DBAY: DREXEL VERSION OF EBAY SYSTEM

ANALYSIS AND DESIGN (A&D) TERM PROJECT

INFO 620

WINTER 2018 GROUP 5

Xiang Liu, Xujian Zhang, Wenyu He

03/17/2018

Acknowledgement

We would like to appreciate this excellent course from Dr. Kinkela. You are so helpful and successfully gave us a wonderful experience in learning UML and its application in system design. Thank you very much. Wish we could have more cooperation in the future.

Contents

1 The Problem Statements

1.1 Context and importance of the system

It is important to develop the DBay system because there are a great number of used or new books, laptops, ipad, etc. from people in Drexel community and these things will no longer be used but potentially useful to some other people. Trading these items within the Drexel community would be beneficial to both the sellers and buyers. By using DBay, for example, people can list items for sale or buy items sold by other people.

1.2 Goal of the system

The overall goal of the system is to provide a platform for all Drexel members to buy and sell used or new items.

1.3 Scope of the project

- IN-Scope:

DBay will include only the users, used or new items selling or buying, feedback, cart, payments.

- OUT-Scope:

DBay will not include detecting users buying habit, transporting items, or being responsible for product quality.

2 Requirements

2.1 Functional requirements

- Allow user to create an account and manage its profile.
- Verify user by Drexel email address and password.
- Provide a platform to buy and sell items or services
  - User can be both buyer and seller but can only have one role in a single transaction.
  - Sellers can post items or services through a user-friendly submission page.
  - Sellers can manage selling items/services within his/her account.
- Offer a search function to find an item or service.
- Provide payment methods including bank checking account, credit card, debit card, and Dragon Dollars.
- Backup the system and database routinely.
- Allow buyers to rate the item or service they have bought.
- Allow users to report inappropriate items or services.

2.2 Data requirements of the system

For user, keep track of user's name, address, phone number, purchase history and items in cart. The purchase history will keep records until the user has graduated from Drexel. Items in cart won't disappear unless the items are unavailable (deleted or sold out).

For each item, Dbay will keep track the item price, date added, description, seller contract, status. Item status contains the number remains/sold out/modifying. Every seller must provide an available Email address or phone number.

For each payment, Dbay will keep track of payment number, date, description, amount, type, account number and bank name.

For feedback, Dbay will keep track feedback content, item name, buying date and rate for the item.

2.3 Business rules and logic

- This service will be free for all users in Drexel community, including students, faculty, staff, etc.
- DBay members must provide a Drexel University email address and verify the email (reply-to email) when registering.
- DBay members must fill out a required form to sign up for an account and/or post items/service advertisements.
- DBay members can be buyer, seller or both in a certain time period.
- DBay members can upload up to 5 photos per advertisement. The longest side of the photo much have at least 500 pixels and the size of the photo cannot exceed 5 MB.
- Each member can have up to 25 advertisements running at the same time.
- Advertisement expire after 2 weeks from the creating date/time; The user can reactivate the advertisement within 30 days after the expiration date.
- The seller can post either a minimum bid price, buy-it-now price, or both for each advertisement. Buy-it-now price must be higher than the minimum bid price. Confirmed bids must meet the minimum.
- Buyer can rate the seller within 30 days after a transaction is made. The ratings range from 1 to 5 stars.
- If a member posts inappropriate content or comments, the member's account will be reviewed according to DBay policies and, if checked, blocked.

2.4 Non-functional requirements

- The system will be a screen-based application (Usability).
- Advertisements will have the option of being sorted according to price, type and create time via drop-down menus (Usability).
- Accounts are password-protected (Security).
- System is password-protected (Security).
- System will be backed up on a daily basis (Back up).
- System will be operational 24/7, with exception for daily back up and weekly maintenance (Operation).
- System administrators have no legal liability regarding advertisements and transactions (Legal).
- System will offer mobile app for iOS and Android devices (Usability).

2.5 Other assumptions

- We will assume DBay will be used by the Drexel community, primarily students.
- DBay will be web-based and accessible to users by common browsers regardless of operating systems.
- Hardware requirements will be further investigated.
- Server OS depends on Drexel preference, but will need to accommodate common web application solutions.
- The daily back up of the system is scheduled and operated automatically.
- DBay needs a small group of staff for developing and maintenance.
- Development of the system will be iterative, focusing on application for traditional web display at first and then move into mobile display and, a separate app if possible.

3 Use Case Model

3.1 Actors and Their Goals

- Buyer: Buyer can create account and update account profile. Buyer can buy item/services by creating order or updating order details, and buyer can also participate in bidding items. Buyer can also request cancellation of order or return of order. Buyer can rate an order after an order is finished.
- Seller: Seller can create account and update account profile. Seller can manage inventory with items he/she is selling.
- Employ: Employee is responsible for verifying the payment methods submitted by the buyer. Employee is also responsible for processing return and refund. Employee can block users if the users make violations.

3.2 Use Case Diagram

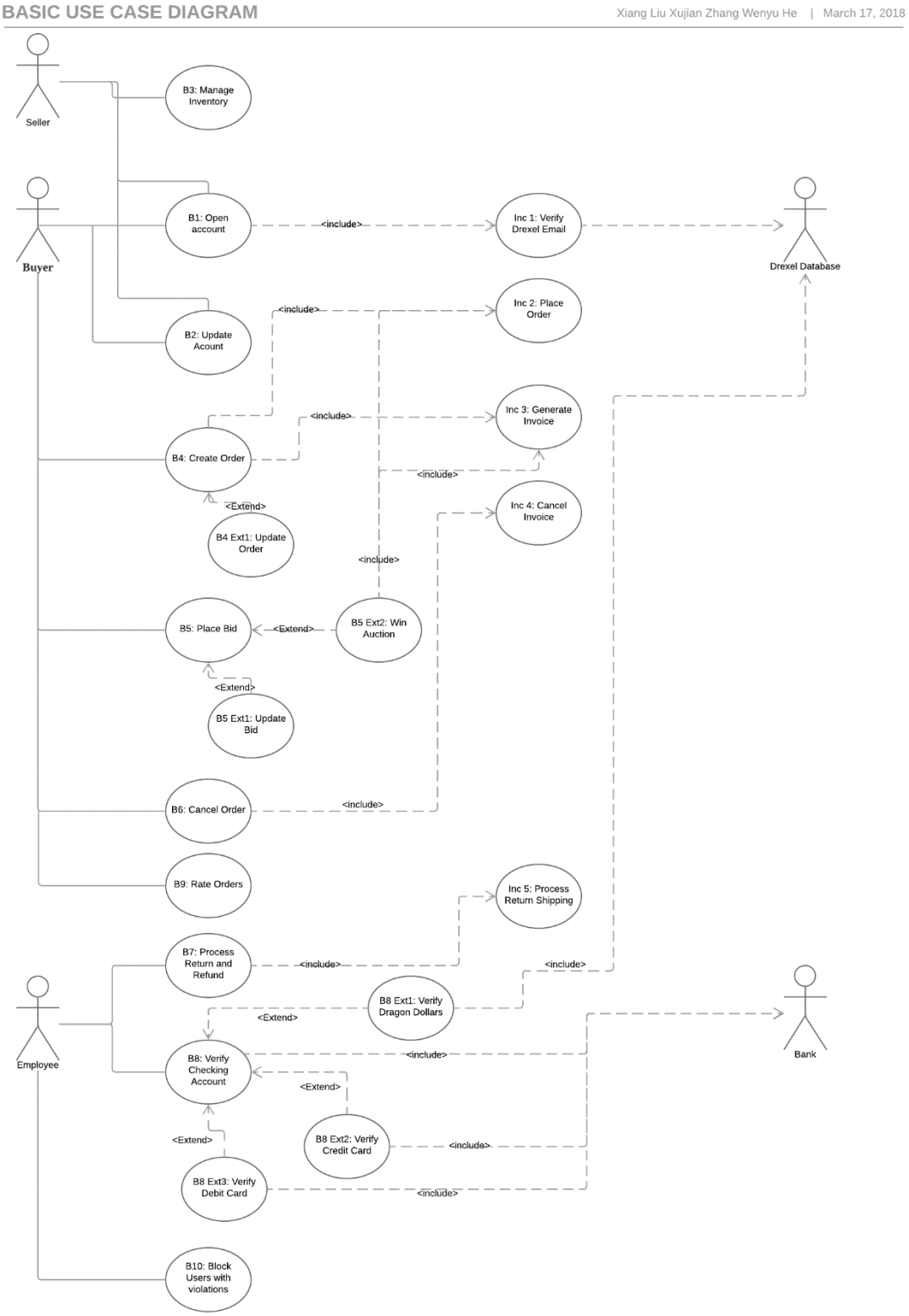


Figure 1 Use Case Diagram for DBay system

3.3 Overview of Use Cases

1) Case B1 Open Account: People in Drexel community can join DBay by opening an account in the sign-up page of the DBay website. He will be asked to provide a username, a password, valid Drexel email address, first name, last name, phone number, address. After submitting those information, a verify email will be send to the email address provided.
2) Included Case Inc 1 Verify Drexel Email: System will check the email address provided by the user as well as his/her first name and last name with the record in Drexel Database. Once information matches, the registering process will be continued. Otherwise, the registration will be rejected.
3) Case B2 Update Account: User can update his/her profile including phone, address, etc. in the user account page after the user has successfully logged into the DBay website with a valid username and password.
4) Case B3 Manage Inventory: Seller can manage the inventory for items/services being sold by him/her in the user account page after the buyer has successfully logged into the DBay website with a valid username and password.
5) Case B4 Create Order: Buyer can buy the items in his/her shopping cart by creating an order. In the page of shopping cart, the buyer clicks on the submit order button, then an order review page will be prompted. After reviewing the items and choosing the payment method, the buyer can click the “Submit Order” button, then the order will be created.
6) Included Case Inc 2 Place Order: After the buyer submits the order, a transaction will be initiated by the system through the payment method the buyer has chosen. When the transaction has been successfully completed, the order will be placed, and a confirmation page will be prompted to the buyer.
7) Included Case Inc 3 Generate Invoice: After the seller has shipped the items to the buyer and provided a tracking information to the system, an invoice will be generated and send to the buyer.
8) Extended Case B4 Ext1 Update Order: After the order has been successfully placed, the buyer can update the order information such as shipping address, etc., within a certain period (24 hours).
9) Case B5 Place Bid: Buyer can submit a bidding price for items on bid by entering a price number on the item page and clicking the “Submit Bid” button. Then the bidding price will be recorded.
10) Extended Case B5 Ext1 Update Bid: Before the finishing time of an auction, the buyer who has already bid for an item and submit a new bid price again. This new bid price must be higher than current highest bid price.
11) Extended Case B5 Ext2 Win Auction: When an auction has finished, the system will check the bid record with the highest bid price. Buyer with highest bid price will win the auction and the order will be placed automatically. Then a confirmation message will be send to the buyer.

12) Case B6 Cancel Order: Within 30 minutes after the order has been placed, the buyer can cancel the order by logging into the order history page, choosing the order record, and clicking on the “Cancel Order” button.
13) Included Case Inc 4 Cancel Invoice: After an order has been successfully canceled, a cancel invoice will be generated and send to the buyer.
14) Case B7 Process Return and Refund: The employee is responsible for processing the return and refund process initiated by the buyer. When the employee logged in the employee page, a list of return request will be prompted. He can click on one request and review the request, after that the employee can click the “Issue” button to issue the return or “reject” button to reject the request.
15) Included Case 5 Process Return Shipping: Once the return has been issued, the employee will ask the buyer to ship the items back to the seller. Once the buyer provided shipping confirmation, the employee will generate a return invoice.
16) Case B8 Verify Checking Account: User can submit checking account as his payment method into the system. Employee is responsible for verifying the checking account with the bank systems. Once the account is verified, the account information will be stored within the user account and can be used in the future. So, the user does not need to enter account every time he wants to use it.
17) Extended Case B8 Ext1 Verify Dragon Dollars: User can submit Dragon Dollars account as his payment method into the system. Employee is responsible for verifying the account with the Drexel database. Once the account is verified, the account information will be stored within the user account and can be used in the future. So, the user does not need to enter account every time he wants to use it.
18) Extended Case B8 Ext2 Verify Credit Card: User can submit credit card as his payment method into the system. Employee is responsible for verifying the credit card with the card-issuing institution. Once the account is verified, the account information will be stored within the user account and can be used in the future. So, the user does not need to enter account every time he wants to use it.
19) Extended Case B8 Ext2 Verify Debit Card: User can submit debit card as his payment method into the system. Employee is responsible for verifying the credit card with the card-issuing institution. Once the account is verified, the account information will be stored within the user account and can be used in the future. So, the user does not need to enter account every time he wants to use it.
20) Case B9 Rate Orders: After an order has been successfully completed, the buyer can rate the items or services according to his/her experience and product qualities. He can log into his user account page, enter the order history page, there is a “Rate” button for each order. The buyer can click on the button and submit the rating and comments.
21) Case B10 Block Users with Violations: When users violate the DBay policies, the employee can block those users. The employee can log into the administration page and click on the “Block User” button. Then the employee can enter the username and reason for blocking. After submitting, the user with the associated Drexel email address will be blocked from the DBay system.

3.4 Use Case Description

3.4.1 Base Case B5 From Xiang Liu

USE CASE #	BASE 5	
USE CASE Name	Place Bid	
ACTOR	Buyer	
Goal (1 phrase)	To place a bid for items sold by auction.	
Overview and scope	Buyer places bid for the item being sold by auction. The system will record his bidding price. After the auction period is finished, the buyer with the highest bidding price will win the auction, and the system will place the order for him.	
Level	Primary	
Preconditions	User enters the webpage for the item being sold by auction.	
Postconditions in words (write in passive and past tense)	Bidding record biddingRecord was created. Attributes biddingRecord_buyID, biddingRecord_itemID, biddingRecord_time, biddingRecord_price were included.	
Trigger	Event is triggered when user click “bid” on item for auction page	
Included Use Cases		
Extending Use Cases	CASE B5 Ext1,CASE B5 Ex2	
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference “included use cases” in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
	Step 1. User enters the item page.	Step 2. System displays the item page.
	Step 3. User clicks on “bid” button	Step 4. System prompts for bidding price.
	Step 5. User enter the bidding price.	Step 6. The system prompts for bidding confirmation.
	Step 7. User clicked on “Confirm” button.	Step 7. The system records this bidding order.

OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension points using EXTEND *eus_name*)	Step 1. User enters the item page.	Step 2. System displays the item page.
	Step 3. User clicks on “bid” button	Step 4. System prompts for bidding price.
	Step 5. User enters the bidding price.	Step 6. System prompts for bidding confirmation.
	Step 6. User clicks the “Edit” button.	Step 7. System displays the review page.
	Step 8. User edits the bidding price and clicks on the “Submit” button.	Step 6. System prompts for bidding confirmation.
	Step 7. User clicked on “Confirm” button.	Step 7. System records this bidding order.
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	Step 1. User enters the item page.	Step 2. System displays the item page.
	Step 3. User clicks on “bid” button	Step 4. System prompts for bidding price.
	Step 5. User enters the bidding price.	Step 6. System detects the bidding price is lower than the current price.
		Step 7. System displays the warning page.
		Step 8. System terminates the bidding process.
Priority in scheduling	Primary	
Frequency	Once per iteration.	
Business rules and data logic	Only valid DBay user can place bid for items in the DBay. When the auction is finished, the buyer with highest bidding price will win the auction. Once a buyer win the auction, system will make the order automatically.	

Other non-functional requirements	Windows standard user interface Database backed up daily.
Superordinates	
Developer	Xiang Liu
Creation date and last modified date	03/17/2018
Other Comments	

USE CASE #	B5 Ext1	
USE CASE Name	Update Bid	
ACTOR	Buyer	
Goal (1 phrase)	To update the bidding price for a existing bid order.	
Overview and scope	The buyer can update his bidding price after he has made a bid for an item. The system will record this new price instead of the old one.	
Level	Extended.	
Preconditions	User has already made a bid for an item, and enters the webpage for that item.	
Postconditions in words (write in passive and past tense)	Bidding record biddingRecord was updated. Attributes biddingRecord_buyID, biddingRecord_itemID, biddingRecord_time, biddingRecord_price were updated.	
Trigger	This event is triggered when user enters the item page and clicks the “Update Bid” button.	
Included Use Cases		
Extending Use Cases		
MAIN	**Actor Action**	System Action

***SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference "included use cases" in this section using INCLUDE *ius_name*	Step 3.1. User enters the item page.	Step 3.2. System displays the item page.
	Step 3.3. User clicks on "Update Bid" button	Step 3.4. System prompts for new bidding price.
	Step 3.5. User enters a new bidding price.	Step 3.6. The system prompts for bidding confirmation.
	Step 3.7. User clicks the "Edit" button.	Step 3.8. The system records this bidding order.
UNSUCCESSFUL SCENARIOS (*erroneous* situations)	**Conditions**	**Actions**
	Step 3.1. User enters the item page.	Step 3.2. System displays the item page.
	Step 3.3. User clicks on "Update Bid" button	Step 3.4. System prompts for new bidding price.
	Step 3.5. User enters a new bidding price.	Step 3.6. System detects the bidding price is lower than the current price.
		Step 3.7. System displays the warning page.
		Step 3.8. System terminates the updating process.
Priority in scheduling	Primary	
Frequency	Repeatable.	
Business rules and data logic	Update Bid process can only be initiated once the buyer has already placed a bid for an item. The new bidding price must be higher than the current highest price.	
Other non-functional requirements	Windows standard user interface	
Superordinates	BASE 5	
Developer	Xiang Liu	

Creation date and last modified date	03/17/2018
Other Comments	

USE CASE #	B5 Ext2	
USE CASE Name	Win auction	
ACTOR		
Goal (1 phrase)	To place the order for the buyer with highest bidding price.	
Overview and scope	When the auction period is finished, the buyer with the highest bidding price win the auction automatically. The system will make the order for him.	
Level	Extended.	
Preconditions	The auction end time is reached, the auction period is finished.	
Postconditions in words (write in passive and past tense)	An auction order auction_Order was created. Attribute auction_Buyer, auction_Seller, auction_Price, auction_item were included. The use case Inc 2 and Inc 3 were initiated.	
Trigger	This event is triggered when the aution end time is reached.	
Included Use Cases	Inc 2, Inc 3	
Extending Use Cases		
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence**	**Actor Action**	System Action
		Step 3.10. System places the auction order for the buyer with highest bidding price.
Reference “included use cases” in this		Step 3.11 Included use case Inc 2 is initiated and a trading order is placed.

section using INCLUDE *ius_name*		Step 3.12 Included use case Inc 3 is initiated and an invoice is places.
	Conditions	**Actions**
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*		Step 3.10. System detects the bidding price from a user is not the highest price.
		Step 3.11 System ignore the bidding record from that user.
		Step 3.12 The bidding record is removed.
	Primary	
	Repeatable.	
Priority in scheduling	Primary	
Frequency	Once per run.	
Business rules and data logic	Once the auction is finished, the buyer with the highest bidding price win the auction automatically.	
Other non-functional requirements	Windows standard user interface	
Superordinates	BASE 5	
Developer	Xiang Liu	
Creation date and last modified date	03/17/2018	
Other Comments		

3.4.2 Base Case B8 From Xujian Zhang

USE CASE #	B8	
USE CASE Name	Verify Bank Account	
ACTOR	Employee, bank	
Goal (1 phrase)	To verify payment method to make sure the payment is valid or not.	
Overview and scope	Before a buyer want to buy something, he needs to create a payment method by using credit card, debit card or dragon card. Buyer should provide all information of his payment. Then employee could verify the payment method.	
Level	Base	
Preconditions	Buyer log in the Dbay system	
Postconditions in words (write in passive and past tense)	A buyer object currentbuyer was created. A payment method object PaymentMethod was created. Attributes PaymentMethod.BuyerID, PaymentMethod.AccountType, PaymentMethod.Account#, PaymentMethod.SecureCode updated.	
Trigger	This event is triggered when buyer places an order	
Included Use Cases	None	
Extending Use Cases	B8 Ext1, B8 Ext2, B8 Ext3	
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference "included use cases" in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
	Step 1. Buyer enters account screen	Step 2. Systems presents account manage page
	Step 3. Buyer enters create payment screen	Step 4. System provides a form for collecting all information for a payment method
	Step 5. Buyer inputs information of his payment method	Step 6. System sends request to bank or Drexel database depends on account type

		Step 7. System sends confirmation to buyer
OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension points using EXTEND *eus_name*)	Step 1. Buyer enters account screen	Step 2. Systems presents account manage page
	Step 3. Buyer enters create payment screen	Step 4. System provides a form for collecting all information for a payment method
	Step 5. Buyer inputs information of his payment method	Step 6. **Extend B6 Ext1** System sends request to bank
		Step 7. System sends confirmation to buyer
OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension points using EXTEND *eus_name*)	Step 1. Buyer enters account screen	Step 2. Systems presents account manage page
	Step 3. Buyer enters create payment screen	Step 4. System provides a form for collecting all information for a payment method
	Step 5. Buyer inputs information of his payment method	Step 6. **Extend B6 Ext2** System sends request to bank
		Step 7. System sends confirmation to buyer
OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension	Step 1. Buyer enters account screen	Step 2. Systems presents account manage page
	Step 3. Buyer enters create payment screen	Step 4. System provides a form for collecting all information for a payment method
	Step 5. Buyer inputs information of his payment method	Step 6. **Extend B6 Ext3** System sends request to Drexel database

points using EXTEND *eus_name*)		Step 7. System sends confirmation to buyer
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	Step 1. Buyer enters account screen	Step 2. Systems presents account manage page
	Step 3. Buyer enters create payment screen	Step 4. System provides a form for collecting all information for a payment method
	Step 5. Buyer inputs information of his payment method	Step 8. Credit card information is wrong
		Step 9. System sends alert to buyer
Priority in scheduling	High	
Frequency	Once create payment method	
Business rules and data logic	There are three ways to make a payment: dragon dollars, credit card and debit card. Instead of the three ways, other ways are illegal and should be dropped.	
Other non-functional requirements	Windows standard user interface Drexel database backup	
Superordinates		
Developer	Xujian Zhang	
Creation date and last modified date	03/17/2018	
Other Comments		

USE CASE #	B8 Ex3
USE CASE Name	Verify debit card
ACTOR	Bank

Goal (1 phrase)	To verify a debit card could be used to pay or not	
Overview and scope	After system send request to bank, bank need to verify if the card is valid or not.	
Level	Extend	
Preconditions	System send request to bank	
Postconditions in words (write in passive and past tense)	A payment method object PaymentMethod was created. Attributes PaymentMethod.BuyerID, PaymentMethod.AccountType, PaymentMethod.Account#, PaymentMethod.SecureCode updated.	
Trigger	This event is triggered when system send a debit card request to bank.	
Included Use Cases	None	
Extending Use Cases	None	
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference "included use cases" in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
	Step 1. Buyer inputs information of his debit cards.	Step 2. System sends request to Bank
		Step 3. Bank verifies debit card information
		Step 4. System sends confirmation to buyer.
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	**Conditions**	**Actions**
	Step 1. Buyer inputs information of his debit cards.	Step 2. System sends request to bank.
		Step 3. Bank can't verify debit card information

		Step 4. System sends alert to buyer.
Priority in scheduling	High	
Frequency	Once create payment method	
Business rules and data logic	If there is anything wrong about the information of the debit card, bank should abort the transaction and system sends alert back.	
Other non-functional requirements	Bank connect Internet	
Superordinates	B8 Verify Bank Account	
Developer	Xujian Zhang	
Creation date and last modified date	03/17/2018	
Other Comments		

USE CASE #	B8 Ex2
USE CASE Name	Verify credit card
ACTOR	Bank
Goal (1 phrase)	To verify a credit card could be used to pay or not
Overview and scope	After system send request to bank, bank need to verify if the card is valid or not.
Level	Extend
Preconditions	System send request to bank
Postconditions in words (write in	A payment method object PaymentMethod was created. Attributes PaymentMethod.BuyerID,

passive and past tense)	PaymentMethod.AccountType, PaymentMethod.Account#, PaymentMethod.SecureCode updated.	
Trigger	This event is triggered when system send a credit card request to bank.	
Included Use Cases	None	
Extending Use Cases	None	
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference "included use cases" in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
	Step 1. Buyer inputs information of his credit cards.	Step 2. System sends request to bank.
		Step 4. Bank verifies credit card information
		Step 5. System sends confirmation to buyer.
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	**Conditions**	**Actions**
	Step 1. Buyer inputs information of his credit cards.	Step 2. System sends request to bank.
		Step 3. Bank can't verify debit card information
		Step 4. System sends alert to buyer.
Priority in scheduling	High	
Frequency	Once create payment method	
Business rules and data logic	If there is anything wrong about the information of the credit card, bank should abort the transaction and send fail information back.	
Other non-functional requirements	Bank connect Internet	

Superordinates	B8 Verify Bank Account
Developer	Xujian Zhang
Creation date and last modified date	03/172018
Other Comments	

USE CASE #	B8 Ex1	
USE CASE Name	Process Dragon Dollars	
ACTOR	Drexel database	
Goal (1 phrase)	To verify a Dexel dragon dollars account could be used to pay or not	
Overview and scope	After system send request to Drexel database, Drexel database need to verify if the account is valid or not.	
Level	Extend	
Preconditions	System send request to Drexel database	
Postconditions in words (write in passive and past tense)	A payment method object PaymentMethod was created. Attributes PaymentMethod.DrexelId, PaymentMethod.DrexelPw, PaymentMethod.userFirstName, PaymentMethod.userLastName updated.	
Trigger	This event is triggered when system send a dragon dollars request to Drexel database.	
Included Use Cases	None	
Extending Use Cases	None	
***MAIN SUCCESSFUL SCENARIO for this Use Case* in**	**Actor Action**	**System Action**
	Step 1. Buyer inputs information of his dragon cards.	Step 2. System sends request to Drexel database

numbered sequence Reference "included use cases" in this section using INCLUDE *ius_name*		Step 3. Drexel database verifies card information
		Step 4. System sends confirmation to buyer.
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	**Conditions**	**Actions**
	Step 1. Buyer inputs information of his dragon cards.	Step 2. System sends request to Drexel database
		Step 4. Drexel database can't verify the dragon card.
		Step 5. System sends alert to buyer.
Priority in scheduling	High	
Frequency	Once create payment method	
Business rules and data logic	If there is anything wrong about the information of the Drexel account, Drexel database should send fail information back.	
Other non-functional requirements	Drexel database connect Internet	
Superordinates	B8 Verify Bank Account	
Developer	Xujian Zhang	
Creation date and last modified date	03/17/2018	
Other Comments		

3.4.3 Base Case B5 From Wenyu He

USE CASE #	B5	
USE CASE Name	Create Order	
ACTOR	Buyer	
Goal (1 phrase)	To create online orders by an online buyers	
Overview and scope	The buyers will choose items they want in Dbay system, Dbay system will calculate price and require users to type shipping information and billing information. The system will help buyers to completes the order, including branches to programs that place order and generate invoice; The system also could aborts transaction when payment declined or buyers do not complete the payment	
Level	Base	
Preconditions	Buyer logs into Dbay account	
Postconditions in words (write in passive and past tense)	An invoice object currentOrder was created. Attribute currentOrder.orderID, currentOrder.buyerID, currentOrder.sellerID, currentOrder.itemID, currentOrder.itemQuantity, currentOrder.orderDate, currentOrder.shippingAddress were created.	
Trigger	This event is triggered when buyer selects a item or items into shopping cart	
Included Use Cases	Inc2 Place Order Inc3 Generate Invoice	
Extending Use Cases	B5 Ext1 Update Order	
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence**	**Actor Action**	**System Action**
	1. Buyer logs into account	2. System presents introductory screen and navigation bar for each classifications
Reference “included use cases” in this section using INCLUDE	3. Buyer go to each classification and select all items they want in Dbay Website	4. System adds items in the shopping cart and ask buyers if they are ready to check out or want to continue shopping

ius_name	5. Buyer go to shopping cart to check out the items	6. System calculates item price
	7. Buyer confirm the price or enter promotion code and continue check out	
	8. Buyer select payment methods	
		9. INCLUDE Place Order
		10. INCLUDE Generate Invoice
OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension points using EXTEND *eus_name*)	1. Buyer logs into account	2. System presents introductory screen and navigation bar for each classifications
	3. Buyer go to each classification and select all items they want in Dbay Website	4. System adds items in the shopping cart and ask buyers if they are ready to check out or want to continue shopping
	5. Buyer go to shopping cart and update their order	6. System show update pop up window on screen to ask users to confirm the update
	7. Buyer confirm the order update	8. System recalculate the new prices for all items
	9. Buyer confirm the price or enter the promotion code	10. System process all updates
	11. Buyer Confirm and Continue to check out	
	12. Buyer select payment methods	
		8. EXTEND B2 Ex1 Update Order

OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension points using EXTEND *eus_name*)	1. Buyer logs into account	2. System presents introductory screen and navigation bar for each classifications
	3. Buyer go to each classification and select all items they want in Dbay Website	4. System adds items in the shopping cart and ask buyers if they are ready to check out or want to continue shopping
	5. Buyer go to shopping cart and update their order	6. System show update pop up window on screen to ask users to confirm the update
	7. Buyer confirm the order update	8. System recalculate the new prices for all items
	9. Buyer confirm the price or enter the promotion code	10. System process all updates
	11. Buyer Confirm and Continue to check out	
	12. Buyer select payment methods	
		13. INCLUDE Place Order
		14. INCLUDE Generate Invoice
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	1. Buyer logs into account	2. System presents introductory screen and navigation bar for each classifications
	3. Buyer go to each classification and select all items they want in Dbay Website	4. System adds items in the shopping cart and ask buyers if they are ready to check out or want to continue shopping
	5. Buyer go to shopping cart to check out the items	6. System calculates item price
	7. Buyer confirm the price or enter promotion code and continue check out	

	8. Buyer select payment methods	
	9. Buyer payment methods are declined.	10. System aborts transaction
Priority in scheduling	High	
Frequency	Once per run	
Business rules and data logic	1) If buyers have promotional code, they could enter code in the check out process 2) After opening account in Dbay system, buyer could create order, update order, cancel order, or place bid based on buyer's needs 3) The Dbay system accepts multiple payment methods, however, Dbay system abort transaction if payment methods are declined 4) There are multiple ways for buyers to cancel order after they placed the order, including phone, fax, or a written request and must be processed by a staff	
Other non-functional requirements	Windows standard user interface Daily database backup Inventory updates	
Superordinates		
Developer	Wenyu He	
Creation date and last modified date	03/17/2018	
Other Comments		

USE CASE #	B5 Ext1

USE CASE Name	Update Order	
ACTOR	Buyer	
Goal (1 phrase)	To update online items into shopping carts	
Overview and scope	The buyers could update items they want in Dbay system, Dbay system will recalculate price and require users to type shipping information and billing information. The system will help buyers to completes the order, including branches to programs that place order and generate invoice; The system also could aborts transaction when payment declined or buyers do not complete the payment	
Level	<<Extend>>	
Preconditions	Buyer logs into Dbay account	
Postconditions in words (write in passive and past tense)	An invoice object currentInvoice was updated. Attribute currentInvoice.orderID, currentInvoice.buyerID, currentInvoice.sellerID, currentInvoice.itemID, currentInvoice.itemQuantity, currentInvoice.orderDate, currentInvoice.shippingAddress updated.	
Trigger	This event is triggered when buyer want to update their online orders before placing their order	
Included Use Cases	Place Order Generate Invoice	
Extending Use Cases		
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence**	**Actor Action**	**System Action**
	1. Buyer logs into account	2. System presents introductory screen and navigation bar for each classifications
Reference “included use cases” in this section using	3. Buyer go to each classification and select all items they want in Dbay Website	4. System adds items in the shopping cart and ask buyers if they are ready to check out or want to continue shopping

INCLUDE *ius_name*	5. Buyer go to shopping cart and update their order	6. System show update pop up window on screen to ask users to confirm the update
	7. Buyer confirm the order update	8. System recalculate the new prices for all items
	9. Buyer confirm the price or enter the promotion code	10. System process all updates
	11. Buyer Confirm and Continue to check out	
	12. Buyer select payment methods	
		13. INCLUDE Place Order
		14. INCLUDE Generate Invoice
UNSUCCESSFUL SCENARIOS (*erroneous* situations*)*	**Conditions**	**Actions**
	1. Buyer logs into account	2. System presents introductory screen and navigation bar for each classifications
	3. Buyer go to each classification and select all items they want in Dbay Website	4. System adds items in the shopping cart and ask buyers if they are ready to check out or want to continue shopping
	5. Buyer go to shopping cart to check out the items	6. System calculates item price
	7. Buyer confirm the price or enter promotion code and continue check out	
	8. Buyer select payment methods	

	9. Buyer payment methods are declined.	10. System aborts transaction
Priority in scheduling	Medium	
Frequency	Once per run	
Business rules and data logic	1) If buyers have promotional code, they could enter code in the check out process 2) After opening account in Dbay system, buyer could create order, update order, cancel order, or place bid based on buyer's needs 3) The Dbay system accepts multiple payment methods, however, Dbay system abort transaction if payment methods are declined 4) There are multiple ways for buyers to cancel order after they placed the order, including phone, fax, or a written request and must be processed by a staff	
Other non-functional requirements	Windows standard user interface Daily database backup Inventory updates	
Superordinates	B5 Create Order	
Developer	Wenyu He	
Creation date and last modified date	03/17/2018	
Other Comments		

USE CASE #	Inc 2
USE CASE Name	Place Order
ACTOR	Buyer

Goal (1 phrase)	After buyers select payment method and enter shipping and billing information, system will process the payment and help users place the order	
Overview and scope	The system accepts the user's payment information and passes information along to the bank's verification system or third party payment method such as Paypal, Dragon dollar etc.	
Level	<<Included>>	
Preconditions	Buyer logs into Dbay account.	
Postconditions in words (write in passive and past tense)	A Payment object *currentTransaction* created. Attribute *currentTransaction.transactionID, currentTransaction.oderID, currentTransaction.buyerID, currentTransaction.sellerID, currentTransaction.pamentMethod,* *currentTransaction.paymentAmount created.*	
Trigger	This event is triggered when buyers select the payment methods	
Included Use Cases		
Extending Use Cases		
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference "included use cases" in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
		1. System Prompts Buyer for Payment
	2. Buyers select payment methods and enter shipping and billing information	3. System passes information along to the bank's verification system or third party payment method
		4. If bank or third party approves payment, System notifies buyers that Order is successfully placed.
UNSUCCESSFUL SCENARIOS (*erroneous*		1. System Prompts Buyer for Payment

situations*)*	2. Buyers select payment methods and enter shipping and billing information	3. System passes information along to the bank's verification system or third party payment method
		4. If bank or third party does not approve payment, System notifies buyer that Order is not complete because payment is declined and aborts Transaction.
Priority in scheduling	Medium	
Frequency	Once per run	
Business rules and data logic	If Payment is declined, transaction is aborted.	
Other non-functional requirements	Windows standard user interface Inventory updates	
Superordinates	B5 Create Order	
Developer	Wenyu He	
Creation date and last modified date	03/17/2018	
Other Comments		

USE CASE #	Inc 3
USE CASE Name	Generate Invoice
ACTOR	Dbay System
Goal (1 phrase)	If bank or third party approves payment, order will be complete, and Dbay system will generate an invoice for item purchased by buyer
Overview and scope	The system generate an invoice when order is successfully placed

	and buyers could use invoice number to check the order status by logging into account	
Level	<<Included>>	
Preconditions	Buyer place the order using Dbay account	
Postconditions in words (write in passive and past tense)	An invoice object currentInvoice was created. Attribute currentInvoice.orderID, currentInvoice.buyerID, currentInvoice.sellerID, currentInvoice.itemID, currentInvoice.itemQuantity, currentInvoice.orderDate, currentInvoice.shippingAddress created.	
Trigger	This event is triggered when buyers successfully place the order on Dbay	
Included Use Cases		
Extending Use Cases		
***MAIN SUCCESSFUL SCENARIO for this Use Case* in numbered sequence** Reference "included use cases" in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
		1. System accept the approval from bank or third party payment method
		2. System notifies buyers that Order is successfully placed and generate an invoice number to buyer
	3. Buyer log into account and use invoice number to check order status	
UNSUCCESSFUL SCENARIOS (*erroneous* situations)		1. Bank or Third Party does not approve payment, System aborts Transaction.
		2. System notifies buyers that Order is failed because payment is declined.
Priority in scheduling	Medium	

Frequency	Once per run
Business rules and data logic	If payment complete, buyers receives invoice number and check order status online If payment does not complete, buyers receives notice that payment is declined and order is failed
Other non-functional requirements	Windows standard user interface
Superordinates	B5 Create Order
Developer	Wenyu He
Creation date and last modified date	03/17/2018
Other Comments	

3.5 Discussion

To form our use cases, we relied on the requirements and goals of the system as well as our investigations on the reference websites (eBay). Once a basic framework of our system has been established, we are able to focus on how users can use the system and how the online selling/buying process is operated. We decided that since DBay is a Drexel version of eBay, it would be reasonable to use eBay as our basic reference. Most use cases and procedures and leaned from eBay system but being simplified due to the limit of our time and capability.

4 Class Model

4.1 Design procedure for Class Diagram

Noun	Elimination Note	Applicable Modeling Rule	Possible attributes
DBay	Irrelevant		
Drexel	Irrelevant		
Buyer		Needs ID, has	Name, email,

		attributes	address, phone
Seller		Needs ID, has attributes	Name, email, address, phone
Item		Needs ID, has attributes	Name, price, sale type,
Drexel community	Can be broken into buyer and seller		
Drexel email	Could be attribute		
order		Needs ID, has attribute	Order ID, item ID, quantity, price
Payment method		Could have different kinds	Type, account number, bank info.
Credit card	Could be one type of payment method.		
Debit card	Could be one type of payment method.		
Dragon dollars	Could be one type of payment method.		
Payment amount	Could be attribute of order		
feedback	Could be same as Rating		
Rating		Has attribute	BuyerID, item ID, rating
Payment		Could have discount	accountID, discount, itemID
Shopping Cart		Has attribute	buyerID, item ID, quantity
bid	could be recorded		
Bid Record		Has attribute	Buyer, item, price,

			time
auction	Same as bid		
Cancellation		Needs ID, has attributes	cancelID, invoice ID, item ID, quantity
Invoice		Needs ID, has attributes	Invoice ID< order ID, item, quantity, trackinginfo
Return		Needs ID< has attribute	returnID, invoice ID, item, quantity
employee		Needs id, has attribute	Employee id, name, email
Violation		Needs id, has attribute	User id, employee id, block reason
Refund			Refund id, amount, return id.

4.2 Class Diagram

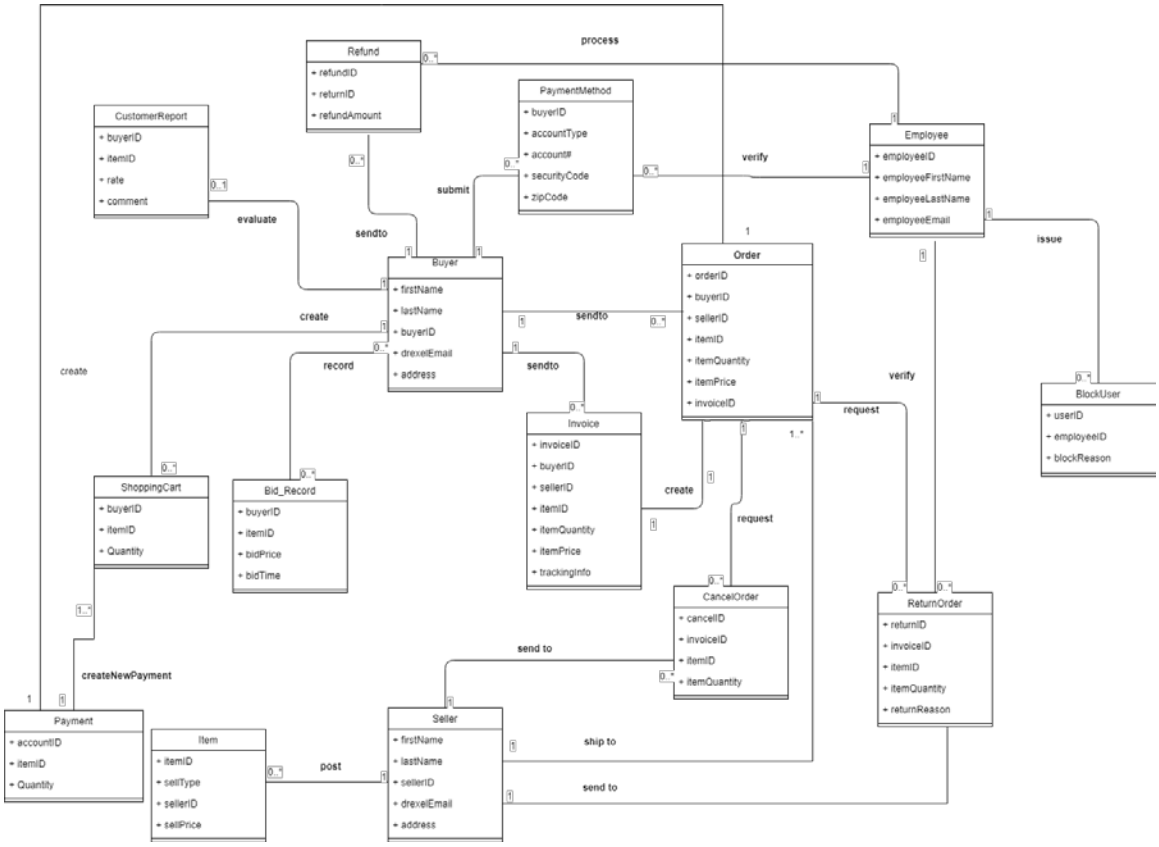


Figure 2 Class Diagram of DBay system

4.3 Selected Class and Association Definition

Buyer: User who purchases items on the DBay website. Buyer can *purchase* **Buy-it-now item** directly or make a *bid* for **Bid item**. Buyer can *initiate* an **Invoice** by making an order and *initiate* a **CancelOrder** or **ReturnOrder** after an order has been made. User needs to *initiate* an **payment** when making an order. After the order has been completed, **User** can *evaluate* the **Seller** as a **CustomReport**.

Seller: User who sells items or services on the DBay website. Seller can *post* **Item** on the system for sale, and the Item can be either **Buy-it-now Item** or **Bid Item**.

Invoice: **Invoice** include the information for an order made by the **Buyer**. It will be *send to* the **Seller** after its being made.

CancelOrder: **Buyer** can cancel part of or entire order within a certain time after the order is made. **CancelOrder** is *initiated* by the **Buyer** and *send to* the **Seller**.

ReturnOrder: Buyer can return part of or entire order within a certain time after the order has been received. **ReturnOrder** is *initiated* by the **Buyer**, and will be *processed* by **Employ**. After being processed, the **ReturnOrder** will be *send to* the **Seller**.

Item: Item listed by the **Seller** for sale on the DBay website. It can be either **Buy-it-now Item** or **Bid Item**.

CustomerReport: After an order has been completed, the **Buyer** can *evaluate* a rating to the **Seller** based on his/her experience towards the item/service.

Refund: When a **ReturnOrder** has been completed, a **Refund** will be send back to the **Buyer. Refund** will be *issued* by the **Employ** and *send to* the **Buyer.**

Employ: Employ who works in the DBay groups. **Employ** can *process* **ReturnOrder**, *verify* **Transaction**, *issue* **Refund**, and *process* **BlockUser**, etc.

BlockUser: User can be blocked from the DBay system when he/she violates the Dbay policies. **BlockUser** is *processed* by **Employ**.

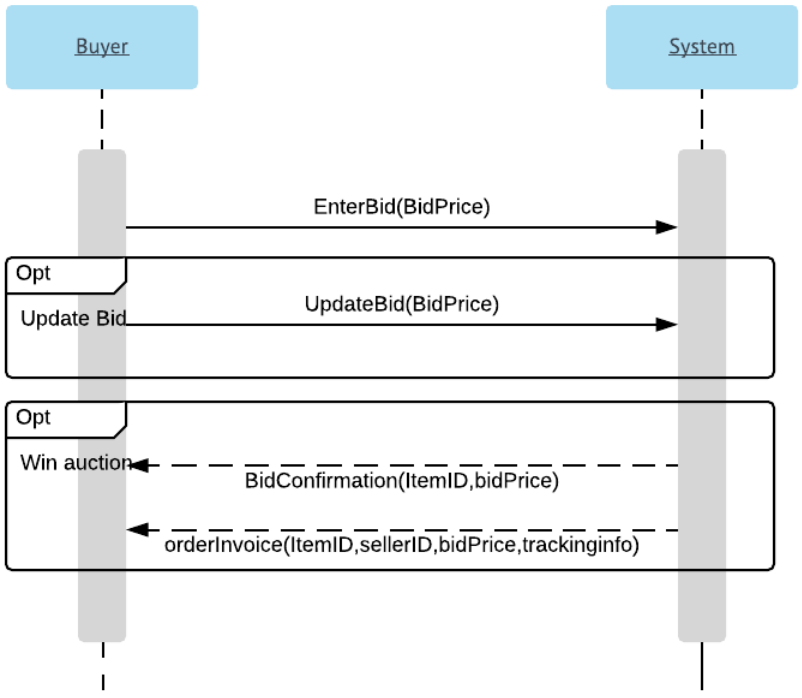
4.4 Discussion

It is a huge challenge for us to design an appropriate class model. We tried to investigate into our previous statements to find our all possible nouns, and discussed together to decide whether a instance should be a class or not.

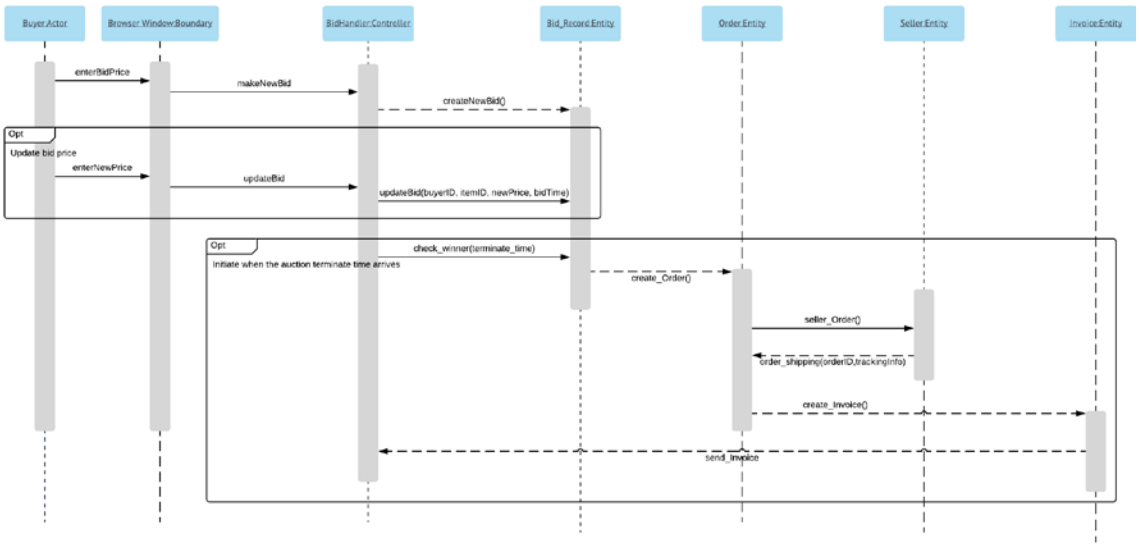
5 Sequence Diagram

5.1 Base Case B5 from Xiang Liu

5.1.1System Sequence Diagram



5.1.2Sequence Diagram



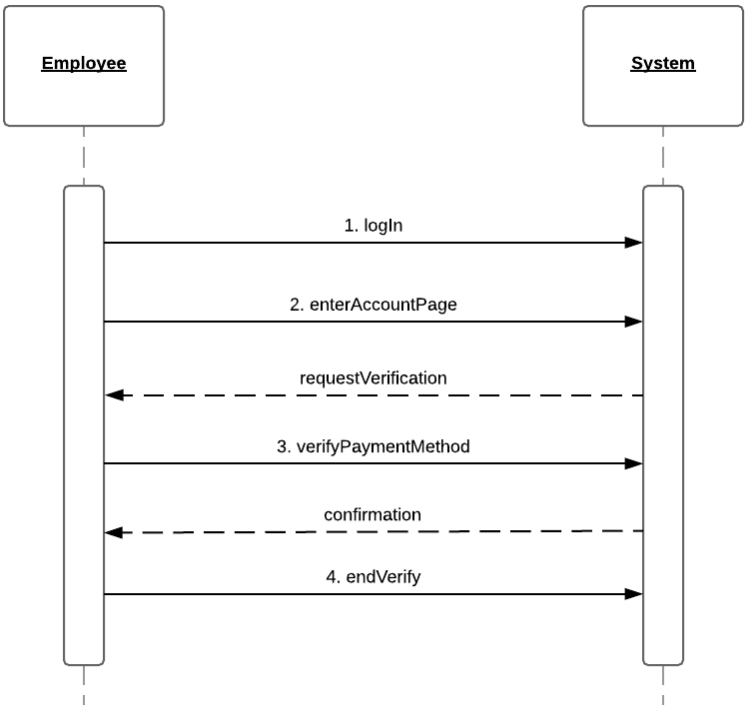
5.1.3 Discussion

This sequence diagram shows the process how a bid order has been completed. First, the buyer enters the item page to submit a bid price. The page will pass the bid price to the bid handler which works as a controller. The controller will create a new bid record to store this bid price in the database. This is the base process of this procedure. As an optional procedure, the buyer can submit a new bid price after he has bid for one item. The updated bid price will be pass to the controller, and the controller will update the bid record entity. As another optional procedure, when the auction finishes, the controller will check the winner in the bid record as the buyer with the highest bid price. Then an new order will be created, and the information will be send to the seller. After the seller ships the item and enter the shipping information into the system, an invoice will be generated and send back to the controller.

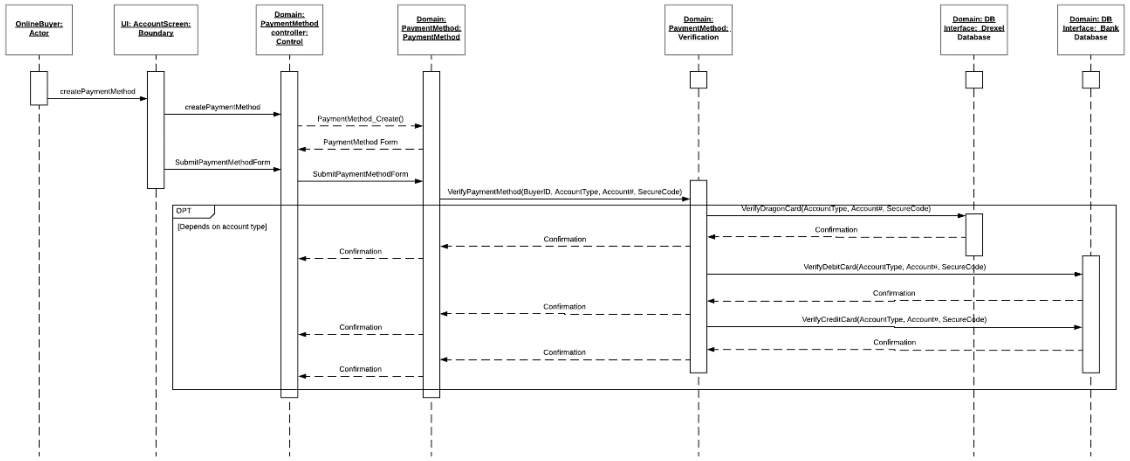
This is the most direct and straight forward process for the auction process we have figured out. We tried some other models, but most of them contains either redundant process or redundant entities.

5.2 Base Case #8 From Xujian Zhang

5.2.1System Sequence Diagram



5.2.2 Sequence Diagram



5.2.3 Discussion

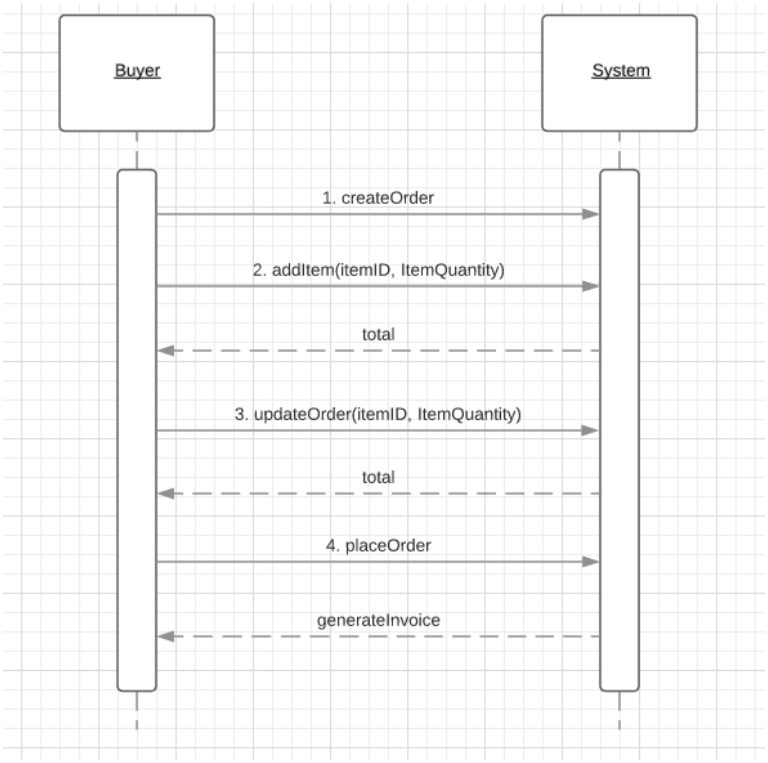
For the sequence diagram of I do, at first, the online buyer as the online buyer wants to create a new payment method. So he opens a new screen which is the boundary of account screen and press to create. Then the system controller gets a request of create payment method. Next system initiates the method and generate new object of payment

method. Then the buyer receives a form and he fills it. Next the form will be sent to payment verification with the buyer's BuyerID, AccountType, Account# and SecureCode. Depends on the different types of payment, the verification will send the request to different destination which is Drexel database or bank database. If the process is successful, buyer will get the confirmation which comes from Drexel database or bank database.

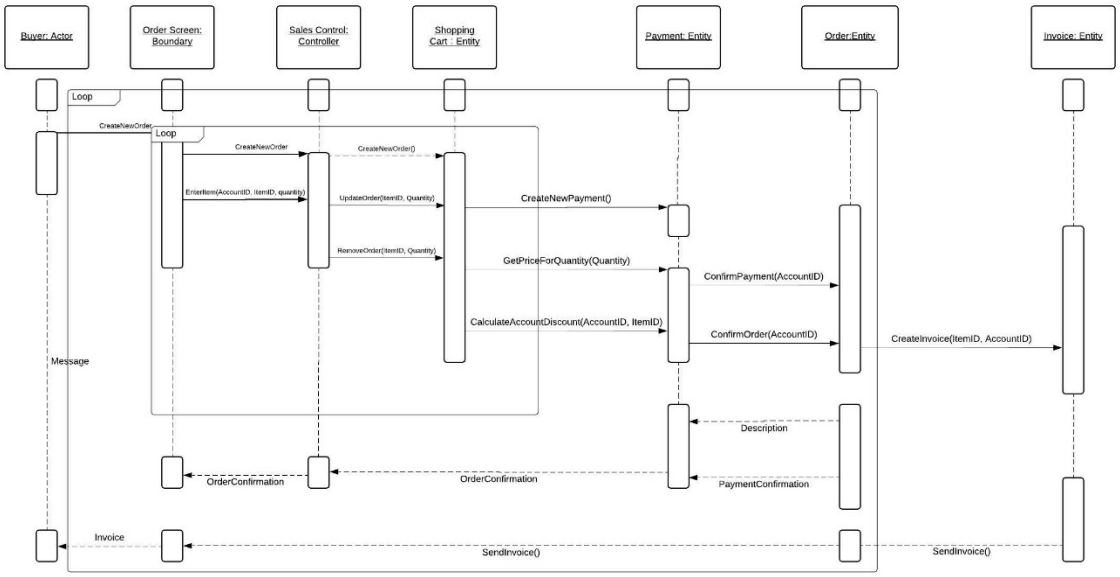
This model is designed to create a new payment method. To compare with the create order model, this model doesn't need to care about the transaction, create order or cancel order. The process occurs once a buyer trying to create a new payment method. By exchanging ideas, this model now is clear and easy understand.

5.3 Use Case #5 from Wenyu He

5.3.1 System Sequence Diagram



5.3.2 Sequence Diagram

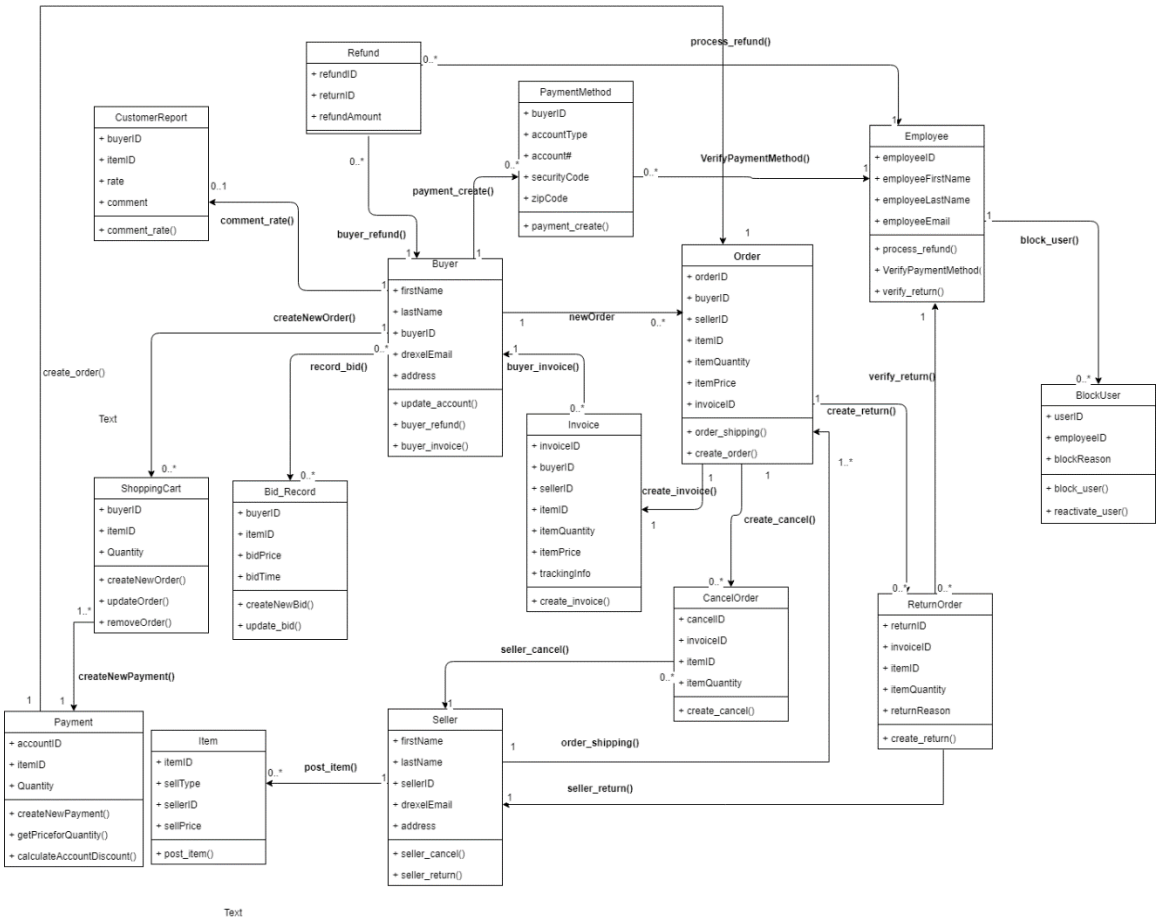


5.3.3 Discussion

Buyer is an actor in the Dbay System and buyer is able to create new order when buyer see the order screen as a boundary in the Dbay System, this process includes add new orders and update existing orders. Buyer is able to add, update, or remove orders by typing ItemID and quantity or go to shopping cart to modify existing orders. Dbay system could control the sales by monitor the AccountID, ItemID, and Quantity of items. When items are ready to check out, system will automatically calculate price and discount based on item and quantity or offer discount to loyal account. Order entity will confirm the payment and order on account based, once order is successfully placed, order entity will send create invoice method to invoice entity and invoice entity send invoice back to order entity, order entity will confirm the payment back to payment entity, payment entity will confirm the order to sales controller and sales controller will communicate with external buyer. This model is the core part in the Dbay system, buyer have access to the following actions such as cancel order, place bid, place payment etc. only when they successfully create the order. In addition, the create order module is similar with placing bid module, buyers allowed to apply loyalty discount based on their account.

6 Design Class Model

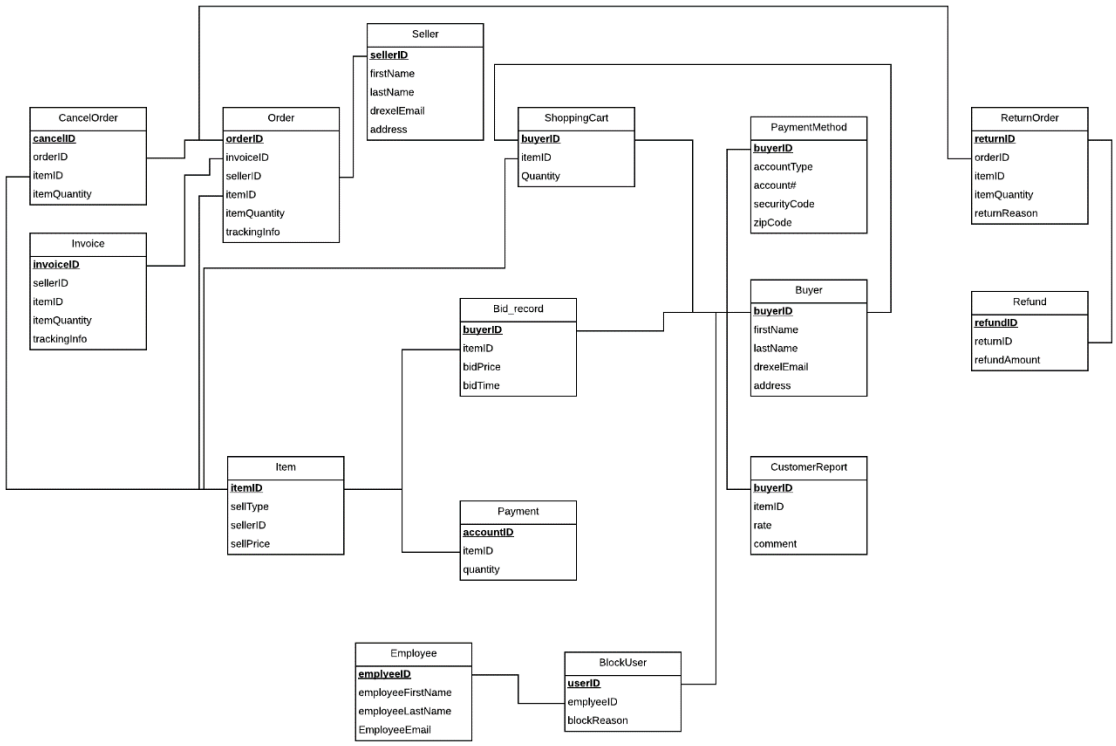
6.1 Design Class Diagram



6.2 Discussion

Every Drexel member include all students, faculties and employees could use the Dbay system and login with his or her Drexel account. For every shopping, the buyer could add items to the shopping cart. The buyer could buy the items immediately or choose to bid. If the buyer wants to pay for items, first he or she needs to create a new payment method. The payment method should be verified by verification system. If create payment method successfully, the buyer could pay for the items and an order will be generated. Also an invoice will be created. The buyer could choose cancel the order or return the order if there is anything wrong. For seller, he or she could provide items after log-in. For employee, he or she could block any user or verify the payment method after a payment method has been created. We always update our DCD and try to make it more readable and usable. Finally, we finished this edition and it shows all the relationships between every entities.

7 Relational Database Schema



8 Evaluation of Analysis & Design

8.1 Evaluation of Project

This project gives us a very good example of the procedure and factors in system designing. We chose this topic since online shopping is common right now, and it is almost the major shopping method for us now. We thought that this topic would be very close to our daily life and it should be relatively easier. However, as our project developed further and further, we found more and more challenges not expected. Though there is only buyer, seller, and employee are involved in our system as actors, we have continuously suffered from difficulties and confusion even in very trivial things.

The effort in finishing this project is great. It is reasonable to separate the whole project into several milestones. We are also able to go back and revise our previous milestones as we are working on new content. This help us understand how an iterative development is conducted. With so much collaboration and effort within the group, we believe that our work successfully shows the basic framework of our system.

8.2 Evaluation of UML and TOOL

UML is a very good tool in system design. It gives very intuitive way to visualize the design of a system. By using UML, we can clearly and easily show the flow of data within a system and show the structure of a system. One possible disadvantage we met is that there are no strict standard rules for the diagrams, such as use case diagram, class

model, and design class model. Different resources could use different shapes and symbols in drawing the diagrams. Though they are similar to each other and there is no difficulties for us to read them, it would be best if there is a general and common strict standards that all tools can follow.

In our project, we have tried UML tools such as Microsoft Visio, online tools draw.io and Lucidchart. Microsoft Vision is most advanced and offers most features. However, we think most of the features are redundant for our current project and it is relatively complicated to use it to draw our graph. Both draw.io and Lucidchart are online tools that support drawing many kinds of graphs. Lucidchart is more intuitively and user friendly, and the style is more attracting. Therefore, we decided to use Lucidchart to finish our project.

9 Appendix I Lesson Learned

Xujian Zhang

From the project, I learned a lot. The topic Dbay is a funny and interesting topic that it can make all of us be involve in it with passion. Dbay is a reality project which is the same as the website we use every day such as Ebay and amazon. Therefore, we could get a clear direction for our project. Depends on every step of the project, we learned the instruction of a shopping website and knew how does it work.

I got a big mistake when we tried to create sequence diagram for my use case of our project. Thanks for professor, she told me what I missed, what part I should change and how to build a good sequence diagram. After exchanging ideas with my team members, I knew the exactly work that my use case should do.

If I get a chance to recreate our project, first of all I will exchange ideas with my teammates and make sure all process of the project. It's the most important thing and it will save you a lot of time.

Wenyu He

In this project, I not only realize how team working is importance in solving problem but also understand the whole process of system development. Firstly, when we confused about the use case diagram, we contact the software engineer to share his experience in Ebay, Professor also give us the deeply instruction on different function that we should add on Dbay system. Secondly, we encounter a lot of problems when developing the sequence diagram, it is hard to figure out the correct object in the sequence diagram related to our use case diagram, core method, and initialization between each object. Prof.Mo give us a lot of instruction to make us to finish this design.

Xiang Liu

The most valuable experience from this project is learn how to work cooperatively with my team. Sometimes team work means difficulties in separating work and share results with each other. Luckily, it is not true this time. Our team works very well and efficiently. I have learned a lot in finishing a project iteratively as separating the whole project into

different milestones. It makes a complicated and huge project into several simple, single targeted, and straightforward small steps. Another precious experience is learning UML and applying it in system design, especially considering I have nearly no experience in this field. I believe this course will give me uncountable benefits in my future career.

10 Appendix II Division of Work

Most of the work have been done cooperatively. The use case description and corresponding sequence diagrams are done separately as mentioned in the section titles. The overall framework of the system is drafted by Wenyu He. Thanks to the Xujian's experience in database, the database schema could be finished smoothly. Xiang did great in drawing the graphs and combining milestones into this final report.