

Team Members: Austin Blanchard, Richard Hutcheson, Noah Lambaria, Joshua McKone

Team Leader: Richard Hutcheson

Project: Bear Market

Project Vision:

The vision for the project is to create a marketplace simulation that allows users to create and log into a marketplace account. Within their account, the user is able to browse and search a catalog of items other users are selling. The user is able to view item details and their price, and then purchase the item in whatever quantity desired and available. The user is also able to create market postings of their own to sell items and have those items be purchased by other users.

Issue Tracking Site: <https://github.com/Richard-Hutch/Marketplace-System/issues>

Website: <https://richard-hutch.github.io/BearMarket/>

Git Link: <https://github.com/Richard-Hutch/Marketplace-System>

Team Member	Use Case Responsibility
Austin Blanchard	create, edit, delete market posting, generate review
Richard Hutcheson	product table, featured items, browse reviews, product descriptions
Noah Lambaria	create account, login, edit account,
Josh McKone	purchase history, Items Being Sold, purchase item

Time Tracker

Team Member	Hours Worked
Richard Hutcheson	94
Austin Blanchard	86
Noah Lambaria	92
Josh McKone	87

Bear Market Project Plan

Apr 27, 2021

<https://richard-hutch.github.io/Marketplace-System/>

Project manager

Richard Hutcheson

Project dates

Feb 22, 2021 - Apr 26, 2021

Completion

0%

Tasks

23

Resources

4

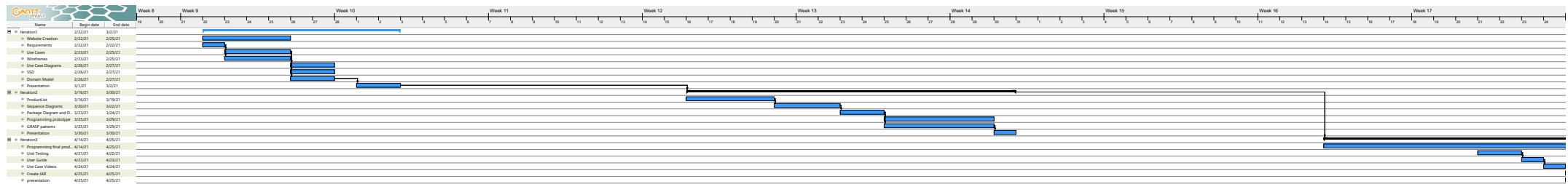
Tasks

Name	Begin date	End date
Iteration1	2/22/21	3/2/21
Website Creation	2/22/21	2/25/21
Requirements	2/22/21	2/22/21
Use Cases	2/23/21	2/25/21
Wireframes	2/23/21	2/25/21
Use Case Diagrams	2/26/21	2/27/21
SSD	2/26/21	2/27/21
Domain Model	2/26/21	2/27/21
Presentation	3/1/21	3/2/21
Iteration2	3/16/21	3/30/21
ProductList	3/16/21	3/19/21
Sequence Diagrams	3/20/21	3/22/21
Package Diagram and Design Model	3/23/21	3/24/21
Programming prototype	3/25/21	3/29/21
GRASP patterns	3/25/21	3/29/21
Presentation	3/30/21	3/30/21
Iteration3	4/14/21	4/25/21
Programming final product	4/14/21	4/25/21
Unit Testing	4/21/21	4/22/21
User Guide	4/23/21	4/23/21
Use Case Videos	4/24/21	4/24/21
Create JAR	4/25/21	4/25/21
presentation	4/25/21	4/25/21

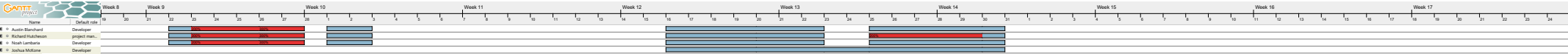
Resources

Name	Default role
Austin Blanchard	Developer
Richard Hutcheson	project manager
Noah Lambaria	Developer
Joshua McKone	Developer

4



Resources Chart



Closed Issues Tab on GitHub

Richard-Hutch / BearMarket

Unwatch

2

Unstar

2

Fork

0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

...

Filters

is:issue is:closed

Labels 9

Milestones 0

New issue

Clear current search query, filters, and sorts

0 Open

✓ 42 Closed

☐

Author

☐

Label

☐

Projects

☐

Milestones

☐

Assignee

☐

Sort

☐

Adding market listing to currently selling table. bug

#42 by Josh-M7 was closed 3 days ago

1

☐

Application did not close bug

#41 by Josh-M7 was closed 3 days ago

1

☐

Delete Currently Selling does not update currently selling table. bug

#40 by Josh-M7 was closed 3 days ago

1

☐

Incorrect pricing in table bug

#39 by Richard-Hutch was closed 3 days ago

☐

Can't login unless there is a purchase history/currently selling items on account bug

#38 by Nitsua365 was closed 3 days ago

1

☐

currently selling window does not show cents or decimal

#37 by Nitsua365 was closed 2 days ago

☐

Name to ID Map for products enhancement

#36 by Josh-M7 was closed 2 days ago

☐

Account Information class update file name. bug

#35 by Josh-M7 was closed 3 days ago

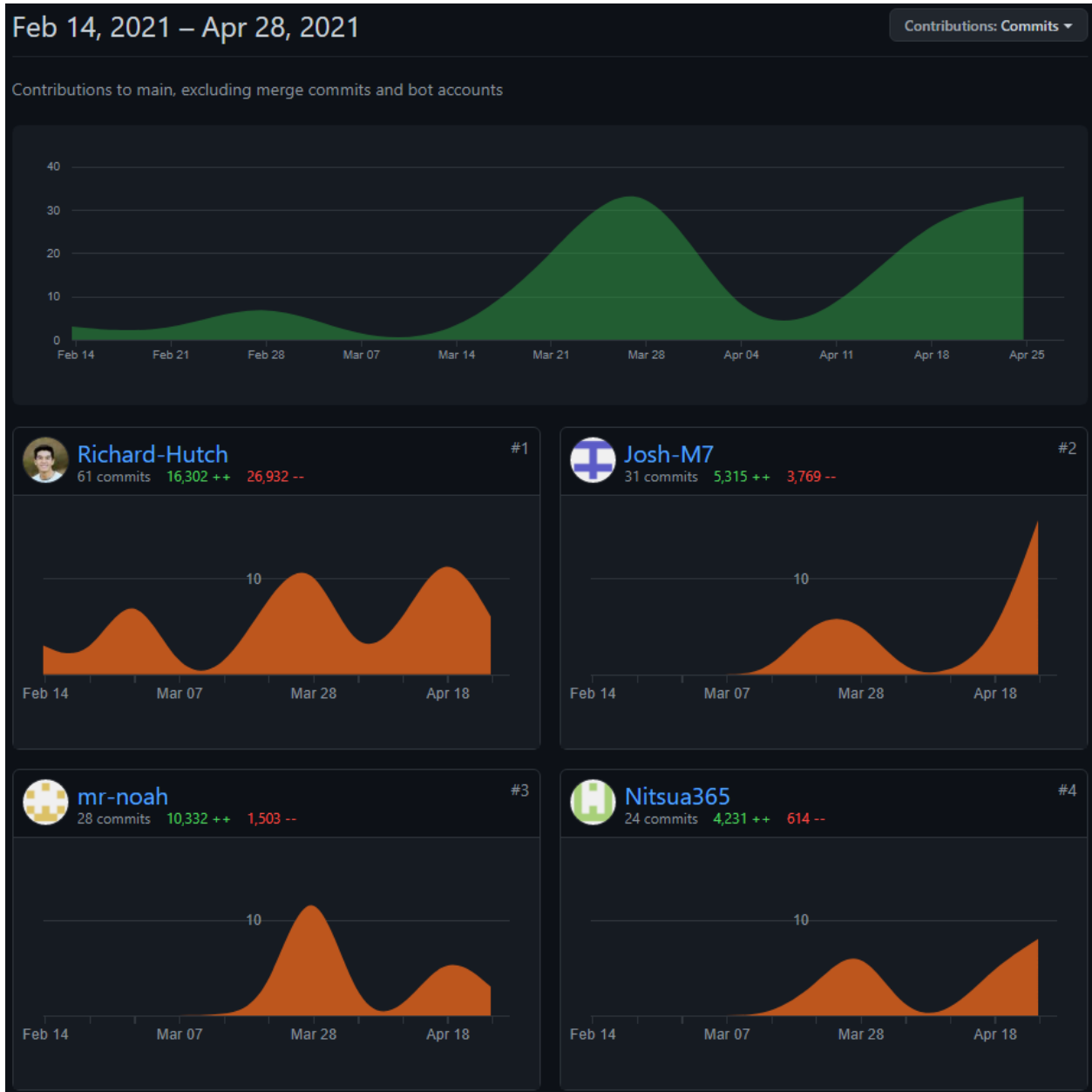
1

☐

Save new Product Table to file bug enhancement

1

Repo Analytics



User guide for Bear Market Application

1. Upon running the application, you are greeted with the log-in screen. If an account has already been made, then you simply need to log in with the correct username and password. (Note: To run the jar file, make sure that the included "src" folder is located in the same folder as the jar file otherwise the program will NOT work. Warning: Mac has potential problems associated with some of the java features.)
2. If an account has not been made, then you will need to create one (Be sure to fill out all of the field. Program will tell you what fields you are missing or if the username is taken.
3. After logging in, you are met with the main application with the product table and now have many options available to do.

4. Potential options: In the top left are the menu options. Here one can view their purchase history, currently selling list, or edit their account. There is a menu that allows one to create a new market posting. There is also the button to exit the program (Note: This is the only way to exit the program. The X button on the right will not close the program.) In the table there are buttons for the Reviews, Description, Purchase and Featured Items section. More details for each of these options will follow.
5. Purchasing an Item – Every product has a purchase button. Upon clicking it, it will show a dialog confirming that the purchase has been made. Purchases will now be in the user's purchase history.
6. Review Button – The review button will pull up the list of reviews for the product. The layout of each review is username, rating and comment.
7. Description Button – This will pull up a window that will display the product's description.
8. Featured Items – Upon clicking the button, a window will open up that will show the details of the product. In this window there is a button for reviews and purchase. Both of which behave the same as the ones in the main table do.
9. Create Market Posting – This option will pull up a form that will allow the user to add a market posting to the store front. This will add it to the user's currently selling window where they may edit or delete it.
10. Edit account – This will pull up a form that looks very similar to the create account form except all of the values have been filled in. Upon saving it will update the user's account information. (Note: if you change the password it will take effect the next time you log in.)
11. View Purchase History – This will pull up a window that will display the user's purchase history. In this window there is the option to close it or switch to the user's currently selling history.
12. View Currently Selling – This will pull up a window that will display the user's currently selling postings. If the user has any currently selling items, they have the option to edit or delete the posting (These are the buttons located on each item).
13. Edit market posting – Upon clicking this button it will pull up a form that will allow the user to edit the market posting. Upon clicking save, it will update the user's currently selling page and the market page itself.
14. Delete market posting – Upon clicking this button, it will ask the user for confirmation. If they confirm, then it will delete the posting from the store front and the user's currently selling page. (Note: After accepting, there is no way to undo this.)
15. Exit – Finally, exiting the program will close the application and save all of the changes to the files. This save will allow the state of the market to continue through multiple runnings.

All the JavaDoc is in the downloadable .zip on the BearMarket website as well as on GitHub

OVERVIEW	PACKAGE	CLASS	USE	TREE	DEPRECATED	INDEX	HELP
SEARCH: <input type="text" value="Search"/>							

All Classes

Class Summary	
Class	Description
Account	This object holds an account for the user.
AccountInformation	Data Object class that holds all pertinent information regarding the current user
BearMarket	This class initiates the startup of the program
CreateAccount	This class implements the create account window
CreateMarketPostWindow	CreateMarketPostWindow - This class is responsible for the functionality and design of the Create Market Post Window.
CSPProduct	Special Product class for the currently selling items.
CSReadFile	Helper function to read in the user's currently selling information from the file.
CSTable	Table that holds the information of the Currently Selling account information
CurrentlySellingWindow	Window class for the currently selling information for the current user
DeletePostWindow	DeletePostWindow - This class is responsible for the prompting of deleting a user market posting from the currently selling window
EditAccount	This class implements the edit account window
EditMarketPostWindow	EditMarketPostWindow - This class is responsible for editing a market posting that is currently selling to the user - It will prompt the user to change the attributes of their product and will update the currently selling menu and the main table accordingly
FeaturedItemsDialog	Create Dialog for Featured items from Product table and handle user interaction with given featured item
LogicTests	
LoginButton	This class acts as a util for the logging in process of BearMarket
LoginMenu	This class implements the entire login screen
MainScreen	Class is the controller for the core actions of the program.
PReadFile	Helper class to read in a file for the product table.
Product	This class represents a product item and contains the data associated with it name, category, description, quantity, ID, rating, price, discounted price, reviews It implements ActionListener to handle user input for the Swing buttons, description, purchase, and reviews
ProductReviewCombiner	

Sample of what our team's logging looks like:

```
1 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.MainScreenFiles.MainScreen createAndShowGUI
2 INFO: Main Screen dialog called and Account obj passed through
3 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.MainScreenFiles.MainScreen <init>
4 INFO: Starting main screen dialog
5 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.MainScreenFiles.MainScreen <init>
6 INFO: Created Menu Bar
7 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.MainScreenFiles.ProductTable createTable
8 INFO: Creating Product Table Function Called
9 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.MainScreenFiles.ProductTable createTable
10 INFO: Table and Table Model created
11 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.MainScreenFiles.ProductTable createTable
12 INFO: Reading in Product file
13 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.ReadProductFile readFile
14 INFO: Product readFile function called
15 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.ReadProductFile readFile
16 INFO: attempting to open and read Product file
17 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
18 INFO: Product constructor called, parsing data and assigning attributes
19 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
20 INFO: Product constructor called, parsing data and assigning attributes
21 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
22 INFO: Product constructor called, parsing data and assigning attributes
23 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
24 INFO: Product constructor called, parsing data and assigning attributes
25 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
26 INFO: Product constructor called, parsing data and assigning attributes
27 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
28 INFO: Product constructor called, parsing data and assigning attributes
29 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
30 INFO: Product constructor called, parsing data and assigning attributes
31 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
32 INFO: Product constructor called, parsing data and assigning attributes
33 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
34 INFO: Product constructor called, parsing data and assigning attributes
35 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
36 INFO: Product constructor called, parsing data and assigning attributes
37 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
38 INFO: Product constructor called, parsing data and assigning attributes
39 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
40 INFO: Product constructor called, parsing data and assigning attributes
41 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
42 INFO: Product constructor called, parsing data and assigning attributes
43 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
44 INFO: Product constructor called, parsing data and assigning attributes
45 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
46 INFO: Product constructor called, parsing data and assigning attributes
47 Apr 21, 2021 11:46:07 PM csi3471.bearMarket.ProductFiles.Product <init>
```