

Software Engineering

Phase 1

Antoinette Stevens , Daniel Guzman, Italia Tran, Sahar Voghoei

9/9/2014



Dawg Trade

1. INTRODUCTION.....	3
1.1. Purpose of System.....	3
1.2. Scope of System.....	3
1.3. Objectives/Success Criteria	4
2. PROPOSED SYSTEM	4
2.1. Overview	4
2.2. Functional Requirements.....	5
2.2.1. Administrators	5
2.2.2. Regular Users	5
2.2.3. Sellers	6
2.2.4. Buyers	6
2.3. Nonfunctional Requirements.....	7
2.4. Constraints.....	7
2.5. System Models	8
2.5.1. Scenarios	8
2.5.1.1. A user placing a bid and winning an auction.....	8
2.5.1.2. User resets password and updates email	9
2.5.1.3. A new member creating an auction	10
2.5.1.4. A user asking an administrator to add a new subcategory	11
2.5.1.5. Administrator updates site to reflect new categories and membership price	12
2.5.1.6. A user placing a bid and then requesting a deletion of the bid	13
2.5.2. Use Cases	14
2.5.2.1. General Users	14
2.5.2.1.1. Create Account.....	14
2.5.2.1.2. Delete Account.....	16
2.5.2.1.3. Reset Password.....	17
2.5.2.1.4. Sign In.....	19
2.5.2.1.5. Sign Out.....	21
2.5.2.1.6. Change Account Setting.....	22
2.5.2.2. Administrators	24
2.5.2.2.1. Create Category	24
2.5.2.2.2. Create Subcategory	26
2.5.2.2.3. Update Category.....	28
2.5.2.2.4. Delete Category.....	31

2.5.2.2.5. Update Membership Price	33
2.5.2.2.6. Add Item Attribute	35
2.5.2.2.7. Delete Item Attribute	37
2.5.2.3. Auctions	38
2.5.2.3.1. Create Auction Item	38
2.5.2.3.2. Update Auction Item	40
2.5.2.3.3. Place Bid	42
2.5.2.3.4. Win Auction Item	44
2.5.2.3.5. Alert User of Outbid	46
2.5.2.3.6. Delete Auction Item	47
2.5.2.3.7. Delete Bid	49
2.5.2.3.8. Update Item Attribute	51
2.5.2.4. General	53
2.5.2.4.1. Search for Item	53
2.5.2.4.2. Inactive for 10 minutes	56
2.5.2.4.3. Navigate as Guest	57
2.5.3. Use Case Model	58
2.5.4. Domain Object Model	59
2.5.4.1. Real Life Objects	59
2.5.4.1.1. Category	59
2.5.4.1.2. Item	59
2.5.4.1.3. Item Attribute	59
2.5.4.1.4. User	60
2.5.4.1.5. Admin	60
2.5.4.2. Business Objects	60
2.5.4.2.1. User Account	60
2.5.4.2.2. Sell (info)	61
2.5.4.2.3. Bid (info)	61
2.5.5. User Interface Sketches	62
3. GLOSSARY	66

1. Introduction

1.1. Purpose of System

The system is intended to provide a place for people in Athens to trade goods by selling to the highest bidder. It should allow administrators to define categories for goods to be sold and associate attributes to them, and in turn allow users to sell items in a given category by creating an auction for them and providing the relevant attribute information for their items. Buyers should be able to come in, place bids, and win auctions, allowing for them to meet with the seller offline and complete the transaction. The system is designed to cater to the residents of Athens and provide them with the means to sell things satisfactorily.

1.2. Scope of System

The system is designed to allow for efficient buying and selling of items, as well as administration of different item categories and sales. The key actors are buyers, sellers, and administrators, and their interests are as follows:

Actor	Interests
Administrator	<ul style="list-style-type: none">- Set price of admission to site- Organize items (create categories, sub-categories, delete categories/subcategories)- Moderate item listings (delete items if necessary)
Buyer	<ul style="list-style-type: none">- Locate items for sale via navigation or search- Bid on items- Keep track of items bid on and won- Receive notifications if bid is winner- Communicate with seller if item is won.
Seller	<ul style="list-style-type: none">- Sell items in appropriate category- Set minimum bid and information regarding items- Monitor bids on items for sale- Communicate with winning bidder to sell items.

1.3. Objectives/Success Criteria

The system should allow administrators to create, edit, and delete categories, define a hierarchy of categories, delete items when necessary, and define a membership price.

Sellers must be able to create, edit, and delete auction listings, place their listings in the appropriate category, keep track of bids submitted, re-enter any unsold items, and contact the winners of auctions they've created.

Buyers must be able to place bids on items for auction, keep track of and increase their bids on items they wish to buy, get notified when their bid is the winning bid for an auction, and get in touch with the seller for items they've won in auctions.

All users must be able to register an account for the administrator's predetermined fee, log in with their username and password, and reset the password via email if necessary. Furthermore, all registered users must be capable of acting as both buyers and sellers.

2. Proposed System

2.1. Overview

The system does not allow guests to view the site. The guest must sign up with a membership fee in order to use any of the functions available to members. Regular members can edit their info, including email and password; and even delete their account entirely. There is also a "soft reset" for members that are inactive past 10 minutes. The system logs the member out and they have to sign in once more to use the site. Members can become sellers by creating and edit an auction to sell old items. They can also send requests to administrators to open up more specific categories and subcategories. Other members can become potential buyers by bidding. While having less power than sellers, they can still send requests to cancel their bids, however it is up to the seller's discretion to approve or not. The system keeps track of potential buyers and sends email regarding buyers who outbid others and winners of auctions.

Administrators have all the functions of regular members with some more advanced powers. They have the power to modify (add/edit/remove) categories and subcategories. They also have the power to add and remove attributes to subcategories to better adjust how members add and search items. They can also delete auctioned items for sellers if they deem it necessary. Administrators can also adjust membership prices if necessary to allow more or less exclusivity.

2.2. Functional Requirements

Functional requirements describe the interactions between the system and its environment independent (actors) from the implementation.

2.2.1. Administrators

Number	Name	Description
2.2.1.1	Create a category	Admin creates a new category for users to search and add items
2.2.1.2	Update/delete category	Admin deletes a category all together or update categories to be more specific
2.2.1.3	Create subcategory	Admin creates a new subcategory to be more specific
2.2.1.4	Update/delete subcategory	Admin deletes an unnecessary subcategory or update to be more specific
2.2.1.5	Delete auctioned items	Admin deletes auctioned item per request of seller
2.2.1.6	Print reports of items sold or on sale	Admin prints out a report on items sold for the past time period or currently on sale
2.2.1.7	Define membership price	Admin adjusts the price for new members of the site
2.2.1.8	Create item attribute	Admin creates a new item attribute for specific subcategories
2.2.1.9	Delete item attribute	Admin deletes an item attribute for specific subcategories

2.2.2. Regular Users

Number	Name	Description
2.2.2.1	Creating an account	Guests can become new users by creating an account on the site
2.2.2.2	Deleting an account	Users can remove themselves from the database by deleting their account
2.2.2.3	Signing in	Users have to sign in in order to view the site
2.2.2.4	Signing out	Users sign out to protect their data and information
2.2.2.5	Reset password	Users can reset password if they forgot it
2.2.2.6	Change profile information	Users can change email and password if they need to

2.2.2.7	Browse items for sale	Users can browse items for sale in each category
2.2.2.8	Search for items	Users can specifically search for items they want

2.2.3. Sellers

Number	Name	Description
2.2.3.1	Add new item for auction	Sellers can add a new item for auction
2.2.3.2	Keep track of bids on items	Sellers can keep track of bids (and bidders) of an auction
2.2.3.3	Update auctioned items	Sellers can update the information of an item
2.2.3.4	Learn identity of winning bidder	Sellers learn who won the auction
2.2.3.5	Delete a bid	Sellers can delete a bid per request of a potential buyer
2.2.3.6	Re-enter unsold item for auction	Sellers can re-enter an item for auctioned if it was not bought
2.2.3.7	Update item attribute	Sellers can update the attributes of an auctioned item

2.2.4. Buyers

Number	Name	Description
2.2.4.1	Place a bid	Buyers can place a bid on items they want to buy
2.2.4.2	Keep track of bids	Buyers can keep track of what items they have bid on
2.2.4.3	Notification emails	Buyers get notified by email if they are outbid or if they are the winner

2.3. Nonfunctional Requirements

Nonfunctional requirements are aspects of the system not directly related to functional behavior.

Number	Description
2.3.1	The site should protect the user information
2.3.2	The user should not be inactive for more than 10 minutes
2.3.3	Searching for items should not take longer than 5 seconds
2.3.4	Responding to submissions/changes should not take longer than 5 seconds
2.3.5	System crashes should not result in user data loss
2.3.6	The system's interface should provide ease of use to the user
2.3.7	The System must be on and functioning 90% of the time

2.4. Constraints

Constraints are imposed by the client or environment. They are also called “pseudo requirements.”

Number	Description
2.4.1	The user has to be a UGA student
2.4.2	The system must be coded in Java
2.4.3	The system must be accessible from a common web browser
2.4.4	The system must provide multi-user access
2.4.5	The system must maintain suitable authorization information and enforce access validation
2.4.6	The system must have easy-to-use user interface
2.4.7	The system must use a persistent data storage
2.4.8	The system should use accepted standards

2.5. System Models

2.5.1. Scenarios

2.5.1.1. A user placing a bid and winning an auction

Scenario	A user placing a bid and winning an auction
Use Cases	Search for Item Place Bid Alert Users of Outbid Win an Auction Item
Actors	John - Buyer Adrian - Seller Sara - Buyer
Description	<p>John, a UGA Student and member of DawgTrades.com, needs to find a new desk for his apartment. Going on the site, he selects the appropriate category and begins his search. He finds a used desk for sale by Adrian on DawgTrades.com. John reads the description of the desk and sees that it is lightly used and only 2 years old. He can see from the picture on the page that the desk seems to be in good condition. John also notices that the highest bid for the desk is only \$30, placed by Sara, and he is definitely willing to go higher.</p> <p>Seeing that he has only has 5 minutes left to bid, John quickly click the 'Place a Bid' button. He types in '\$40' as his bid and waits for the success message from the system. John and Adrian receive an email from DawgTrades.com notifying both parties that the bid was placed. The bid ends and John is the winner. Both, John, Sara, and Adrian receive an email that John had the winning bid for the desk with \$40. John and Adrian exchange information and set up a time for John to get the desk and pay Adrian.</p>

2.5.1.2. User resets password and updates email

Scenario	User resets password and updates email
Use Cases	Reset Password Change Account Setting
Actors	Tony - User
Description	<p>Tony, an existing user of DawgTrades, has not been on the site for quite a while. When he tries to login, the system does not recognize his password so he decides to reset it. When prompted, he enters his registration email address, however, Tony intends to change this to a more recent email address as the he rarely uses the previous.</p> <p>When reading the email, DawgTrades directs him to a link asking him for a new password and for him to confirm it. After changing it and logging in, Tony goes to his personal information panel and changes his email address to reflect his new email.</p>

2.5.1.3. A new member creating an auction

Scenario	A new member creating an auction
Use Cases	Create Account Create Auction Item Update Auction item Update Item Attribute
Actors	Jane - Seller Tom - Buyer
Description	<p>Jane, a student at UGA, heard about DawgTrades through one of her friends and decided to try and sell her biology textbook on it. When she goes on the website, DawgTrades.com, the screen asks her to sign-in or create a new account. When making a new account, the site prompts her for a few things: first and last name, desired username, email address, a contact number and a checkbox if she receives texts. After she entered all the information, DawgTrades asks her to pay the membership fee and Jane agrees to it.</p> <p>Once done with registration, Jane starts her auction on her biology textbook. She selects the appropriate category and entered a few details about the auction: time constraint of the auction, starting bid, and a small description of the textbook. After reviewing over her information, Jane submits the item for bids and waits for a winner. However, Tom, another user, messages Jane to ask if her textbook is the correct edition. Realizing she wrote down an older edition of the textbook. Jane goes back on the item's page and edit the correct edition in.</p>

2.5.1.4. A user asking an administrator to add a new subcategory

Scenario	A user asking an administrator to add a new subcategory
Use Cases	Create Subcategory Add Item Attribute
Actors	Jane - User Wesley - Administrator
Description	<p>Jane, an existing user for a few months, enjoys her time on DawgTrades and sees that it is a good site to sell things she does not need anymore. However, she realizes that the site does not include a subcategory of art supplies under the school supplies category. Jane sends a request to Wesley, an administrator, asking if he can create a new subcategory for her.</p> <p>Wesley, seeing that the subcategory was a good idea, decides to go ahead and add it. At the administrative panel, Wesley chooses the option to create a new subcategory. When prompted, Wesley types, submits, and confirms the new subcategory. After the subcategory is created Wesley is then able to edit the attributes of the subcategory in order to achieve a more specific way for users to search and add items.</p>

2.5.1.5. Administrator updates site to reflect new categories and membership price

Scenario	Administrator updates site to reflect new categories and membership price
Use Cases	Delete Category Update Category Update Membership Price
Actors	Adam - Administrator
Description	<p>Adam, an administrator of DawgTrades, decides to do clean up the site a bit as it has been a while since the initial launch of DawgTrades. When perusing categories, he realizes that not many auctions are taking place in “kitchenwares.” However, “tools and gadgets” are doing well, so he decides deletes “kitchenwares” and update “tools and gadgets” to include kitchenwares. After confirming the deletion of “kitchenwares,” Adam goes and updates the attributes of “tools and gadgets” to include kitchenwares from the deleted category.</p> <p>Although the small community is doing well, he realizes that the membership price is too high and it might be a reason why there’s not a lot of new sign-ups. Deciding to change the price, he goes to the admin panel and lower the membership price in hope to attract new users.</p>

2.5.1.6. A user placing a bid and then requesting a deletion of the bid

Scenario	A user placing a bid and then requesting a deletion of the bid
Use Cases	Place Bid Delete Bid
Actors	John - Buyer Mary - Seller Quinn - Buyer
Description	<p>John, an existing user, sees a chair that he wants to buy from Mary. With the auction only having 10 hours left, he decides to place and confirm a bid. The item page shows John being the highest bidder, replacing Quinn, the previous highest bidder. However, with 5 hours left in the auction, John realizes that he found a cheaper chair somewhere else and wants to cancel his bid. John goes to the item page and sends a request to Mary to delete his bid.</p> <p>Mary receives an email notification about the request. Being a nice person, Mary goes and deletes his bid. The updated item page shows that Quinn is now the highest bidder for the chair. John and Quinn gets an email notifying them of the changes: John's request is heard and Quinn is now the highest bidder.</p>

2.5.2. Use Cases

2.5.2.1. General Users

2.5.2.1.1. Create Account

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to create an account for users. They need to be a member for selling, bidding, or buying an item.
Business Trigger:	A user decides to create an account to be part of the system.
Preconditions:	User should be a UGA member.

Basic Flow:		
Assumptions: User is in the “Landing Page” and ready to create a new account.		
Line	System Actor Action	System Response
	User click on “Sign Up” bottom.	The system displays a page with First Name, Last Name, UGA Email, Phone, User Name, Password, and Confirm your Password text box and “Sign Up” bottom. Also There is a link to Sign In page.
	The user provides the required attribute values to the system and clicks “Submit”.	The system displays a home page of website with “Hi Username” on top left corner of page and a dropdown list next to it.
Post Condition:	The new member added to the system users and user is already logged in to the system. In the home page User can see his/her name on top left corner of page with a dropdown menu next to it. In dropdown menu there are Logout, Account Setting, and My Items.	

Alternate Flow (AF1): User decides to Sign In to the system.		
If at line 1 in Basic Flow or line 1 in EF2, user find out that he/she has an account and wants to Sign In, then:		
Line	System Actor Action	System Response
	User click on: Sign In” link.	System displays a “Log In” page
The use case terminate and go to Use Case 1.4		
Post Condition:	N/A	

Exception Flow (EF1): Create an account from an Non-UGA member		
If at line 1 in Basic Flow the user is not UGA member, then:		
Line	System Actor Action	System Response
1	User attempt to add an email address that is not UGA email address	System displays a message that “this email is not UGA email address. Please enter your UGA email address.”
The use case restarts at line 1		
Post Condition:	N/A	

Exception Flow (EF2): Create an account with email address that already exists.		
If at line 1 in Basic Flow the user is already exist, then:		
Line	System Actor Action	System Response
	User attempt to add an email address that is already exist.	System displays a message that “this email is already exist. Please enter your UGA email address or simply Sign In .”
The use case restarts at line 1		
Post Condition:	N/A	

Exception Flow (EF3): Password and re-entered password is not match.		
If at line 1 in Basic Flow the user is providing incorrect re-enter password, then:		
Line	System Actor Action	System Response
1	User attempt to pass Re-enter Password field.	System displays a message that “Re-Enter the same Password.”
The use case restarts at line 1		
Post Condition:	N/A	

Business Rules:
Functional Requirements 2.2.2.1

Non Functional Requirements:
Non Functional Requirements 2.3.2
Non Functional Requirements 2.3.4
Non Functional Requirements 2.3.6
Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
Email	Email format	Message: “UGA email address only – please re-enter”
Password	More than 8 char	Message: “Password should be more than 8 character – please re-enter”

2.5.2.1.2. Delete Account

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to delete an account.
Business Trigger:	A user decides to delete his/her account.
Preconditions:	User should have an account.

Basic Flow:		
Assumptions: User is in the “Account Setting” page and ready to delete his/her account.		
Line	System Actor Action	System Response
1	User click on “Delete Account” bottom at Account Setting page	The system displays a warning message “Dear Username are sure that you want to delete your account”, and a “Submit” and “cancel” Bottom.
2	The user click on “Submit”.	The system displays the landing page.
Post Condition:	User Account will be deleted and no longer active in system.	

Alternate Flow (AF1): Cancel the delete process.		
If at line 1 in Basic Flow the user click on Cancel, then:		
Line	System Actor Action	System Response
1	The user click on “Cancel” bottom.	System displays the Account Setting page.
The use case restarts at line 1		
Post Condition:	N/A	

Business Rules:	
1	User must have an account on DawgTrades
Functional Requirement 2.2.2.2	

Non Functional Requirements:	
Non Functional Requirements 2.3.4	
Non Functional Requirements 2.3.5	
Non Functional Requirements 2.3.6	
Non Functional Requirements 2.3.7	

2.5.2.1.3. Reset Password

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to allow users to reset passwords.
Business Trigger:	A user clicks on reset password from profile page, or on a link in his/her email that sent through forgets password process from Sign in page.
Preconditions:	User is Signed in to the system or received an email to how reset a password.

Basic Flow:		
Assumptions:		
Line	System Actor Action	System Response
	User click on “Reset Password” bottom at his/her profile page. Or, click on a like witch sent to his/her email to reset password.	The system displays a page with New Password and Re-Enter New Password and Submit bottom.
	The user provides the required attribute values to the system and clicks “Submit”.	The system displays the Sign In page.
Post Condition:	The User’s password will be reset.	

Exception Flow (EF1): Incorrect Password format.		
If at line 1 in Basic Flow the user is providing in correct password format, then:		
Line	System Actor Action	System Response
	User attempt to enter password.	System displays a message that “password format is incorrect. Please try again”
The use case restarts at line 1		
Post Condition:	N/A	

Exception Flow (EF2): Password and re-entered password is not match.		
If at line 1 in Basic Flow the user is providing incorrect re-enter password, then:		
Line	System Actor Action	System Response
1	User attempt to pass Re-enter Password field.	System displays a message that “Re-Enter the same Password.”
The use case restarts at line 1		
Post Condition:	N/A	

Business Rules:
User should be a UGA member with valid UGA email address. Functional Requirements 2.2.2.5

Non Functional Requirements:
Non Functional Requirements 2.3.4 Non Functional Requirements 2.3.5 Non Functional Requirements 2.3.6 Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
Email	Email format	Message: "UGA email address only – please re-enter"
Password	More than 8 char	Message: "Password should be more than 8 character – please re-enter"

2.5.2.1.4. Sign In

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to allow user to sign In to the system.
Business Trigger:	A user decides to sign In to the system.
Preconditions:	User should have a valid account.

Basic Flow:		
Assumptions: User knows his/her email and password for this website.		
Line	System Actor Action	System Response
1	User click on “Sign In” bottom at the menu bar under “Your Account”.	The system displays a page with Username and Password text box and “Sign In” bottom. It shows “Sign Up” link and a links for “Forgot your password?”
2	The user provides Email/ID and Password to the system and clicks “Sign In”.	The system displays a home page of website with “Hi user name” on top left corner of page with a dropdown menu next to it.
Post Condition:	User is signed in to the system and is eligible to see and act as an authorized user in the system	

Alternative Flow (AF1): Sign Up to the system		
If at line 1 or 2 in Basic Flow user decides to sign up to the system, then:		
Line	System Actor Action	System Response
1	User Click on “Sign Up” bottom.	System displays a sign up page
The use case terminates and it goes to Use Case 1.1		
Post Condition:	N/A	

Alternative Flow (AF2): Forgot password		
If at line 1 or 2 in Basic Flow user came click on forgot your password and then:		
Line	System Actor Action	System Response
1	Click on “forgot password” link.	System displays a new page with email textbox and “Continue” bottom.
	User provide Email and click “Continue”	System shows a message that tell user to go and check the email to reset his/her password.
The use case terminate and goes to Use Case 1.3		

Post Condition:	User receives an email with a link in it to go to reset password page.
-----------------	--

Exception Flow (EF1): Inputs invalid User Name.		
If at line 2 in AF2. User enter invalid Email address, then:		
Line	System Actor Action	System Response
1	Enter invalid Email	System displays a message that “your Email is not register in this system”
The use case restarts at line 1 in AF2.		
Post Condition:	N/A	

Exception Flow (EF2): Inputs invalid Username or Password		
If at line 2 in Basic Flow the user enter invalid Username or password, then:		
Line	System Actor Action	System Response
1	Enter Email and password and click on submit.	System displays a message that “your Username or Password is incorrect”
The use case restarts at line 1		
Post Condition:	N/A	

Business Rules:
Only the member of system should be allowed to sign in to the system.
Only UGA email can be provide as email address to reset password.
Functional Requirements 2.2.2.3

Non Functional Requirements:
Non Functional Requirements 2.3.4
Non Functional Requirements 2.3.5
Non Functional Requirements 2.3.6
Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
Email	Email format	Message: “UGA email address only – please re-enter”
Password	More than 8 char	Message: “Password should be more than 8 character – please re-enter”

2.5.2.1.5. Sign Out

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to allow user to sign out from system.
Business Trigger:	A user decides to sign out from system.
Preconditions:	User should be signed in before and be an active user.

Basic Flow:		
Assumptions: User is in the “Your Account” menu bar and ready to sign out.		
Line	System Actor Action	System Response
1	User click on “Sign out” bottom at the menu bar.	The system displays the Sign In page.
Post Condition:		User will be signed out from system and is no longer an active user in the system.

Business Rules:	
1	User is a member of DawgTrades Functional Requirements 2.2.2.4

Non Functional Requirements:	
	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5 Non Functional Requirements 2.3.6 Non Functional Requirements 2.3.7

2.5.2.1.6. Change Account Setting

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to give a user chance to edit his/her account information.
Business Trigger:	A user wants to edit his/her account information and click on “Account Setting” on menu bar under “Hi Username”.
Preconditions:	User should be Longed in User.

Basic Flow:		
Assumptions: User is already Longed in to the system.		
Line	System Actor Action	System Response
1	User click on “Account Setting” bottom at the menu bar.	The system displays Change Account Setting page with Name, Email, and Phone labels and an “Edit” bottom in front of each. And password shown in star format and “Reset” bottom in front of it and “Delete Account”, and at the end “Done” bottom.
2	User click on one of the “Edit” bottom.	The system displays a new page to provide new information based on witch bottom were clicked. This page has a textbox to provide the new information and “Save Change” and “Cancel” bottoms
	User Enter the new information in the page and click “Save Change”	System save the information and back to the Account Setting page with edited information.
	User click “Done”	System goes back and display home page.
Post Condition:	User Information changed.	

Alternate Flow (AF1): Password Reset		
If at line 2 in Basic Flow the user click on Reset in front of password, then:		
Line	System Actor Action	System Response
1	User click on Reset Password.	System displays the reset page.
The use case wait for “Use case 1.3” to be done and back to line 4 of Basic Flow		
Post Condition:	Password changed.	

Alternate Flow (AF2): Cancel the edit process

If at line 3 in Basic Flow the user click on Cancel bottom in edit page, then:		
Line	System Actor Action	System Response
1	User click on Cancel in edit page	System back to the User Setting Page without any change.
The use case terminates and reset to line 1 in Basic Flow.		
Post Condition:	N/A	
Business Rules:		
User must be an existing member of DawgTrades		
2	Functional Requirements 2.2.2.6	

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
1 Email	Email format	Message: "UGA email address only – please re-enter"
2 Password	More than 8 char	Message: "Password should be more than 8 character – please re-enter"

2.5.2.2. Administrators

2.5.2.2.1. Create Category

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/4/14
2.0	Uniformed use cases	Italia Tran	9/8/14

Brief Description:	Allows the administrator to create a new category for items to be included and for users to search.
Business Trigger:	The administrator elects to create a new category for items.
Preconditions:	The administrator has to be logged in.

Basic Flow:		
Assumptions: The necessity for a new category to better sort items.		
Line	System Actor Action	System Response
1	The administrator clicks link for adding categories at the administrator panel.	The system directs the administrator to a page with a prompt, submit, and cancel button. The list of categories and its sub-categories is shown below the prompt.
2	The administrator types and submits the category to the system.	The system acknowledges the category and adds it to the list of categories. The new category is added at the bottom of the list.
3	The administrator leaves the listing as is or rearrange in alphabetical order.	The system changes the listing based on what the administrator has requested.
Post Condition:	The listing of categories is updated and ordered to include the new category.	

Alternate Flow (AF1): Canceling the submission		
If at line 1 the administrator thinks that the category is not needed, then he can press cancel to not submit.		
Line	System Actor Action	System Response
1	The administrator decided to cancel the submission.	The system redirect the administrator back to line 1.
The use case restarts at line 1		
Post Condition:	N/A	

Exception Flow (EF1): Error with an existing category		
Line	System Actor Action	System Response
1	The administrator submits an existing	The system prints out a message saying

	category.	that the category already exists on the previous page.
--	-----------	--

Business Rules:		
1	Only the administrator has the ability to add a new category	
2	Functional Requirements 2.2.1.1	

Non Functional Requirements:		
1	Non Functional Requirements 2.3.4	
2	Non Functional Requirements 2.3.5	
3	Non Functional Requirements 2.3.6	
4	Non Functional Requirements 2.3.7	

Screen Entry Exception Table:		
Field	Constraint	Response
1 Category	Duplicate	“Category already exists”

2.5.2.2.2. Create Subcategory

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/4/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	Allows the administrator to create a new sub-category for more specific items to be included and for users to search.
Business Trigger:	The administrator elects to create a new sub-category for items.
Preconditions:	The administrator has to be logged in. A more general category has already been created.

Basic Flow:		
Assumptions: The necessity for a new sub-category to better sort items.		
Line	System Actor Action	System Response
1	The administrator clicks link for adding sub-categories at the administrator panel.	The system directs the administrator to a page with a prompt, submit, and cancel button. The list of categories and its sub-categories is shown below the prompt.
2	The administrator chooses the category that the sub-category most likely fits in. The administrator types in the name of the sub-category and submits.	The system acknowledges the sub-category and adds it to the specified category. If more than one sub-categories exist, the newest is added at the bottom.
3	If more than one sub-categories exist, the administrator can leave as is or sort it alphabetically.	The system changes the listing based on what the administrator has requested.
Post Condition:	The listing of categories and sub-categories is updated and ordered to include the new sub-category.	

Alternate Flow (AF1): Cancelling the submission		
If at line 1 the administrator thinks that the sub-category is not needed, then he can press cancel to not submit.		
Line	System Actor Action	System Response
	The administrator decided to cancel the submission.	The system redirect the administrator back to line 1.
The use case restarts at line 1		
Post Condition:	N/A	

Exception Flow (EF1): Error with an existing category		
Line	System Actor Action	System Response
1	The administrator submits an existing	The system prints out a message saying

	sub-category.	that the category already exists on the previous page.
--	---------------	--

Business Rules:		
1	Only the administrator has the ability to add a new sub-category.	
2	Functional Requirements 2.2.1.3	

Non Functional Requirements:		
1	Non Functional Requirements 2.3.4	
2	Non Functional Requirements 2.3.5	
3	Non Functional Requirements 2.3.6	
4	Non Functional Requirements 2.3.7	

Screen Entry Exception Table:		
Field	Constraint	Response
1 Sub Category	Duplicate	“Sub category already exists”

2.5.2.2.3. Update Category

Version	Description	Changed By	Date
1.0	First draft	Daniel Guzman	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	Administrators can edit category information in order to change the title or modify the attributes in a category
Business Trigger:	Information about a category changes and the site should reflect that.
Preconditions:	The system must be online and the category to be edited must already exist and the actor must be logged in as an administrator.

Basic Flow: The administrator selects a category to be edited, changes the information as required, then submits the changes. The system then updates the category.

Assumptions: A category is in need of updating

Line	System Actor Action	System Response
1	The administrator selects a category to be edited in the admin panel.	The system retrieves information about category and returns it to the administrator in a form to be edited.
2	The administrator enters the new category information and/or attributes and submits them.	The system will update the category information in the database.
Post Condition:	The category information will be updated and reflect the administrator's latest changes.	

Alternate Flow (AF1): Changing category to one that's already in use

If at line 2 in the basic flow the administrator changes the title of the category to one that is already in use by a different category, then:

Line	System Actor Action	System Response
1		The system will return an error indicating that the name is already in use.

The use case restarts at line 2 of the basic flow

Post Condition:	Category is not updated (resumes basic flow at line 2).
-----------------	---

Alternate Flow (AF2): Changing category that does not exist		
If at line 1 in the basic flow the administrator selects a nonexistent category, then:		
Line	System Actor Action	System Response
1		The system will return an error indicating that the category does not exist.
The use case restarts at line 1 of the basic flow		
Post Condition:	Category is not updated (resumes basic flow at line 1).	

Alternate Flow (AF3): Cancelling changes		
If at line 2 in the basic flow the administrator opts to cancel the edit and clicks the cancel button, then:		
Line	System Actor Action	System Response
1		The system will redirect them back to the categories page in the admin panel.
The use case restarts at line 1 of the basic flow.		
Post Condition:	Category is not updated (resumes basic flow at line 1).	

Business Rules:
Only administrators should be allowed to edit categories. Editing categories should not require any further interaction (i.e. items in a category should not need to be edited if the category is edited) Functional Requirements 2.2.1.2

Non Functional Requirements:
Non Functional Requirements 2.3.4 Non Functional Requirements 2.3.5 Non Functional Requirements 2.3.6

Screen Entry Exception Table:		
Field	Constraint	Response
Category	Duplicate	“There already exists a category”
Category	Non-Existing	“There are no such category”

2.5.2.2.4. Delete Category

Version	Description	Changed By	Date
1.0	First draft	Daniel Guzman	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	Administrators can opt to delete categories and remove them from the site.
Business Trigger:	A category is no longer needed
Preconditions:	The system must be online, the category to be deleted must already exist, the actor must be logged in as an administrator.

Basic Flow: The administrator selects a category to be deleted. The system confirms then deletes the category.

Assumptions: A category is in need of deletion.

Line	System Actor Action	System Response
1	The administrator selects a category to be deleted in the admin panel.	The system retrieves the count of items in the category. It then asks the administrator to confirm the deletion, showing the item count if any are found.
2	The administrator confirms deletion.	The system deletes all items in the category, then deletes the category itself.
Post Condition:	The system will have removed all items in a category as well as the category itself.	

Alternate Flow (AF1): Deleting category that does not exist

If at line 1 in the basic flow the administrator selects a nonexistent category, then:

Line	System Actor Action	System Response
1		The system will return an error indicating that the category does not

		exist.
The use case restarts at line 1 of the basic flow		
Post Condition:	Category is not deleted (resumes basic flow at line 1).	

Alternate Flow (AF2): Cancelling deletion		
If at line 2 in the basic flow the administrator opts to cancel the deletion and clicks the cancel button in the confirmation prompt, then:		
Line	System Actor Action	System Response
1		The system will redirect them back to the categories page in the admin panel.
The use case restarts at line 1 of the basic flow.		
Post Condition:	Category is not deleted (resumes basic flow at line 1).	

Business Rules:
Only administrators should be allowed to delete categories. Functional Requirements 2.2.1.2

Non Functional Requirements:
Non Functional Requirements 2.3.4 Non Functional Requirements 2.3.5 Non Functional Requirements 2.3.6 Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
Category	Non-Existing	“That category does not exist”

2.5.2.2.5. Update Membership Price

Version	Description	Changed By	Date
1.0	First draft	A. Stevens	9/3/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to describe the case when an administrator needs to update the membership price for DawgTrades.com
Business Trigger:	The membership price needs to be updated based on a business need.
Preconditions:	The system must be running and functioning properly. Actor must have administrator rights.

Basic Flow:		
Assumptions: The user is an administrator with the authorization to alter the membership price		
Line	System Actor Action	System Response
1	The administrator asks the system to change the listed membership price for DawgTrades.com	The system verifies that the user is an administrator.
2	The administrator enters and submits the new membership price.	The system validate the input, updates the database with the new membership price and returns a message upon success.
Post Condition:	The membership price should be updated in the system and site should display the new price wherever the membership price is shown.	

Alternate Flow (AF1): 2.5.1 Update from an unauthorized user		
If at line 1 in Basic Flow the user is not authenticated as a legitimate administrator, then:		
Line	System Actor Action	System Response
1	User attempts to update the membership price	The system attempts to verify user credentials, fails, and returns an error.
2	----	System logs attempted unauthorized access

Alternate Flow (AF2): 2.5.2 Submitting invalid input		
If at line 2 in Basic Flow the user enters invalid input then:		
Line	System Actor Action	System Response
1	User inputs invalid input	The system attempts to validate input and returns an error message upon failure

Business Rules:	
1	If the user is not an administrator, the user should not see an option to update the

membership price.

2 Functional Requirements 2.2.1.7

Non Functional Requirements:

1 Non Functional Requirements 2.3.4

2 Non Functional Requirements 2.3.5

3 Non Functional Requirements 2.3.6

4 Non Functional Requirements 2.3.7

Screen Entry Exception Table:

Field	Constraint	Response
1 New Membership price	Non negative decimal number with two decimal places	Message: "Price cannot be negative – please re-enter"

2.5.2.2.6. Add Item Attribute

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/8/14

Brief Description:	An Administrator adds what type of attribute an item might have
Business Trigger:	Necessity of attribute for an item
Preconditions:	Category and subcategory are created

Basic Flow:		
Assumptions: Category and subcategory have been created for the attributes. The administrator is logged in.		
Line	System Actor Action	System Response
1	The administrator clicks on the edit attributes in the admin panel.	The system directs the admin to the edit attributes page. The page has a list of checkboxes for subcategories and a confirm button.
2	The admin selects the appropriate subcategory and confirms.	The system directs the admin to the next screen with a list of attributes to that category. Each attribute has a checkbox next to it. There is a prompt below the list asking what attribute should be added with a confirm and delete button below the prompt.
3	Admin types in an appropriate attribute and confirms.	The system adds the new attribute to the subcategory.
Post Condition:		New attribute is created for the subcategory

Exception Flow (EF1): Adding an invalid attribute		
If at line 3 the admin types in an invalid attribute, then the system reloads the page with an error message.		
Line	System Actor Action	System Response
1	Admin types in an invalid attribute	System reloads the page with an error message saying that the attribute is invalid
The use case restarts at line 3		
Post Condition:		

Business Rules:	
1	Only an administrator can add an item attribute

2 Functional Requirements 2.2.1.8

Non Functional Requirements:

- | |
|-------------------------------------|
| 1 Non Functional Requirements 2.3.4 |
| 2 Non Functional Requirements 2.3.5 |
| 3 Non Functional Requirements 2.3.6 |
| 4 Non Functional Requirements 2.3.7 |

Screen Entry Exception Table:

Field	Constraint	Response
1 Attribute Prompt	String value	"Error, invalid input"

2.5.2.2.7. Delete Item Attribute

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/8/14

Brief Description:	The administrator goes and delete an unnecessary attribute
Business Trigger:	Need to delete an unnecessary attribute
Preconditions:	Attribute exists and administrator is logged in.

Basic Flow:		
Assumptions: Attribute already exists and the administrator is logged in		
Line	System Actor Action	System Response
1	The administrator clicks on the edit attributes in the admin panel.	The system directs the admin to the edit attributes page. The page has a list of checkboxes for subcategories and a confirm button.
2	The admin selects the appropriate subcategory and confirms.	The system directs the admin to the next screen with a list of attributes to that category. Each attribute has a checkbox next to it. There is a prompt below the list asking what attribute should be added with a confirm and delete button below the prompt.
3	The admin clicks the appropriate checkbox denoting which attribute to delete and presses the delete button.	The system removes the attribute from the subcategory.
Post Condition:		The attribute is removed from the subcategory.

Business Rules:	
1	Administrator must be logged in
2	Functional Requirement 2.2.1.9

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

2.5.2.3. Auctions

2.5.2.3.1. Create Auction Item

Version	Description	Changed By	Date
1.0	First draft	Sahar Voghoei	9/7/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The purpose of the use case is to an item for sell.
Business Trigger:	A user decides to sell an item in this system and click on sell bottom.
Preconditions:	There should be a category for this item and user should be a signed in member.

Basic Flow:		
Assumptions: User have been signed in to the system and have an item to sell.		
Line	System Actor Action	System Response
1	User click on “Sell” bottom at the menu bar.	The system displays a page for searching a category. This page has a list of categories and cancel bottom.
2	User select a category from the list	System will display a sub category of the selected category in the other list box
3	User select a product	System add the continue bottom to the displayed page
4	User click on “Continue” bottom	System displays a page for user to describe the item. This page has Title, Condition, photo, and some specific attribute that is displayed base on the category that user selected, Minimum price, Due date, and Submit and Cancel bottoms.
	User provide all information for the item and click “Submit”	System displays a new page that congratulate user to put an item for sell successfully. This page has a link to Seller “My Item” page and to the home page. There is another bottom to place another Item for sell.
Post Condition:	The new Item is saved in the system for that seller and the Item is listed in seller Item List.	

Alternate Flow (AF1): Cancel the process of adding Item
If at line 1, 2, 3 or for in Basic Flow the user click on “Cancel”, then:

Line	System Actor Action	System Response
1	User click on cancel bottom.	System goes back to the home page.
The use case terminates.		
Post Condition:		N/A

Exception Flow (EF1): User creates in correct values for field.		
If at line 5 in Basic Flow the user provide incorrect information values, then:		
Line	System Actor Action	System Response
1	User attempt to enter a value for a field.	System displays a message that “Incorrect value formats. Please try again”
The use case restarts at line 5		
Post Condition:		N/A

Business Rules:	
1	User should be a signed in before try to add an item for sell.
2	Functional Requirements 2.2.3.1

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
1 Price	Integer	Message “Please Enter integer value”
2 Dead line Date	Date At least 2 days after current date	Message “Enter an appropriate date value.”

2.5.2.3.2. Update Auction Item

Version	Description	Changed By	Date
1.0	First draft	Daniel Guzman	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	Once an auction item has been created, the seller can update the item's details.
Business Trigger:	Seller needs to update information about an item.
Preconditions:	Item must exist and seller must be logged in.

Basic Flow: Seller selects item to edit, edits the item's attributes, then submits it. The system then updates the item.		
Assumptions: Item exists and the user is the seller of the item.		
Line	System Actor Action	System Response
1	User selects item to be edited.	System sends form with item's attributes ready to edit.
2	User changes fields to update item attributes and submits them to the server.	System alters item data and notifies current bidder of changes.
Post Condition:	Item has new values for attributes and bidder is notified of changes.	

Alternate Flow (AF1): Editing attributes to invalid values		
If at line 2 of the main flow the seller submits invalid attribute values (i.e. string in a number field), then:		
Line	System Actor Action	System Response
1		System will return an error saying that the input is invalid and prompt the user to try again.
The use case restarts at line 2 in the basic flow.		
Post Condition:	System remains unchanged.	

Alternate Flow (AF2): Editing an item the user doesn't own		
If at point of the main flow the seller attempts to edit an item that they do not own, then:		
Line	System Actor Action	System Response
1		System will return an error stating that the user does not own this item and is not authorized to edit it.
The use case terminates.		

Post Condition:	System remains unchanged.
-----------------	---------------------------

Alternate Flow (AF3): Editing an item that doesn't exist		
If at point of the main flow the seller attempts to edit an item that they do not own, then:		
Line	System Actor Action	System Response
1		System will return an error stating that the item doesn't exist.
The use case resumes flow at line 1 of the basic flow.		
Post Condition:	System remains unchanged.	

Business Rules:	
1	Items can only be edited by their owners.
2	If an item has already been bid on, then the bidder should be notified of updates.
3	Changing the minimum bid isn't allowed once a bid has been made.
4	Functional Requirements 2.2.3.2
5	Functional Requirements 2.2.3.3

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
1 Quantity	Number, Read-only	Message: "You cannot change the quantity of an existing item. To do so, you must delete and recreate the item."

2.5.2.3.3. Place Bid

Version	Description	Changed By	Date
1.0	First draft	A. Stevens	9/3/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	This use case describes what happens when the user places a bid.
Business Trigger:	The user wants to bid on an item on Dawg Trades
Preconditions:	The system is running and functioning properly.

Basic Flow:		
Assumptions: User is authenticated. User is not the seller of the bid item. User's bid is a non-negative decimal. User is on the item's page.		
Line	System Actor Action	System Response
1	User clicks on the 'Place a Bid' button	System returns are input form to accept user's bid
2	User inputs bid	System validates user input
3	User submits bid	System accepts bid and updates the item's page to reflect new bid and sends an email to seller and buyer.
Post Condition:		The new bid will be shown on the item's page.

Exception Flow (EF1): Inputs invalid dollar amount		
Line	System Actor Action	System Response
1	User inputs negative dollar amount.	System attempts to validate input and returns an error message.
2	User re-enters valid dollar amount.	System validates user input. Return to step 3 of basic flow.

Business Rules:	
1	The user should not be able to enter a negative dollar amount
2	Functional Requirements 2.2.4.1
3	Functional Requirements 2.2.4.2

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
1 Bid Amount	Non-negative dollar amount	Message: Amount cannot be negative – re-enter amount”

2.5.2.3.4. Win Auction Item

Version	Description	Changed By	Date
1.0	First draft	A. Stevens	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	This use case describes what happens when a user wins an auction. This use case has a path for when the winning bid is below the seller's asking price and one for when the winning bid is above the seller's asking price.
Business Trigger:	The seller's auction deadline has been reached.
Preconditions:	The system must be on and functioning properly.

Basic Flow:		
Assumptions: The buyer placed the highest bid and it is above the seller's asking price. The seller's deadline has expired.		
Line	System Actor Action	System Response
1	Buyer placed winning bid	The system checks that the winning bid is above asking price.
2	----	The system notifies both the buyer and the seller of the winning bid and marks item as sold.
Post Condition:		The auction item is no longer available for auction on the system.

Alternate Flow (AF1): Winning bid is below asking price, but seller approves price		
If at line 1 in Basic Flow the buyer's winning bid is below the seller's asking price then		
Line	System Actor Action	System Response
1	Buyer placed winning bid below user's asking price.	The system checks that the winning bid is above asking price and notifies the seller that it is not.
2	The seller approves the bid	The system notifies both the buyer and the seller of the winning bid and marks the item as sold.
Post Condition:		The auction item is no longer available for auction on the system.

Alternate Flow (AF2): Winning bid is below asking price, but seller denies price		
If at line 1 in Basic Flow the buyer's winning bid is below the seller's asking price then		
Line	System Actor Action	System Response
1	Buyer placed winning bid below user's asking price.	The system checks that the winning bid is above asking price and notifies the seller

		that it is not.
2	The seller denies the bid	The system notifies both the buyer and the seller that the winning bid was rejected by the seller.
3	Seller choose whether or not to re-auction the item	--
Post Condition:		The auction item is not marked as sold.

Business Rules:	
1	The buyer and the seller cannot be the same person
2	If the winning bid is the below the seller's original asking price, the seller must approve the winning bid.
3	Functional Requirements 2.2.3.4
4	Functional Requirements 2.2.4.3

Non Functional Requirements:	
1	The system will close bids within half a second of the deadline ending.
2	Non Functional Requirements 2.3.4
3	Non Functional Requirements 2.3.5
4	Non Functional Requirements 2.3.6
5	Non Functional Requirements 2.3.7

2.5.2.3.5. Alert User of Outbid

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/4/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The system emails the previous bidder if someone else has outbid him on an item.
Business Trigger:	When a new user has outbid the previous user on an item.
Preconditions:	A user has bid on an item.

Basic Flow:		
Assumptions: The user gives a correct email during sign-up.		
Line	System Actor Action	System Response
1	A new user outbids the previous user on an item.	The system searches for the email address that corresponds with the previous bidder. The system sends a generic email to the previous bidder stating the item that he was outbided on and the new price.
Post Condition:		The email is successfully sent and the outbided user is notified.

Business Rules:	
1	Both users are members of the site.
2	Outbided users have a valid email address.

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

2.5.2.3.6. Delete Auction Item

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/4/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	The seller requests that an administrator deletes a current auctioned item.
Business Trigger:	The seller clicks the “Request Deletion.”
Preconditions:	The seller has an item in auction.

Basic Flow:		
Assumptions: The administrator is logged in. The seller, buyer, and administrator has a valid email address.		
Line	System Actor Action	System Response
1	On the item’s page, there’s a “Request Deletion” button. The seller can press the button to notify an administrator of the request.	The system emails all administrator about the request. The email contains a link to the item page.
2	When the administrator is on the item page. There should be a checkbox with the caption “Delete Item” and a submit button. The administrator can click the checkbox and submit.	The system directs the administrator to a new page with a warning that the item is about to be deleted. A prompt is given to the administrator (“Are you sure you want this item deleted?”) with a Yes or No button available.
3	The administrator accepts and click the Yes button.	The system removes the desired item from the database. The system emails the current highest bidder that the item is unavailable. The system emails the seller that the item has been deleted.
Post Condition:		The item is deleted. The corresponding parties are notified of what happened.

Alternate Flow (AF1): Clicked No instead of Yes button.		
In line 2, if the administrator clicks No instead of Yes, then the system redirects the administrator to the item page.		
Line	System Actor Action	System Response
1	When the administrator is on the item page. There should be a checkbox with the caption “Delete Item” and a submit button. The administrator can click the checkbox and submit.	The system directs the administrator to a new page with a warning that the item is about to be deleted. A prompt is given to the administrator (“Are you sure you want this item deleted?”) with a Yes or No

	button available.
The use case continues at line 3.	
Post Condition:	The item is deleted and the seller is notified.

Sub-Flow: Delete unbid item after auction time is over.		
Line	System Actor Action	System Response
1	Seller creates an item to be auctioned with a set amount of time.	After the time has passed without any bids, the system removes the item from the database. The system emails the seller that his item has been removed due to insufficient interest.

Business Rules:	
1	The seller only request the deletion when necessary.
2	The administrator responds in an adequate amount of time.
3	The current buyer is not upset over the loss of the item.
4	Functional Requirements 2.2.1.5

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

2.5.2.3.7. Delete Bid

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/4/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	If the bidder decides that he does not want to bid, he can ask the seller to remove the bid from the auction. The seller has the choice to accept or decline the request.
Business Trigger:	The bidder sends a request to the seller asking to remove his bid.
Preconditions:	The bidder has to have bid on an item. The bidder must also be the highest bidder at the time of request.

Basic Flow:		
Assumptions: The user is currently the highest bidder. The seller is signed in.		
Line	System Actor Action	System Response
1	User makes a bid on an item, making him the highest bidder on the auctioned item.	The system refreshes the auctioned item's page, showing a checkbox displaying a "cancel bid" and a submit button below the bidden price.
2	The user click the checkbox and hits submit.	The system refreshes the page, displaying that the request is submitted. The system then sends an email to the seller with the message that a user wants to cancel his bid.
3	Seller goes to a specified page with a link to the item's page, two checkboxes (yes and no), and a submit button. The seller selects "yes," and submits.	The system sends an email notifying the user that his bid has been cancelled. The system then replaced the highest bid with the next highest bid and that user's name. The system then emails that price's user saying that he is currently the highest bidder.
Post Condition:	The item's page is updated with the new info reflecting the price and current bidder's name.	

Alternate Flow (AF1): Selecting "no" instead of "yes"		
If at line 3, the seller selects no instead of yes. The winning bidder is notified if his request is accepted or not.		
Line	System Actor Action	System Response
1	Seller selects "no," and submits.	The system sends an email to the user that

		made the request saying that the seller has denied the request to cancel his bid.
The use case terminates.		
Post Condition:	The item's page does not reflect the change as the request to cancel the bid did not take place.	

Business Rules:	
1	Both the seller and bidder are members of the site.
2	Both the seller and bidder have valid emails address.
3	The bidder does not cry foul if the request is denied.
4	Functional Requirements 2.2.3.5

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

2.5.2.3.8. Update Item Attribute

Version	Description	Changed By	Date
1.0	First draft	Italia Tran	9/8/14

Brief Description:	A seller updating an attribute of an item
Business Trigger:	The necessity to update an attribute of an item
Preconditions:	Item is up for auction and seller is logged in

Basic Flow:		
Assumptions: The item is up for auction, the seller is logged in, and the attribute is being used		
Line	System Actor Action	System Response
1	The seller goes onto the item page. Each attribute has a field to be entered in. At the bottom is a confirm button. The seller edits and confirm the changes to the appropriate attribute.	The system refreshes the page with the appropriate changes to the attribute.
Post Condition:		The attribute of the item is updated

Exception Flow (EF1): Input wrong type of attribute		
If at line 1 the seller does an invalid input, then the system refreshes the page with an error message		
Line	System Actor Action	System Response
1	The seller gives an invalid input for the attribute field.	The system refreshes page and warns the seller of a wrong input.
The use case restarts at line 1		
Post Condition:		

Business Rules:	
1	The seller is a member of DawgTrades
2	Functional Requirements 2.2.3.7

Non Functional Requirements:	
1	Non Functional Requirements 2.3.2
2	Non Functional Requirements 2.3.4
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

Screen Entry Exception Table:

Field	Constraint	Response
1 Attribute Field	Specific to attribute	“Error, wrong input for attribute”

2.5.2.4. General

2.5.2.4.1. Search for Item

Version	Description	Changed By	Date
1.0	First draft	Daniel Guzman	9/5/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	Searching essentially allows a bidder to locate an item of some kind via search criteria rather than looking through the entirety categories manually. The item may be specific or vague (i.e. a particular model or simply a type of item).
Business Trigger:	The user arrives at the DawgTrade site with an item in mind and wishes to locate it on the site (if present).
Preconditions:	The system must be online and functioning correctly.

Basic Flow: The buyer arrives at the search page, enters their search criteria, and submits the search query. The system will then return a page containing the list of results matching the search criteria, or a page saying that no results were found.

Assumptions: Categories for items exist.

Line	System Actor Action	System Response
1	Actor arrives at search page.	System fetches information regarding what categories have been configured and presents the actor with a list of categories to search from.
2	Actor selects a category to search from.	System gets a list of attributes associated with that category. It returns them as fields for the actor to provide as search criteria.
3	Actor provides search criteria for each field that they're interested in searching with.	System searches through database for items matching the search criteria in the chosen category. It then returns the matches to the actor.

Post Condition:	The actor is returned a page listing all items matching their search and can click on any of them to proceed to view them.
-----------------	--

Alternate Flow (AF1): Searching fails to find a match		
If at line 3 in the basic flow the system fails to find any items matching the search criteria, then:		
Line	System Actor Action	System Response
1		The system will return a page stating that no matches could be found.
The use case terminates.		
Post Condition:	Actor will know there are no matching items and can adjust search criteria if necessary.	

Alternate Flow (AF2): Receiving invalid category from the actor		
If at line 2 in the basic flow the actor sends a nonexistent category selection, then:		
Line	System Actor Action	System Response
1		The system will direct the actor back to the category selection page along with an error suggesting they try selecting a category again.
The use case resumes the basic flow at line 2.		
Post Condition:	System will await input again from actor.	

Alternate Flow (AF3): Receiving invalid search criteria from the actor		
If at line 3 in the basic flow the actor sends invalid input (for example, a string in an integer field), then:		
Line	System Actor Action	System Response
1		The system will direct the actor back to the search page along with an error stating that their input was invalid.
The use case resumes the basic flow at line 3.		

Post Condition:	Actor will be able to resubmit their query, this time with correct inputs. System will NOT execute queries that are likely to return errors.
-----------------	---

Business Rules:
Valid inputs for integer data should be reasonable (i.e. no negative integers) Functional Requirements 2.2.2.8

Non Functional Requirements:
Non Functional Requirements 2.3.4 Non Functional Requirements 2.3.5 Non Functional Requirements 2.3.6 Non Functional Requirements 2.3.7

Screen Entry Exception Table:		
Field	Constraint	Response
1. Search Fields	Dependent on field	“Invalid input”

2.5.2.4.2. Inactive for 10 minutes

Version	Description	Changed By	Date
1.0	First draft	A. Stevens	9/3/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	This use case describes what happens when a user is inactive for 10 or more minutes.
Business Trigger:	N/A
Preconditions:	The system is functioning and running correctly.

Basic Flow:		
Assumptions: User is logged in to DawgTrades.com as an authenticated user.		
Line	System Actor Action	System Response
1	User is inactive on the system for 10 or more minutes.	The system logs the user out.
Post Condition:		The user is logged out of the system and must re-authenticate

Business Rules:	
1	A logged in user cannot be inactive for 10 or more minutes.

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4
2	Non Functional Requirements 2.3.5
3	Non Functional Requirements 2.3.6
4	Non Functional Requirements 2.3.7

2.5.2.4.3. Navigate as Guest

Version	Description	Changed By	Date
1.0	First draft	A. Stevens	9/3/14
2.0	Uniformed all cases	Italia Tran	9/8/14

Brief Description:	When an unauthenticated user goes to DawgTrades.com they should only see a page prompting them to sign in or sign up. This use case is meant to outline what that experience should be like.
Business Trigger:	User wants to gain access to the site
Preconditions:	The system is up and functioning properly

Basic Flow:		
Assumptions: User is not authenticated		
Line	System Actor Action	System Response
1	The user navigates to DawgTrades.com	The system recognizes that the user is not authenticated and displays the landing page.
Post Condition:	The system should not be able to be access by a user if the user is not signed in.	

Business Rules:	
1	Unauthenticated users should not be able to gain access to the system.

Non Functional Requirements:	
1	Non Functional Requirements 2.3.4

2.5.3. Use Case Model

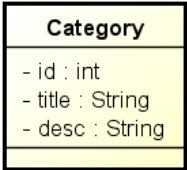


powered by Astah

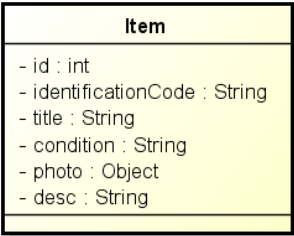
2.5.4. Domain Object Model

2.5.4.1. Real Life Objects

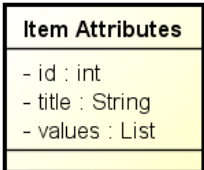
2.5.4.1.1. Category

Object	Description
 <pre>classDiagram class Category { -id : int -title : String -desc : String } </pre> <p>powered by Astah</p>	<ol style="list-style-type: none">1. Id: Unique identification of Category2. Title: Title/Name of the category3. Description: If there is any specific description for the category

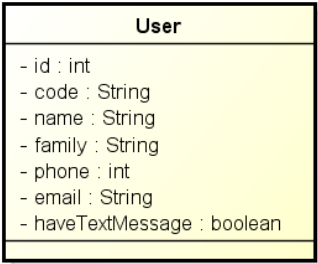
2.5.4.1.2. Item

Object	Description
 <pre>classDiagram class Item { -id : int -identificationCode : String -title : String -condition : String -photo : Object -desc : String } </pre> <p>powered by Astah</p>	<ol style="list-style-type: none">1. Id: Unique identification of Item2. Code: Unique identification code for Item3. Title: Title/Name of the Item4. Condition: the condition of product5. Photo: The image of the Item6. Description: If there is any specific description for the category

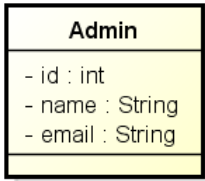
2.5.4.1.3. Item Attribute

Object	Description
 <pre>classDiagram class ItemAttributes { -id : int -title : String -values : List } </pre> <p>powered by Astah</p>	<ol style="list-style-type: none">1. Id: Unique identification of Item Attribute2. Title: Title/Name of the Item Attribute3. Values: Attribute value/ values

2.5.4.1.4. User

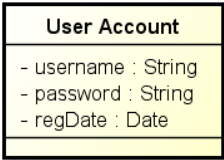
Object	Description
 <pre> classDiagram class User { -id : int -code : String -name : String -family : String -phone : int -email : String -haveTextMessage : boolean } </pre> <p>powered by Astah</p>	<ol style="list-style-type: none"> 1. Id: Unique identification of user 2. Code: Unique identification for each user 3. Name: Name of the user 4. Family: Last name of the user 5. Phone: phone number that user likes to provide 6. Email: ONLY UGA email address is valid 7. HaveTextMessage: Attribute value/values

2.5.4.1.5. Admin

Object	Description
 <pre> classDiagram class Admin { -id : int -name : String -email : String } </pre> <p>powered by Astah</p>	<ol style="list-style-type: none"> 1. Id: Unique identification of Admin 2. Name : Name of the Admin 3. Email: Admin email address

2.5.4.2. Business Objects

2.5.4.2.1. User Account

Object	Description
 <pre> classDiagram class UserAccount { -username : String -password : String -regDate : Date } </pre> <p>powered by Astah</p>	<ol style="list-style-type: none"> 1. Username: Unique identification of user account 2. Password : user password 3. RegDate: Registration date

2.5.4.2.2. Sell (info)

Object	Description
<div><div>Item Sell Info</div><div><ul style="list-style-type: none">- id : int- quantity : int- minBidPrice : int- deadline : Date- desc : String</div></div> <div>powered by Astah</div>	<ol style="list-style-type: none">1. Id: Unique identification of user account2. Quantity: The number of item for sell3. MinBidPrice: The minimum price for item sell4. Deadline: The time that is set for bid5. Desc: Description

2.5.4.2.3. Bid (info)

Object	Description
<div><div>Item Bid</div><div><ul style="list-style-type: none">- id : int- bid : int- date : Date</div></div> <div>powered by Astah</div>	<ol style="list-style-type: none">1. Id: Unique identification of bid2. Bid: The price bid by buyer3. Date: Date and time of bid

2.5.5. User Interface Sketches

Landing Page for guests

DawgTrades.com

Username:

Password:

Sign in

Don't have an account?

[Sign up!](#)

Page for resetting password

DawgTrades.com

Hi, (User) ▾

Reset Password

Enter your uga.edu email address

Continue

System emails new password to user

DawgTrades.com

Hi, (User) ▾

Reset Password

A temporary password was sent to your email

Sign in

Registration page

DawgTrades.com

Registration

First Name:

Last Name:

E-Mail:

Username:

Password:

Sign up

Account settings page

DawgTrades.com Hi, (user) ▾

Account Settings

Change Account Settings

Name: (name)

Email: (email)

Password: *****

Phone Number: (phone number)

Delete Account

Page keeping track of items

DawgTrades.com Hi, (user) ▾

My Items

My Bids {{number of bids}}

My Items for Sell {{number of items}}

Page dedicated to searching for items

DawgTrades.com Hi, (user) ▾

Search

Filters

(Filter Option Title)

☐ (Option)

(Filter Option Title)

☐ (Option)

(Filter Option Title)

☐ (Option)

(Filter Option Title)

☐ (Option)

(Filter Option Title)

☐ (Option)

(Filter Option Title)

☐ (Option)

(Filter Option Title)

☐ (Option)

Search Results

Page for making an auction

Choose a category that your item fits best in: ✕


Collectibles >	iPads, Tablets & eBook Readers >	Apple Laptops
Computers, Tablets & Networking >	iPad/Tablet/eBook Accessories >	PC Laptops & Netbooks
Consumer Electronics >	Laptops & Netbooks >	
Crafts >	Desktops & All-In-Ones >	
Dolls & Bears >	Laptop & Desktop Accessories >	
DVDs & Movies >	Cables & Connectors >	
Entertainment Memorabilia >	Computer Components & Parts >	
Gift Cards & Coupons >	Drives, Storage & Blank Media >	
Health & Beauty >	Enterprise Networking, Servers >	
Home & Garden >	Home Networking & Connectivity >	
Jewelry & Watches >	Keyboards, Mice & Pointing >	


Selected category:
Computers/Tablets & Networking > Laptops & Netbooks > Apple Laptops


Close Continue


Computers/Tablets & Networking > Laptops & Netbooks > Apple Laptops

Describe your item

* Title 


* Condition 


* Photos 
Add up to 12 photos for free.
[Add photos](#)

Item specifics 


More item specifics (recommended)


Brand Apple	Product Family iBook
Release Year 2013	MPN
Screen Size 12.1"	Processor Type Intel Core Duo
Processor Speed 1.33 GHz	Memory 1 GB
Hard Drive Capacity 128 GB	Operating System Mac OS 8
Country/Region of Manufacture Afghanistan	


* Details 


Times New Roman 13 B I U  [Switch to HTML editor](#)

Select format and price

* List as 

* Starting price 

☐ Add a Buy It Now option so buyers can get it without bidding 

* Listing duration 


☐ Schedule your listing's start time

List it

Modify and delete Item

DawgTrades.com

Hi, {User}

		Views / Watchers	Bids	Price	Time left	Actions
	sass (141400282699)	3 0	0	\$10.00 Calculate	6d 23h	Sell similar More actions
<div><div>Add to list</div><div>End</div><div>Edit</div><div>Delete</div></div>						

Ending the auction early

DawgTrades.com Hi, {User} ▾

End My Auction Early

Select a reason for ending your auction early.

☐ (Reason)

☐ (Reason)

☐ (Reason)

☐ (Reason)


End Auction

Bid Page

DawgTrades.com Hi, {User} ▾

{Auction Item} {Seller}

Place A Bid



{Auction Item}

(Item Description -- item desc, min. price, condition)

Bids

Current Highest Bid: {Username} - {bid} - {timestamp}

Previous Bids

{Username} - {bid} - {timestamp}

{Username} - {bid} - {timestamp}

{Username} - {bid} - {timestamp}

{Username} - {bid} - {timestamp}

Navigation Toolbar of site

DawgTrades.com Hi, {User} ▾

Account Settings

My Items

Sign Out

3. Glossary

Administrator: A user who has access to the entire site, often called an admin. The admin takes care of the site, making sure it performs up to task with the other types of users.

Buyer: A user that wins an auction hosted by a seller.

Constraint: Something that hinders the system from being fully optimized.

Domain Object Model: An easier way to represents objects of a system by categorizing what needed in an object.

Functional Requirement: Set of requirements that define what the system and its users are capable of doing.

Guest: An unregistered user that is using the site.

Nonfunctional Requirement: A set of criteria that is used to judge how well the system is running.

Scenario: A scene with multiple actors using different use cases to complete an objective.

Seller: A user that creates an auction for an item he wants to sell.

System: The code that runs most of the site. It automates things more easily for the administrator to handle.

Use Case: A series of steps handled by a user and the system. More than one users can use a single use case.

User: A person who is a registered member of DawgTrades.

User Interface: The design of the site that the user sees when browsing.