outages.



Builds

Normal scenarios:
A customer or employee can list a computer build using several parts that can be bought from the store. Customers have the option to buy the build after making it.

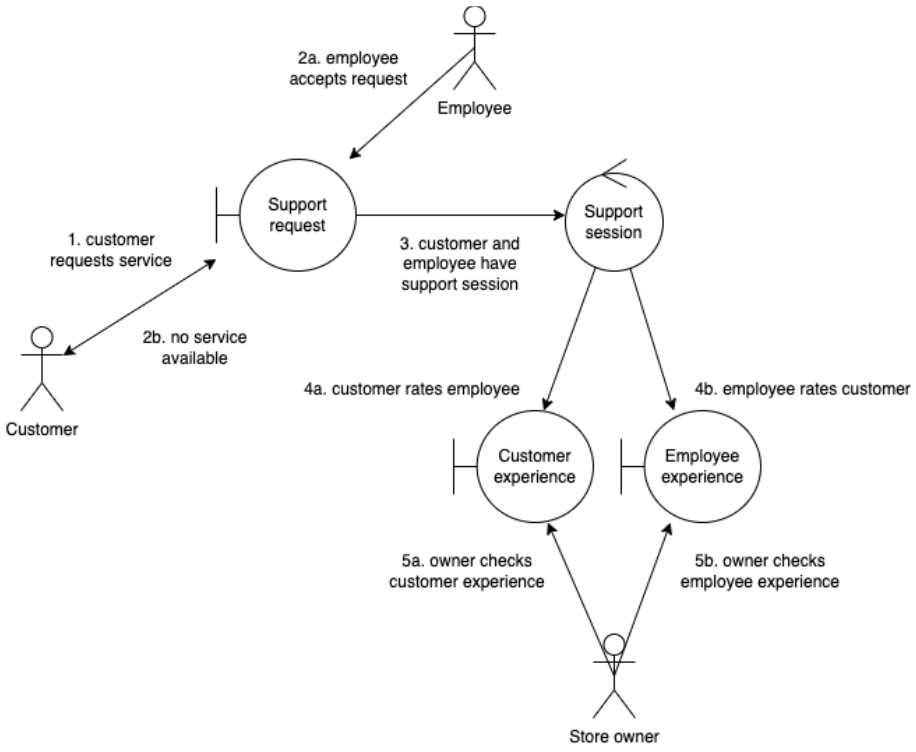
Exceptional scenarios:
A build cannot be created by any user if it contains missing or incompatible parts that prevent the computer from functioning properly.
A banned employee cannot create a build.
A terminated employee cannot create a build.
Creating or ordering a build cannot be done in cases of website or server outages.



Customer support

Normal scenarios:
A customer can speak with an employee regarding any business related to the computer store, such as product inquiries, order or shipping inquiries, and other general questions or complaints.
After a session of customer support, the customer and employee can rate their experiences positively or negatively with reasoning for their choice.
A store owner can review the conversation between the two parties, and issue a number of compliments or warnings to either party if they deem it necessary.

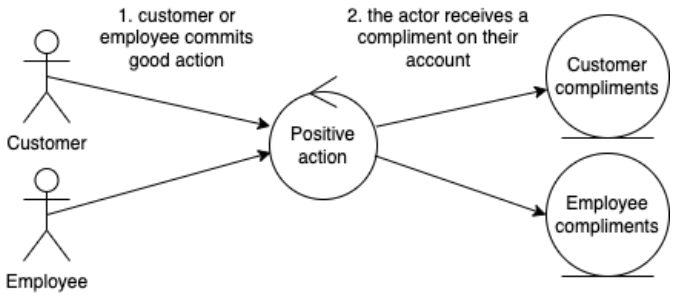
Exceptional scenarios:
Customer support service cannot be offered by employees when they are offline or unavailable (i.e. outside of store hours).
A banned customer account cannot request customer support.
A terminated employee cannot provide customer support.
Customer support service cannot be given to customers, provided by employees, or monitored by store owners in cases of website or server outages.



Compliment

Normal scenarios:
A complement may be issued involuntarily to customers or employees for positive behavior in any of the necessary use-cases above. This may give the recipient extra perks, such as coupons or a promotion.

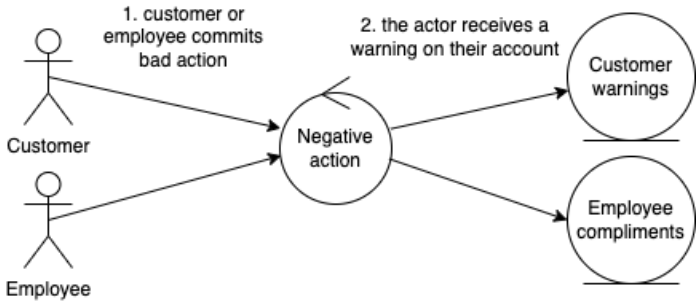
Exceptional scenarios:
Complements cannot be rewarded to recipients in cases of website or server outages, or if the supporting information necessary to award the complement is lost before it can be awarded.
A banned customer account cannot receive compliments.
A terminated employee cannot receive compliments.



Warning

Normal scenarios:
A warning may be issued involuntarily to customers or employees for negative behavior in any of the necessary use-cases above. This may take away features of the computer store from the recipient, or cause further punishments, like suspension.

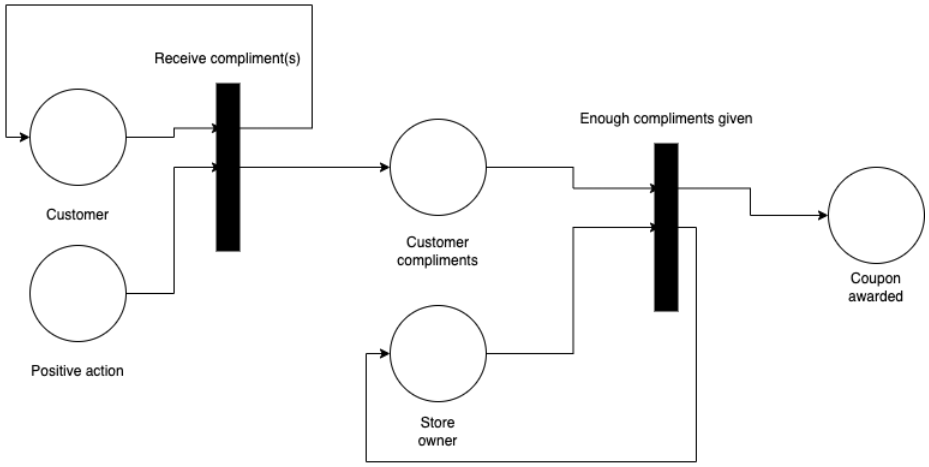
Exceptional scenarios:
A banned customer account cannot receive warnings.
An terminated employee cannot receive warnings.
Warnings cannot be given to recipients in cases of website or server outages, or if the supporting information necessary to give the warning is lost before it can be given.



Coupon

Normal scenarios:
A coupon is awarded involuntarily to customers by store owners who have received a certain number of compliments. They have the choice to apply a certain percentage discount on their next purchase.

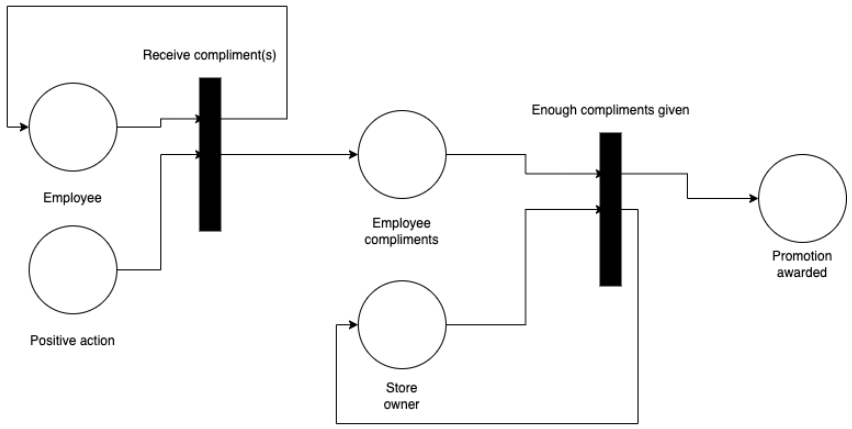
Exceptional scenarios:
A banned customer account cannot receive coupons.
Coupons cannot be rewarded to recipients or redeemed at checkout in cases of website or server outages.



Promotion

Normal scenarios:
A promotion is awarded involuntarily to employees by store owners who have received a certain number of compliments. An employee given a promotion gets a boost in pay.

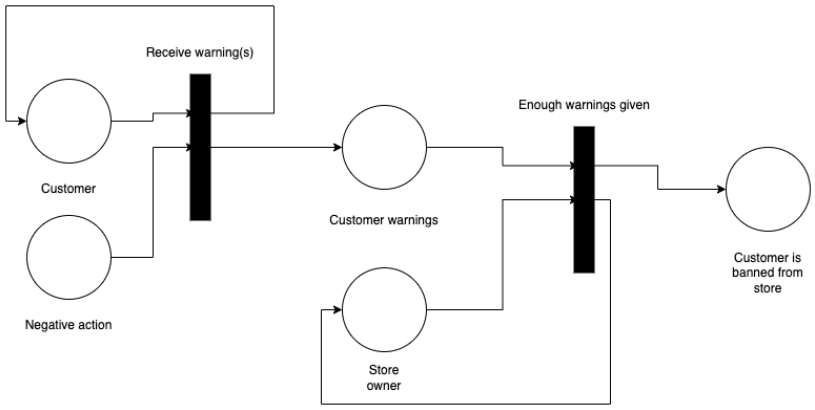
Exceptional scenarios:
A terminated employee cannot receive promotions.
Promotions cannot be rewarded to recipients in cases of website or server outages.



Ban

Normal scenarios:
A ban is sentenced involuntarily to customers by store owners who have received a certain number of warnings. They will no longer be able to access any of the features of the store using their current account, and must speak with an employee at the store in-person to contest the ban.

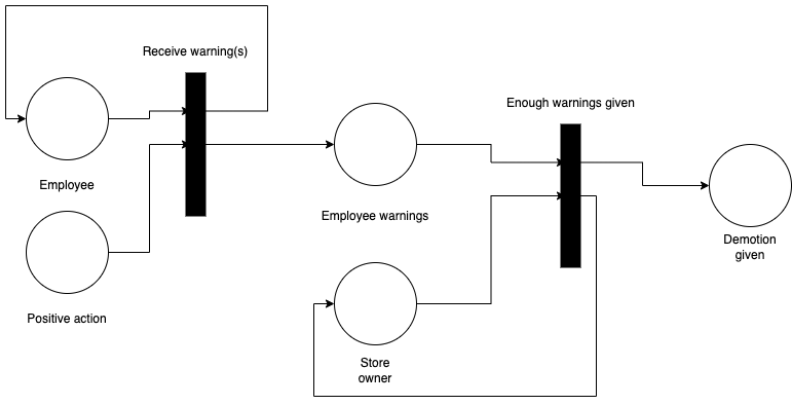
Exceptional scenarios:
Bans cannot be sent to recipients or sentenced by store owners in cases of website or server outages.



Demotion

Normal scenarios:
A demotion is given involuntarily to employees by store owners who have received a certain number of warnings. An employee given a demotion gets a cut in pay.

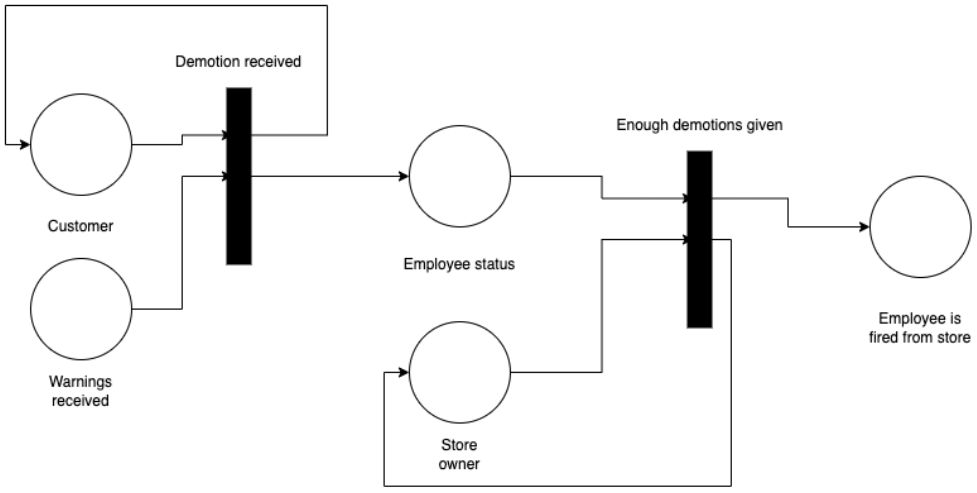
Exceptional scenarios:
A terminated employee cannot receive demotions.
Demotions cannot be given to recipients online in cases of website or server outages.



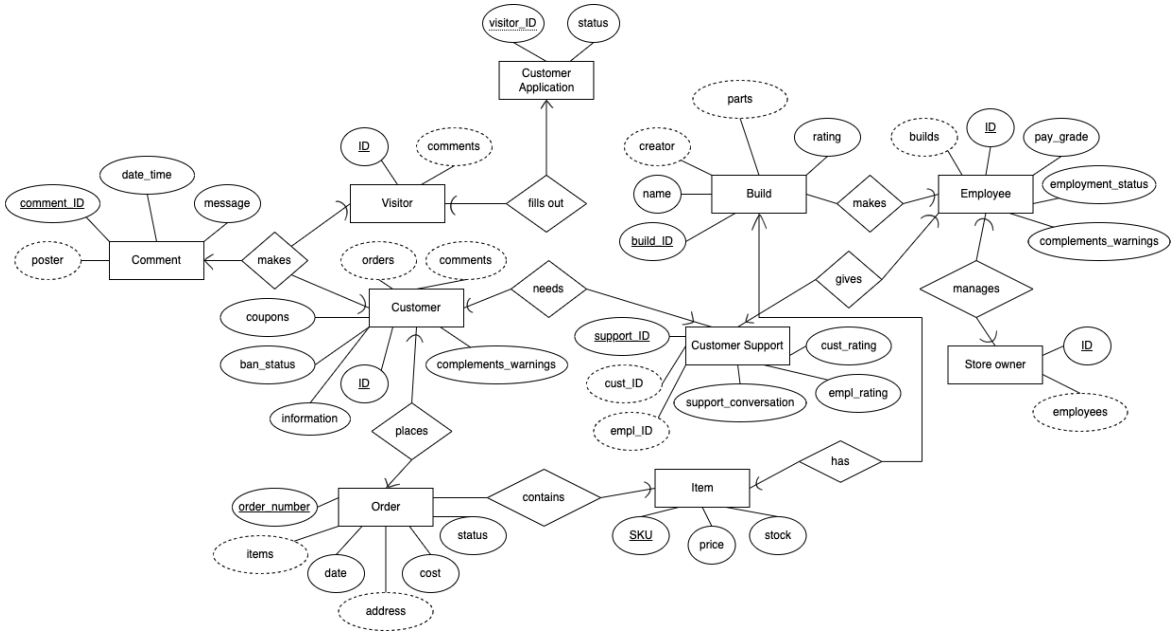
Termination

Normal scenarios:
A termination is given involuntarily to employees by store owners who have received a certain number of demotions. A terminated employee is no longer an employee: they will lose access to their staff account and benefits/features as a staff member of the store.

Exceptional scenarios:
Terminations cannot be given to recipients online in cases of website or server outages.



3. E-R diagram for the entire system
attributes and key for each class should be provided



Visitor:

- Key: ID - visitor's unique internal ID number

- Other attributes:
    - comments - visitor's comments (inherited from **Comment**)

Customer:

- Key: ID - customer's unique internal ID number
- Other attributes:
    - orders - customer's orders (inherited from **Order**)
    - comments - customer's comments (inherited from **Comment**)
    - information - customer's personal information (name, address, balance, etc.)
    - coupons - customer's available coupons
    - complements_warnings - customer's number of warnings and/or compliments
    - ban_status - states whether or not a customer's account has been banned

Employee:

- Key: ID - employee's unique internal ID number
- Other attributes:
    - builds - employee's listed builds (inherited from **Build**)
    - pay_grade - employee's salary
    - complements_warnings - employee's number of warnings and/or compliments
    - employment_status - states whether or not an employee is still working for the store

Store owner:

- Key: ID - store owner's unique internal ID number
- Other attributes:
    - employees - list of active employees (inherited from **Employee**)

Customer application:

- Key: visitor_ID - ID number of visitor who sent the application (inherited from **Visitor**)
- Other attributes:
    - status - outcome of application from staff (and reasoning if declined)

Item:

- Key: SKU - internal retail ID of item
- Other attributes:
    - price - item retail price
    - stock - store's availability of the item

Build:

- Key: build_ID - build's internal ID number
- Other attributes:

    - name - name of the build
    - creator - creator of the build (inherited from **Employee**)
    - parts - items needed for the build (inherited from **Items**)
    - rating - rating of the build by customers

Comment:

- Key: comment_ID - comment's internal ID number
- Other attributes:
    - date - date and time comment was posted
    - poster - person who wrote the comment (inherited from **Visitor, Customer**)
    - message - contents of the comment

Order:

- Key: order_number - order's ID number
- Other attributes:
    - items - customer's items
    - date - date and time order was placed
    - address - customer's delivery address (inherited from **Customer**)
    - cost - cost of the order
    - status - order's status (canceled, shipped, etc.)

Customer support:

- Key: support_ID - support session's ID number
- Other attributes:
    - cust_ID - customer's ID (inherited from **Customer**)
    - empl_ID - employee's ID (inherited from **Employee**)
    - support_conversation - full conversation between employee and customer
    - cust_rating - customer's rating of their experience
    - empl_rating - employee's rating of their experience

4. Detailed design:
   for EVERY method use pseudo-code to delineate the input/output and
   main functionalities

```
customer_application:

# input: visitor information
# output: employee decision

if user is visitor, open application
```

visitor fills out information
if information is complete, allow application to be sent

if user is employee, open all sent applications
for each application, check information
if information is valid, create customer account and return employee decision as valid
if information is invalid, return employee decision as the employee's reasoning for rejection

comment:

 # input: comment message
 # output: comment on item listing in the shop

user enters their comment message
if comment contains inappropriate language, prevent comment from being posted and send user a warning
else, create comment and post on the intended item

build:

 # input: parts needed to build the computer
 # output: listing for the build in the shop

if user is customer, choose build parts
if parts are incomplete or incompatible, customer must rechoose parts
else, customer may buy the build
    if customer does not buy the build, return nothing
    else, customer may choose to upload the build on the store
        if customer refuses, return nothing
        else, post a listing for the build in the shop

if user is employee, choose build parts
if parts are incomplete or incompatible, employee must rechoose parts
else, post a listing for the build in the shop

order:

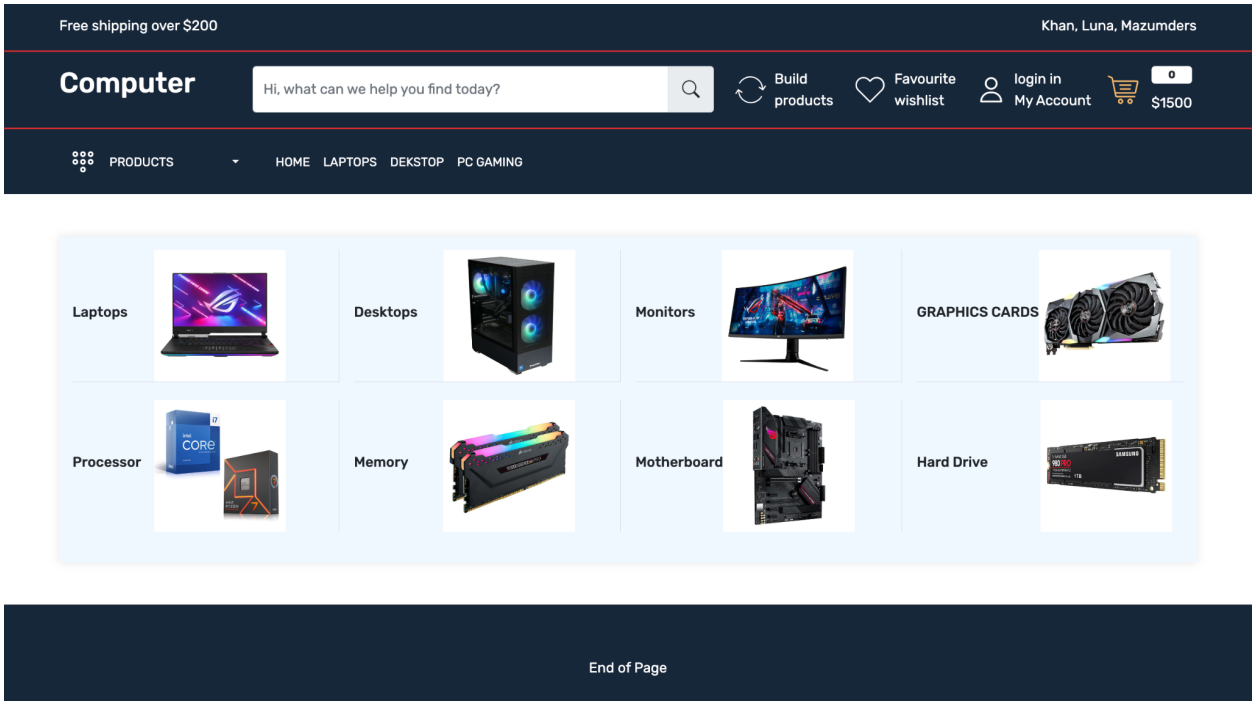
```
# input: customer information, item(s)
# output: order number, order status

if user is customer, add desired items to cart from listings
from cart, customer goes to checkout
customer enters customer information
if customer information is complete, allow customer to place
order
if customer balance is insufficient, cancel order and give
warning
else, accept order, return order number and order status as
pending

if user is employee, check orders with status = pending
for each pending order,
     if order is okay, fulfill order and set status as complete
     else, cancel order and set status as canceled
```

customer_support:

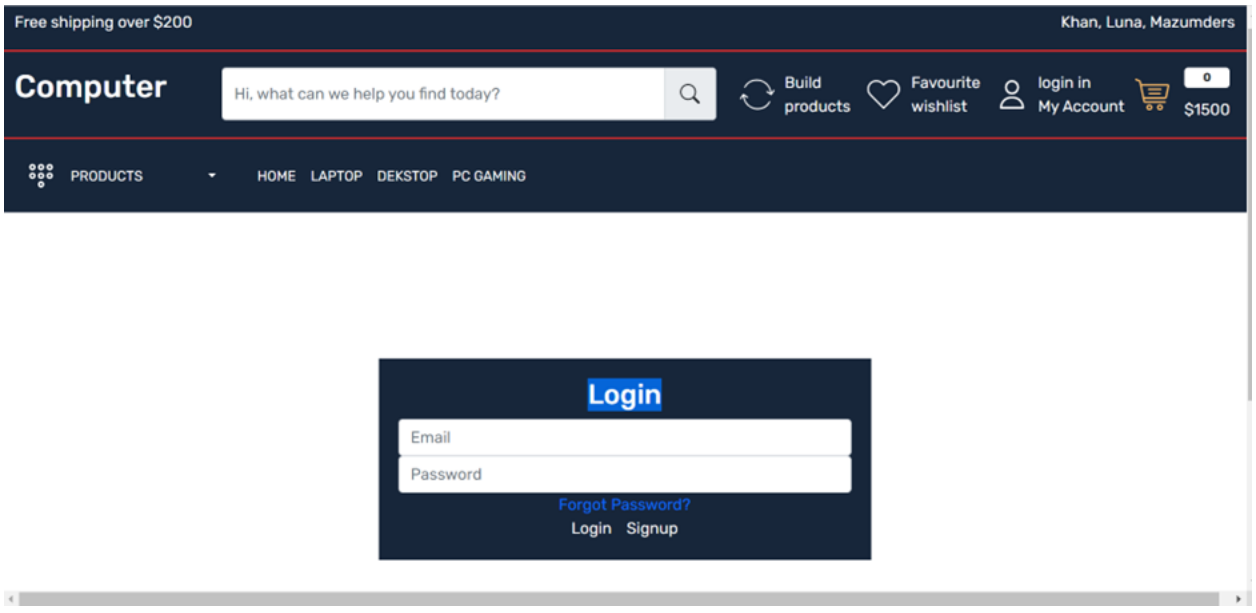
```
# input: user input, user abort, user rating
# output: customer support session object

if user is customer, request support
if no employee is available, cancel support
else, start customer support

if user is employee, check support
if no customer is available, cancel support
else, start customer support

if customer leaves, end support
customer rates experience as positive or negative
if no rating left, set customer experience as positive
employee rates experience as positive or negative
```

5. System screens:
   demonstrate major GUI screens of the system and a sample prototype of one functionality of your own choice.

The home page for our system is shown below. This will be the first page that a visitor or customer sees upon entering the website.



The Login page for our system is shown below. This is where visitors can create a new account or customers / employees can log into their accounts.



6. Memos of group meetings and possible concerns of team work

As a group, we started with implementing the system into the necessary charts, which would allow us to have a framework for the development of our system. This would lead us to then developing the frontend for our system to continue developing on top of. We will continue to continue developing frontend pages and backend functions to complete our computer store system and satisfy its requirements.

Our biggest concern regarding the team work for this system is deciding how to best divide and consolidate the workload. We have to choose the best role that each of us fits when it comes to developing the frontend and backend of the website, especially since some of us have little to no experience in web development.

We have also had some issues with group time management and communication: since we are working democratically, we haven't been able to find the time to fully work together because some of us in the team have other outside responsibilities to attend to at a given moment. This also causes the same problem for our communication: some of us aren't able to hold a full discussion at the same time, so it creates some gaps of time between when we are able to talk to each other regarding our project, further producing some inefficiencies.

7. Address of the git repo (github, gitlab, bitbucket, etc) of your team's work so far - put all materials including this report there

The address of the GitHub repository can be found below:
https://github.com/ishtiaq-khan/computer-store