CS 3354 Software Engineering
Final Project Deliverable 2

Storage Hub

Kuanlin Liu
Cameron Glick
Jesse Ladyman
Kevin Tran
Khiem Huynh
Yeswanth Bogireddy
Zareef Kabir

Delegation of Tasks:
Software process model and requirements. [Kuanlin and Kevin]
Diagramming models (use case, sequence, class). [Cameron and Jesse]
Architectural design. [Khiem and Zareef]
Project scope. [Yeswanth]
Project scheduling. [Cameron]
Cost, effort, and pricing estimation. [Yeswanth]
Estimated cost of hardware and software products. [Kuanlin]
Estimated cost of personnel. [Zareef]
Test plan for a unit of software. [Khiem]
Comparison of work to similar designs. [Kevin]
Evaluation and conclusion. [Jesse]
References. [All]

Project Deliverable 1 Content Begins

Project Description:
This software would be a storage system that would act as a complete inventory management of the stock it needs to keep track of. It would allow users to order and request items as well.

Instructor Feedback:
Please make sure that you include the following in your final project deliverable 2:

a) A comprehensive research to find similar project implementations. Cite your findings properly using IEEE citation format.

b) Make sure that you are adding extra feature(s) to uniquely differentiate your design from already existing similar implementations. Clearly explain what these feature(s) are.

c) A comparison of your design with similar implementations in the field. This comparison could be presented in any format of your choice, such as a table, paragraphs, charts, etc.

Proposal:
To address these points, we will look at other softwares like Amazon and eBay to analyze the similarities and see how we can make our program more unique. The comparison will most likely be in a venn diagram. We plan to add specifications towards the warehousing aspects of the market.

GitHub Repository URL: https://github.com/KT-CSML/3354-StorageHub

Project Scope:
1.1 Inventory
  1.1.1 Stock - Number of items available.
  1.1.2 Updating Stock - Items that are received for stocking.
  1.1.3 Location of Stock in Market Place
  1.1.4 Misc Items
1.2 User Account
  1.2.1 Purchase items
  1.2.2 Request Items
  1.2.3 Refund Items
  1.2.4 Add & Delete Items
  1.2.5 Pay by Credit Card or Gift Card
  1.2.6 Pick up from Store
  1.2.7 Delivery to House
1.3 Market Location
  1.3.1 Number of Market Places
  1.3.2 Available Items
1.4 Discounts
  1.4.1 Items on Sale
  1.4.2 Employee Discount

Software Process Model:
The incremental process model would most likely be the best option to apply to this program as the storage system would initially be basic, with standard usage and capabilities. Each increment could work on user feedback of how to improve the inventory management or ordering system that is worked on alongside other new features to make the system more functional.

Software Requirements:

Functional:

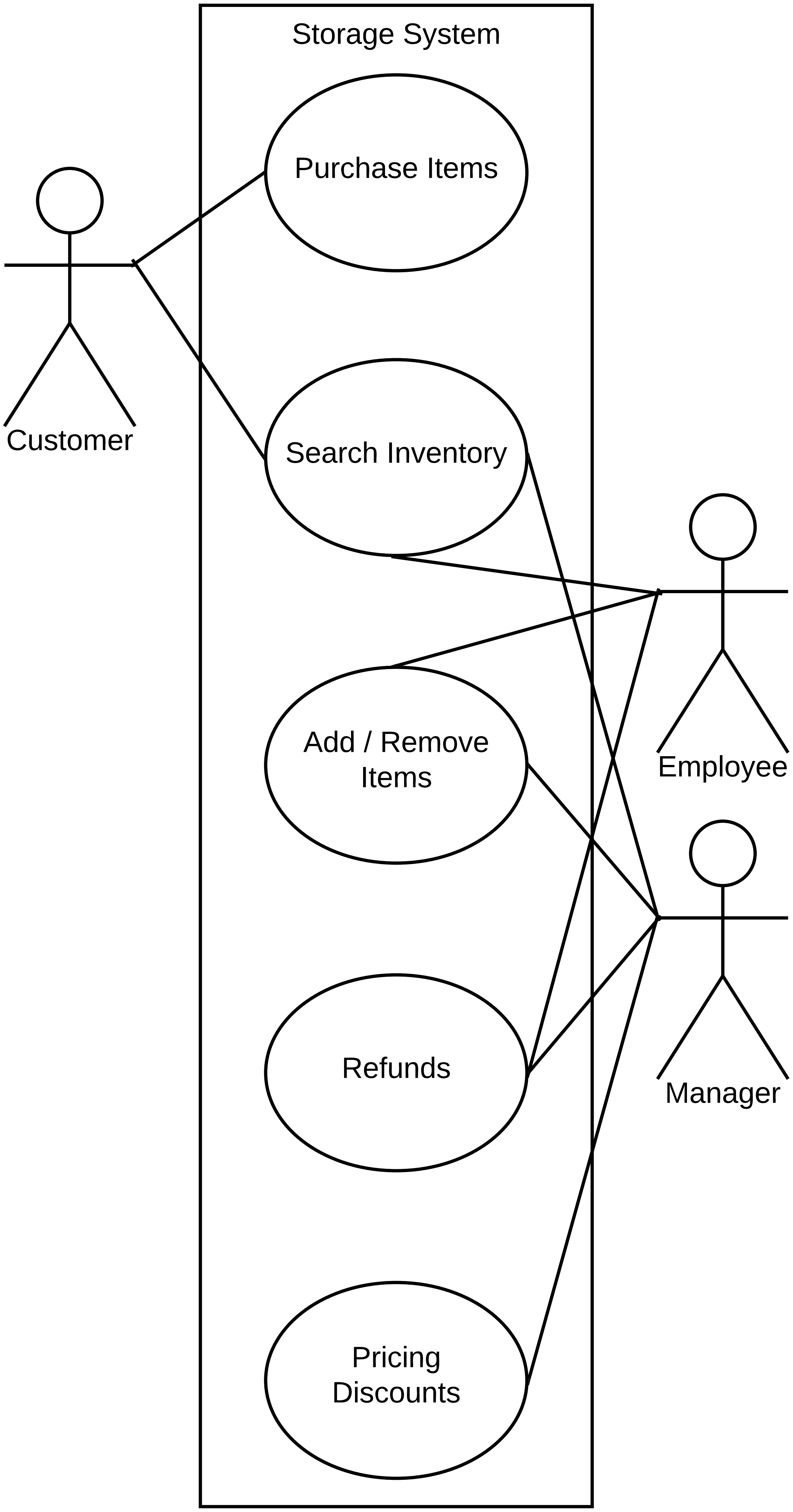
- The employee should be able to order, update and remove items in the inventory.
- The users should be able to purchase, request and refund items in the inventory.
- The users should be able to choose their payment option.
- The users should be able to choose to pick up or deliver.
- The employee should be able to decide which item has discount and have employee discount for themselves.
- The market should be able to show the available items to users and employees.

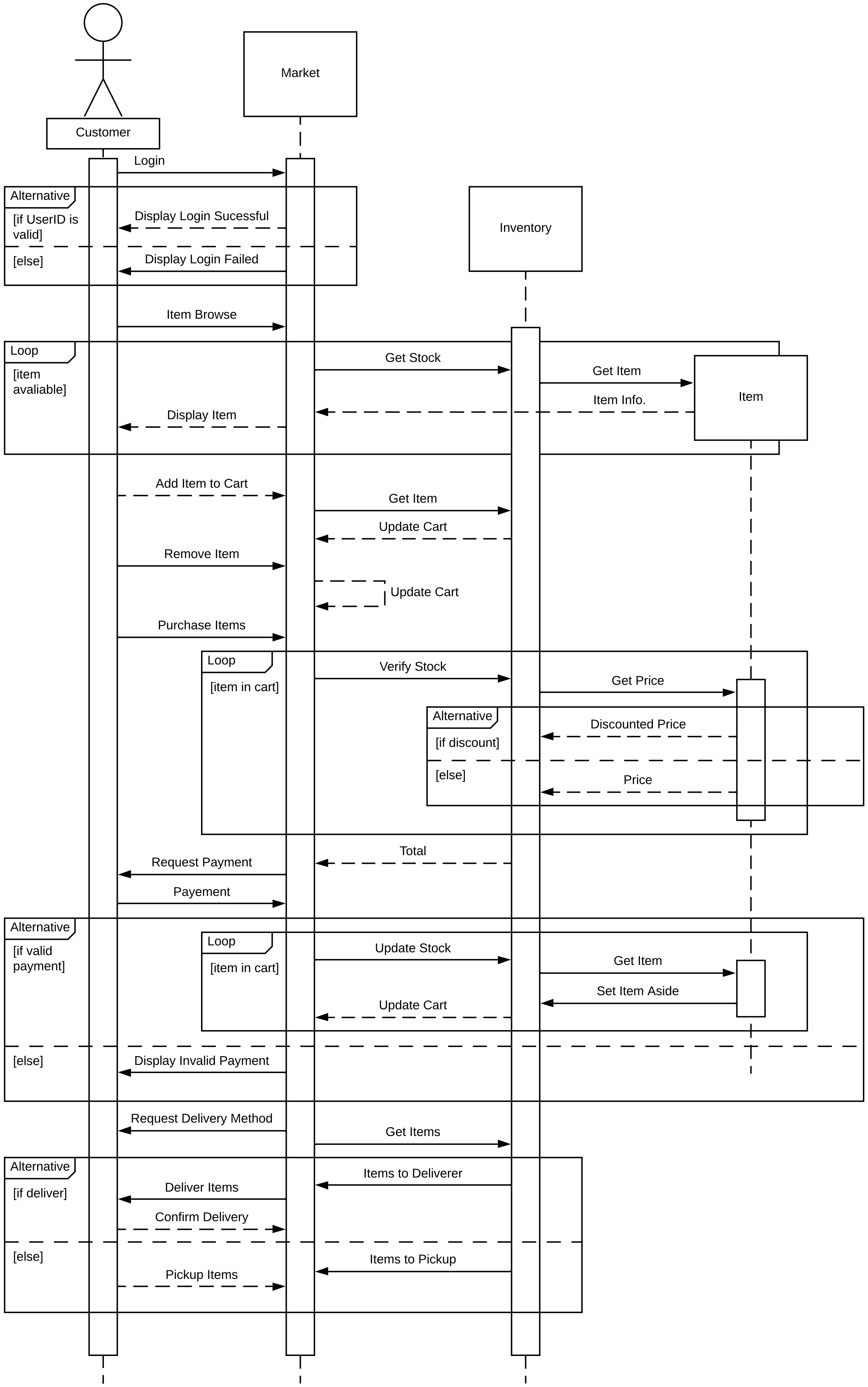
Nonfunctional:

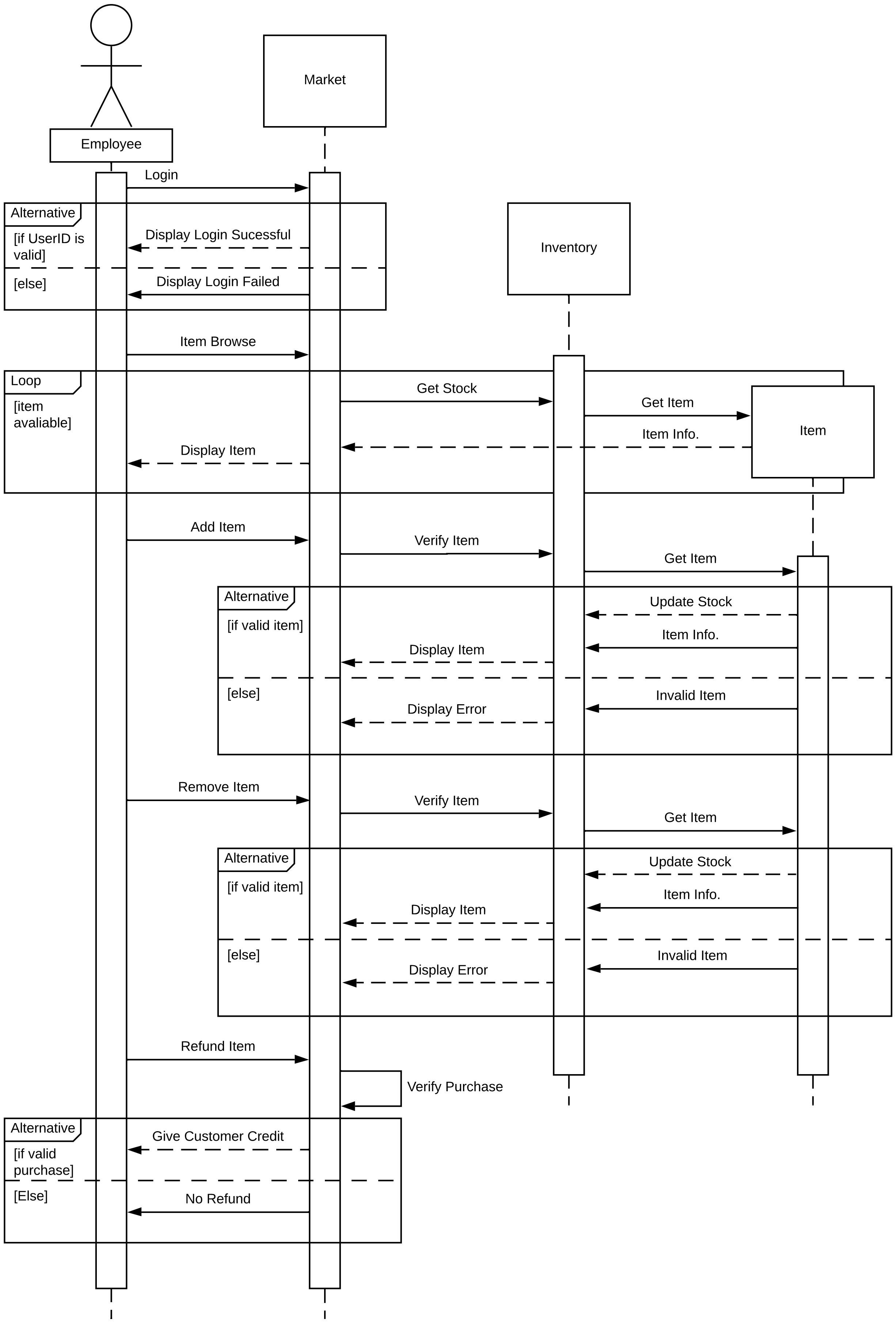
- Be able to use the storage application with response time less than 0.1 seconds.
- Have at least 10+ terabytes of storage space for data storage of inventory.
- Dependable communication between databases should be less than 5 seconds to present accurate/updated information.
- Be able to handle and support over 10,000 users concurrently.
- Systems are available at any time with maintenance frequent, lasting no longer than an hour.
- Process transactions in less than 10 seconds.
- Protect the account info of users.
- Able to connect accounts to banks and send/receive data between them.
- Ease of use with less than 5 days of training to understand the system.
- Employee accounts must be registered with a unique code provided by management.
- System operates on Windows 7, 8, and 10, macs and Linux environments.
- Development of code in C++/C#.
- Must be regulated by admins to prevent any service downtime.
- Have decently large inventories.
- Any downtime/maintenance occurs early mornings from 2-4 AM.

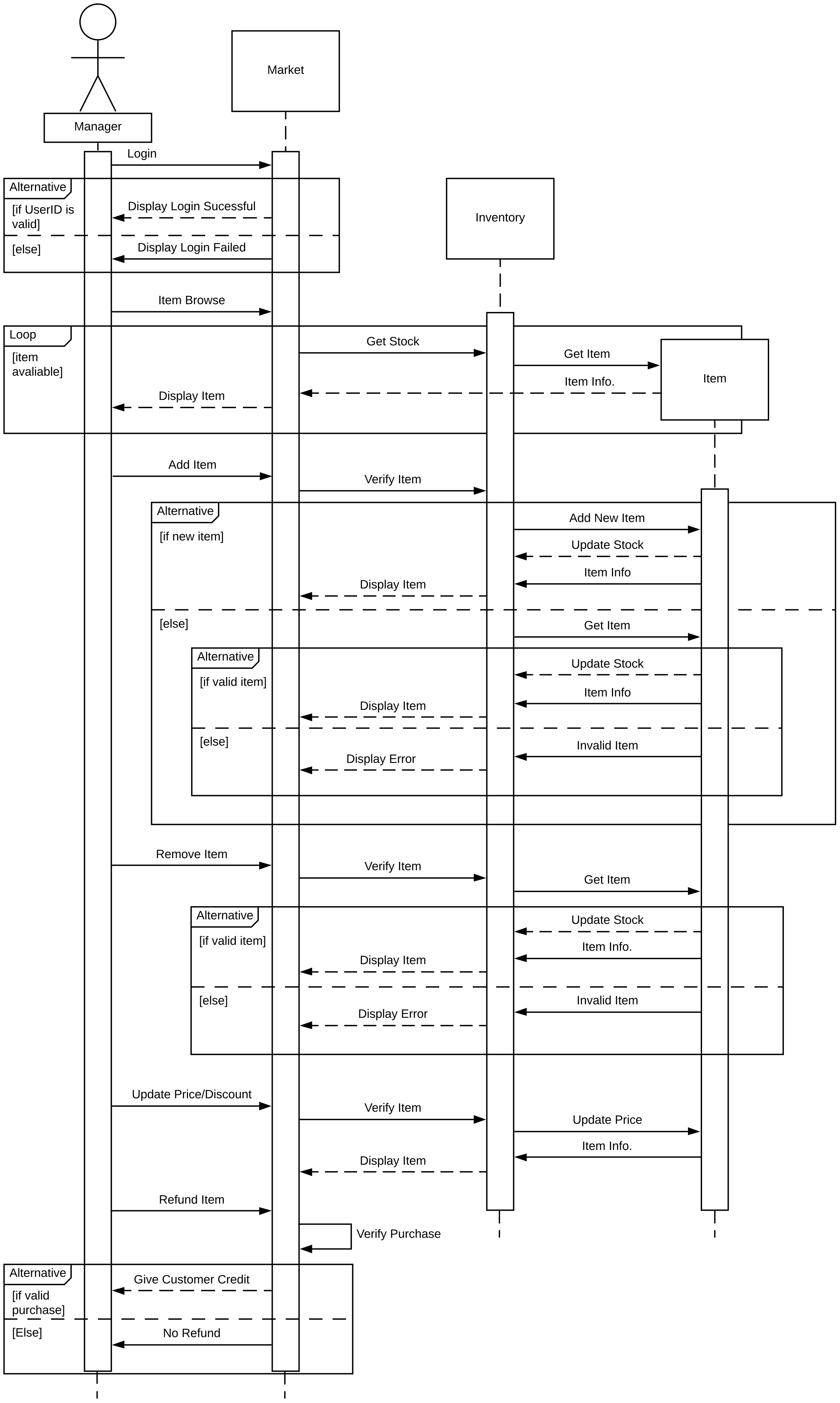
Use Case, Sequence, and Class Diagrams (+ Architectural Design as Client Server):

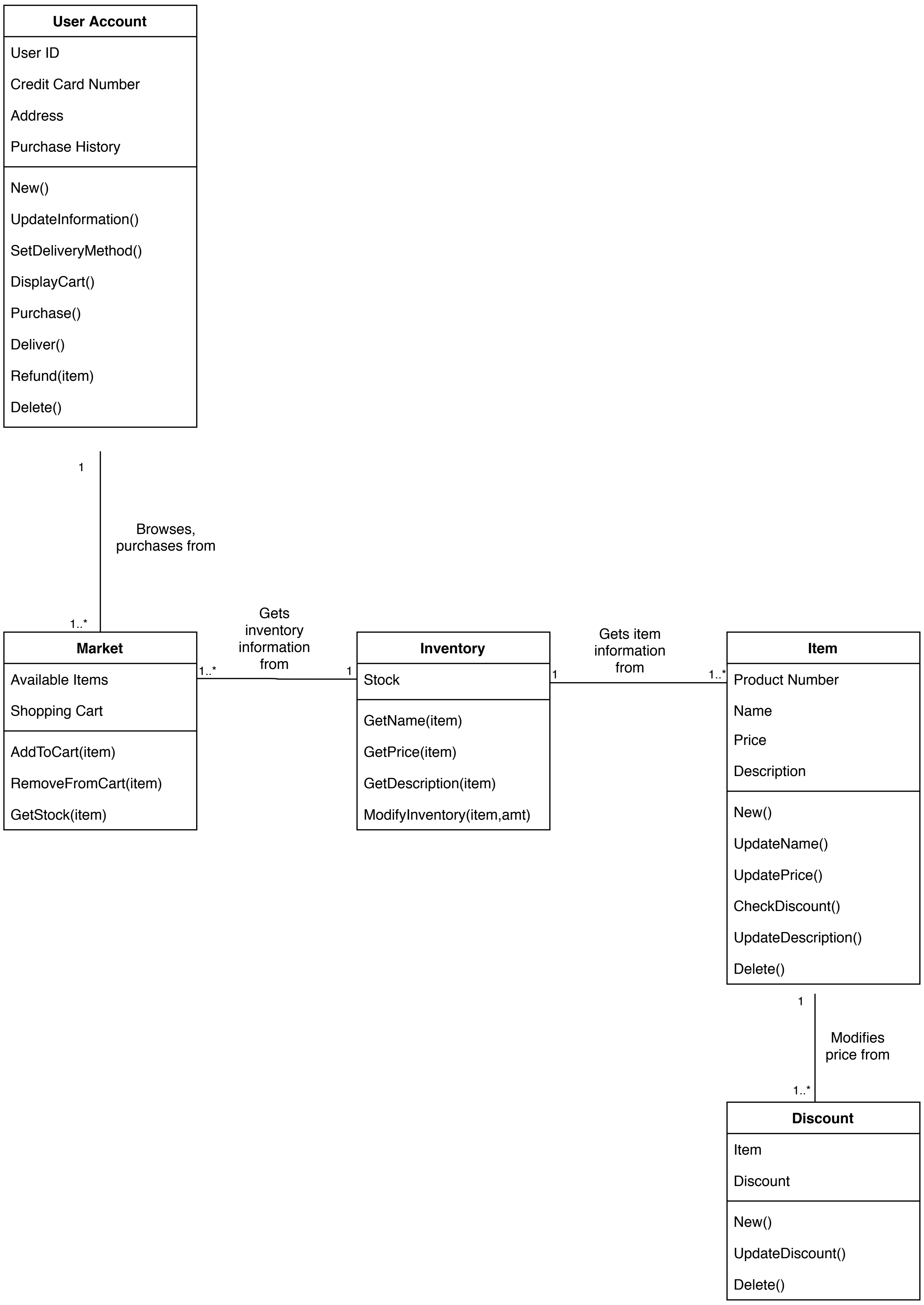
Note: *Each sequence diagram is based on the user type and contains more than one use case per sequence diagram.*

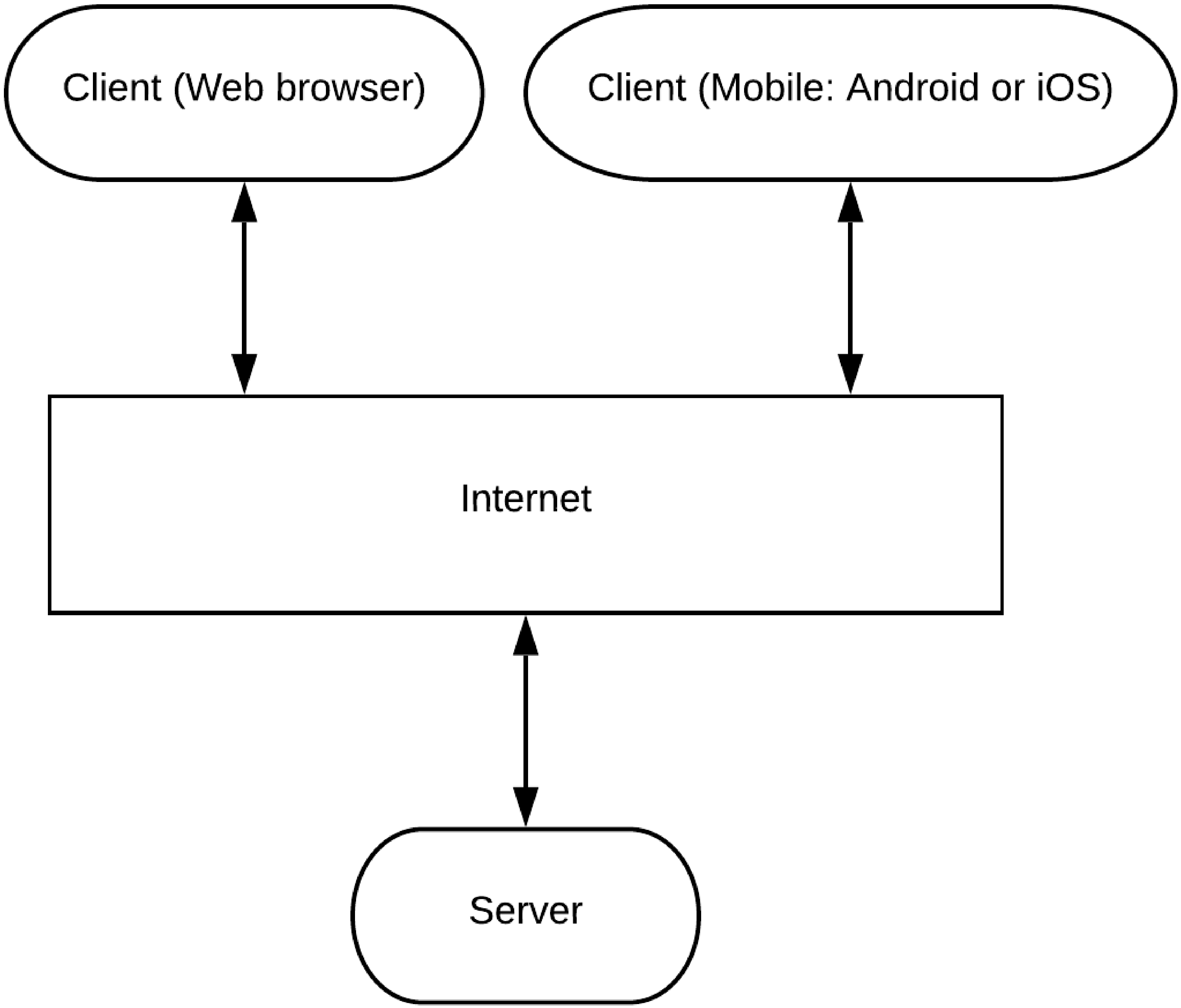

Storage System
Purchase Items
Search Inventory
Add / Remove Items
Refunds
Pricing Discounts
Customer
Employee
Manager


Customer
Market
Inventory
Item
Login
Alternative
[if UserID is valid]
Display Login Sucessful
[else]
Display Login Failed
Item Browse
Loop
[item avaliable]
Get Stock
Get Item
Item Info.
Display Item
Add Item to Cart
Get Item
Update Cart
Remove Item
Update Cart
Purchase Items
Loop
[item in cart]
Verify Stock
Get Price
Alternative
[if discount]
Discounted Price
[else]
Price
Total
Request Payment
Payement
Alternative
[if valid payment]
Loop
[item in cart]
Update Stock
Get Item
Set Item Aside
Update Cart
[else]
Display Invalid Payment
Request Delivery Method
Get Items
Alternative
[if deliver]
Items to Deliverer
Deliver Items
Confirm Delivery
[else]
Items to Pickup
Pickup Items


Employee
Market
Login
Alternative
[if UserID is valid]
Display Login Sucessful
[else]
Display Login Failed
Inventory
Item Browse
Loop
[item avaliable]
Get Stock
Get Item
Item
Item Info.
Display Item
Add Item
Verify Item
Get Item
Alternative
[if valid item]
Update Stock
Item Info.
Display Item
[else]
Invalid Item
Display Error
Remove Item
Verify Item
Get Item
Alternative
[if valid item]
Update Stock
Item Info.
Display Item
[else]
Invalid Item
Display Error
Refund Item
Verify Purchase
Alternative
[if valid purchase]
Give Customer Credit
[Else]
No Refund


Manager
Market
Login
Alternative
[if UserID is valid]
Display Login Sucessful
[else]
Display Login Failed
Inventory
Item Browse
Loop
[item avaliable]
Get Stock
Get Item
Item
Item Info.
Display Item
Add Item
Verify Item
Alternative
[if new item]
Add New Item
Update Stock
Item Info
Display Item
[else]
Get Item
Alternative
[if valid item]
Update Stock
Item Info
Display Item
[else]
Invalid Item
Display Error
Remove Item
Verify Item
Get Item
Alternative
[if valid item]
Update Stock
Item Info.
Display Item
[else]
Invalid Item
Display Error
Update Price/Discount
Verify Item
Update Price
Item Info.
Display Item
Refund Item
Verify Purchase
Alternative
[if valid purchase]
Give Customer Credit
[Else]
No Refund


User Account
User ID
Credit Card Number
Address
Purchase History
New()
UpdateInformation()
SetDeliveryMethod()
DisplayCart()
Purchase()
Deliver()
Refund(item)
Delete()
1
Browses, purchases from
1..*
Market
Available Items
Shopping Cart
AddToCart(item)
RemoveFromCart(item)
GetStock(item)
1..*
Gets inventory information from
1
Inventory
Stock
GetName(item)
GetPrice(item)
GetDescription(item)
ModifyInventory(item,amt)
1
Gets item information from
1..*
Item
Product Number
Name
Price
Description
New()
UpdateName()
UpdatePrice()
CheckDiscount()
UpdateDescription()
Delete()
1
Modifies price from
1..*
Discount
Item
Discount
New()
UpdateDiscount()
Delete()


Client (Web browser)
Client (Mobile: Android or iOS)
Internet
Server

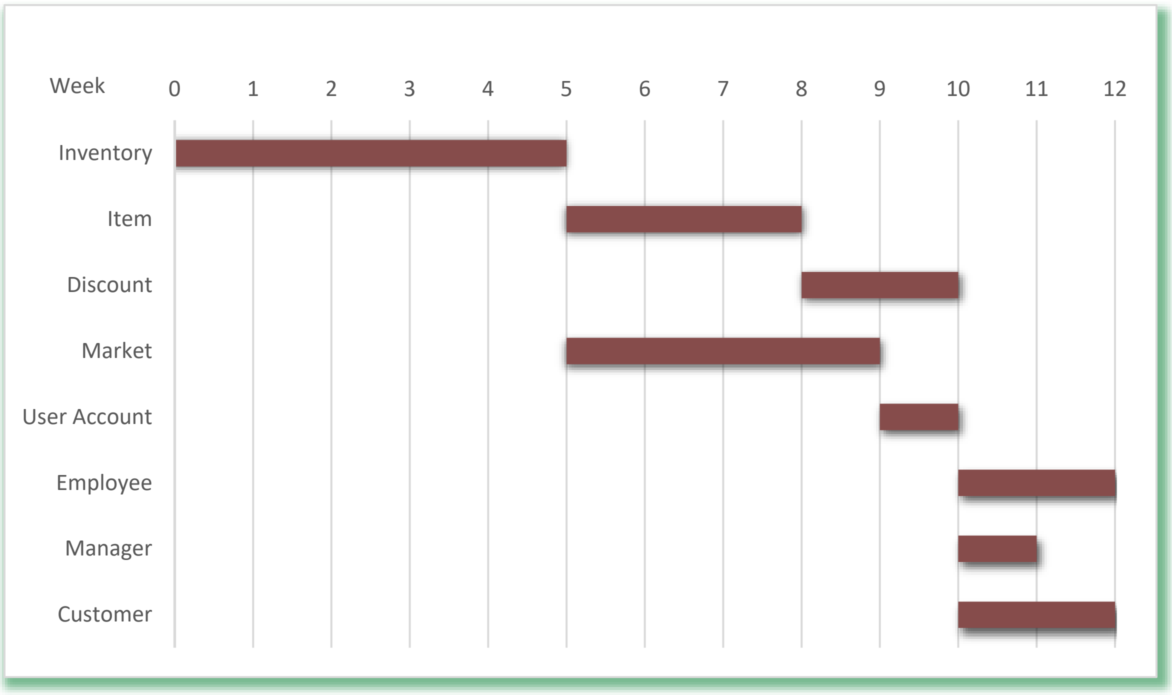
Project Deliverable 1 Content Ends

Page Left Blank on Purpose

Project Deliverable 2 Content Begins

Project Scheduling

Task	Effort (person-days)	Duration	Dependencies
Inventory	75	25	
Item	20	15	Inventory
Discount	5	10	Item
Market	25	20	Inventory
User Account	5	5	Market
Employee	10	10	User Account
Admin	5	5	User Account
Customer	10	10	User Account



Cost, Effort, and Pricing Estimation:

Assume:
All functions have complexity of simple except number of files, which is average.
Team size of 10.
Productivity of 60 function points per person-week.

Function Point Estimation:

Function Category			Complexity			Count x Complexity
		Count	Simple	Average	Complex	
1	Number of User Input	8	3	4	6	24
2	Number of User Output	5	4	5	7	20
3	Number of User Queries	6	3	4	6	18
4	Number of Files and Relational Tables	100	7	10	15	1000
5	Number of External Interface	5	5	7	10	25
					GFP	1087

Processing Complexity:
Moderate (2) - #7, #9
Average (3) - #4, #5, #6, #8, #10, #11, #12, #14
Significant (4) - #2, #3, #13
Essential (5) - #1

GFP = 1087
PCA = 0.65 + 0.01 (5 + 4 x 3 + 3 x 8 + 2 x 2) = 1.1
FP = GFP x PCA = 1087 x 1.1 = 1195.7 FP

Estimating Effort:
E = FP/productivity = 1195.7/60 = 19.9283 = about 20 person-weeks

Project Duration:
D = E/team size = 20/10 = 2 weeks

Project Cost:

Component	Software/Tool	Cost
Front End	Android Studio, IOS	$ 1000
Back End	MongoDB, PG Admin, DataBricks	$ 10000
Cloud Services	AWS	$ 5000
	Total Cost	$ 16000

- This estimation is for the cost of software and training required to build the system. This was based on scalability and reliability.
- These tools were the most reliable because a lot of software developers use them for other projects.
- The most expensive one is AWS because the services are charging per hour or week. Since we need to use these services to prevent a failure, there were not any other cheap alternatives.

Estimated Cost of Hardware Products:

Delivery Fee: Depends on the distance We estimate $50 for gas fee per car per week.
Physical Store Rental Fee: Usually a storefront with 2500 SF space can take about $5500 – $8000 per month.
Storage Space Fee: For a medium size $75000/month.
Machine Cost: The machine to move the heavy stuff we estimate about $10000 in total.

Estimated Cost of Software Products:

Maintenance: $100/month.
Storage System Version Update: $500/month.

Estimate Cost of Personnel:

Assuming we have about 6-8 employees (including one person being delegated to training others), costs can reach high rather quickly. According to Glassdoor the average software developer in the U.S. makes $76,526 a year, however sites such as Fixr and UsNews cite the figure as high as $93,350 and $103,620 respectively. However, Forbes lists the number as low as $72,210 for the salary of a software developer of an application [1, 2, 3, 4].

Given this information we can begin to estimate the cost of the salary of all the employees combined. Now if we ourselves were to actually develop a system of this sort in our current state with our current experience in a small company just starting out, we can be assumed to be achieving a salary on the lower ends of the spectrum; $72,210 was the lowest salary listed on Forbes and if were to multiply this number by 6 and 8, we'd get a value in the range of $433,260 - $577,680. In total, we'd have to pay all employees this much.

However, if we were just to calculate how much it may cost to develop, we may achieve a much different number. We can use the website Thumbtack which goes into detail into how much development may cost for a software per hour. Going off the fact that we'd likely need a combination of not just Front-end and Back-end development but also API, Desktop, and mobile app development we can hope to assume a rather large cost here as well. Thumbtack lists the following as the prices for development [5].

- Basic C development: $75-$150 per hour.
- Front-end web development: $50-$75 per hour.
- Back-end web development: $75-$150 per hour.
- API development: $75-$150 per hour.
- Desktop app development: $30-$100 per hour.
- Mobile app development: $30-$150 per hour.

For the purposes of our app, we can assume the costs to be closer to the lower ends of the presented ranges (as we'd be a smaller development team). We can also expect close to around 100 hours in total for each piece of development, through this, we can achieve a value of around $33,500.

Test Plan:

Identifier: StorageHub_Login.

References: UserCustomerSequenceDiagram.pdf.

Introduction: The objective is to test the login functionality of the 'Storage Hub.'

Test Items: Login function.

Features: Login validation.

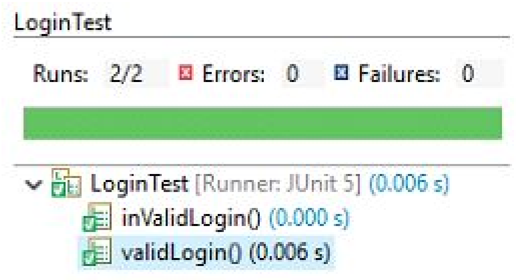
Pass/Fail Criteria: Login function should validate user information. The test should pass if there is an existing user, fail if there is none. The pass percentage should be 100% and there should not be any critical bugs.

Suspension Criteria: The test should stop after it passed or failed.

Test Deliverables: Test case, test code.

Environmental Needs: eclipse IDE, JUnit.

Results:



```
LoginTest
Runs: 2/2    Errors: 0    Failures: 0
LoginTest [Runner: JUnit 5] (0.006 s)
    inValidLogin() (0.000 s)
    validLogin() (0.006 s)
```

Test code included in the zip files.

Comparisons to Other Software:

Our software design mostly resembles that similar to online marketplaces such as Amazon, eBay, and Craigslist where there is some user and buyer interaction via a central website that carries and stores the information and data of the products they have. Most notably, the main similarities that these websites share is that they are able to show users a wide variety of goods that are available and their respective prices from different sellers. StorageHub will likely be similar in concept, but will most likely be a software that is able to be used by many smaller companies to host pricings on their own website rather than being a complete standalone website like the others. Features such as code reusability, UI customization, and detailed inventory management are some that would make our software more unique compared to the other software.

Unlike our software, however, is how Amazon has a bigger focus on the customers rather than the management behind the storage system as it is its own website [6]. Our design is focused on providing for companies that have a subsection for product placements. eBay, on the other hand, shares many features but does not have a place for warehouse directories as their sellers are direct, whereas our implementation of StorageHub, Amazon, and Craigslist have direct or host warehouses [7, 8]. Similarly, Craigslist is mainly based upon a unique buyer/seller interaction like in eBay, while in Amazon and StorageHub there are more brands or companies that sell to the consumers. Design-wise, StorageHub's user interface would be similar to Craigslist's, simple and straightforward [9]. Amazon and eBay have more stylized UIs, such to distinguish themselves from others, but StorageHub is meant for multiple different companies to be able to utilize for their own endeavours [6, 10].

	StorageHub	Amazon	eBay	Craigslist
Standalone Website	x	o	o	o
Mass Company Reusability	o	x	x	x
Implements Warehouses	o	o	x	o
Simple UI	o	x	x	o
Unique Buyer/Seller Oriented	x	x	o	o
Company/Brand Interaction	o	o	x	x
Online Marketplace	o	o	o	o

Conclusion:

The project largely went according to plan. As a whole, we didn't really have to go back and make many changes. This is largely due to our flexible project plan; we didn't lock anything down until we had to, so we rarely had to change anything that we had explicitly decided on. Admittedly, part of this was almost certainly due to the nature of the project, in that we were only pretending to design a software. If we had actually been creating it, we likely would have run into problems that needed to be solved and thus would have had to revise our approach on some occasions, but as it stands, we were working largely in the hypothetical, so we never found out about any such obstacles. All in all, our planning was fairly effective.

References:

[1] "Salary: Software Developer," Glassdoor. [Online]. Available: https://www.glassdoor.com/Salaries/software-developer-salary-SRCH_KO0,18.htm. [Accessed: 19-Apr-2020].

[2] DePietro, "Here's How Much Money Software Developers Earn In Every State," Forbes, 21-May-2019. [Online]. Available: https://www.forbes.com/sites/andrewdepietro/2019/05/21/software-developer-salary-state/#3d548a8c2f2c. [Accessed: 19-Apr-2020].

[3] "How Much Can a Software Developer Expect to Get Paid?," U.S. News & World Report. [Online]. Available: https://money.usnews.com/careers/best-jobs/software-developer/salary. [Accessed: 19-Apr-2020].

[4] Y. Kato, "How Much Does a Software Developer Make in Every State?," Fixr Blog, 03-Dec-2019. [Online]. Available: https://www.fixr.com/blog/2015/02/20/how-much-does-a-sofware-developer-make-us/. [Accessed: 19-Apr-2020].

[5] T. Editors, "How much does software development cost?," Thumbtack, 26-Sep-2017. [Online]. Available: https://www.thumbtack.com/p/software-development-costs. [Accessed: 19-Apr-2020].

[6] Amazon.com. "Help: Site Features" Amazon. [Online]. Available: https://www.amazon.com/gp/help/customer/display.html?nodeId=10197041. [Accessed

Apr. 18, 2020].

[7] The eBay Community. “Is there such a thing as eBay warehouse?” [Online]. Available: https://community.ebay.com/t5/Archive-Shipping-Returns/warehouse/td-p/20208258. [Accessed Apr. 18, 2020].

[8] Amazon.com. “Amazon Warehouse: FAQs.” Amazon. [Online]. Available: https://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000656811. [Accessed Apr.
18, 2020].

[9] Craigslist Homepage. Craigslist. [Online]. Available: https://www.craigslist.org/. [Accessed Apr. 19, 2020].

[10] “Customer Service.” eBay. [Online]. Available: https://www.ebay.com/help/home. [Accessed Apr. 19, 2020].