

SW Engineering CSC648/848 Fall 2019

Team 14 / Global Team

Milestone 4

Gator Trader:

Beta Launch, QA and Usability Testing and Final

Team Members: Shubham Gupta (TEAM LEAD/GITHUB MASTER)

Email - sguptadrums@gmail.com

Suraj Mondem (BACK END LEAD)

Gregory Han (FRONT END LEAD/FULL STACK DEVELOPER)

Michael Remmert (BACK END DEVELOPER)

Date Submitted: 12/12/19

Date Revised:

Product Summary

Gator Trader

Unregistered Users:

1. Users shall be able to have a sort of directory available for users to navigate.
2. Users shall be able to click sell Item button to easily start selling an item.
3. Users shall be able to do lazy registration (don't ask for login until last steps).
4. Users shall be able to search book by class at SF state.
5. Users shall be able to filter by price either high to low or low to high.
6. Users shall be able to search items.

Registered Users:

7. Users shall have access to everything unregistered users have.
8. Users shall be able to log in.
9. Users shall be able to message user.
10. Users shall be able to sell items by categories. Categories would be Electronics, Furniture, Notes, Books, Miscellaneous.
11. Users shall be able to post items by categories. Categories would be Electronics, Furniture, Notes, Books, Miscellaneous.
12. Users shall be able to have a dashboard to show messages from potential buyers.

Administrator:

13. Admin shall be able to delete listings.
14. Admin shall NOT be able to edit listings
15. Admin shall be required to approve listings before going live.

All initial front end work, security, and price filtering will be done by the Fulda team. The rest of the functions as well as small responsiveness fixes and back-end will be done by SFSU team. Our product is directed specifically toward students who are enrolled or an alumni of San Francisco State University. Users will be able to search for items based on their class using our search function. We provide a service that gives SFSU students the ability to buy/sell their school items only with other SFSU students. One of the primary advantages of keeping the product exclusive, is the fact that SFSU students will only buying and selling items from other SFSU students, leaving a high probability of similar items across classes, as well as items that may be specifically relevant for SFSU students.

Usability Test Plan

Function : Upload Item

Test Objectives: Successful uploading of picture and item description to the application
Visibility of item in user dashboard.

Test background and setup:

- Login to your Device
- Open a latest Browser (for eg. Chrome, Firefox, Safari)
- Make sure your Internet Connection is working
- Type in the Specified URL in the URL address bar

Q1. This application works on my preferred browser.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Q2. The URL was easy to type.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Q3. The Website was responsive on my mobile device.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Usability Task Description:

Consider an item that you wish to sell. Find your best picture of that item. Upload the item using our application view the item in your user profile.

We measure the **effectiveness** by comparing how many tasks the testers can solve. We divide the amount of correct solved tasks by the amount of total tasks. If we do this we get the percentage.

We measure the **efficiency** by looking at how many Mouse clicks people need to solve the tasks. We will divide the amount of clicks by the fewest amount of clicks which are necessary to solve the task. This value gives us the percentage of efficiency .

Lickert Subjective Test

Q1. Uploading an Item was easy.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Q2. I could upload a picture of my item.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Q3. I could find my posted item in my dashboard.

Strongly Disagree Disagree Neutral Agree Strongly Agree

QA Test Plan

Test Objectives: Successful uploading of picture and item description to the application.
Visibility of item in user dashboard. Post isn't visible until Admin approval. You can't post an item unless you are logged in.

Hardware and Software Setup:

Login to a personal computer or mobile device, open 4 of the latest browsers(Chrome, Firefox, Opera, Safari) and navigate to this URL:

<http://ec2-18-188-166-67.us-east-2.compute.amazonaws.com:3000/>

Test#	Title	Description	input	output	result
1	Upload Picture	Upload a picture when in the post item menu and confirm that it is the correct picture.	Picture And item description of Umbrella	User should be redirected to Dashboard	PASS
2	Post Item	Ensure a successful clicking of the item.	After clicking post	Verification of picture in search function.	PASS
3	Item in Dashboard	After Item is posted, it should be visible in the User Dashboard.	Navigate to the User Dashboard selling tab.	Item in dashboard.	PASS

Code Review:

We used a declarative coding style while maintaining spatial consistency with regards to functionality on the back-end and containerization on the front-end. The application was divided into various equivalence classes based on similar functionality or shared scope. The equivalence classes were then modularized into folders. Functions were then further modularized into files based on the granularity of program scope.

Example of our Code Review:

Hey,
This is Suraj Mondem. I have implemented 2 feature and pushed the code to the Development branch on GitHub
I am attaching the code in this mail can you code review it. please find the code snippet below

thank you

CODE:

```
router.get('/', async function (req,res,next) {  
  console.log("lowest");  
  var user = await req.session.user;  
  
  //Database call for the cheapest item  
  let dbcheapestitems = await dbcheapestitem();  
  
  console.log("recent");  
  //Database call for recent item  
  let dbrecentitems = await dbrecentitem();  
  
  res.render('index', {cheapestdb : dbcheapestitems[0], recentdb : dbrecentitems[0], user : user});  
});
```



Shubham Gupta

to Suraj ▾

Hello Suraj,

The code itself looks good, but please add in a header explaining the file.

Excellent work.

...

Security:

Attempts were made at ensuring a safe user experience by attempting to encrypt user password, so that it is not stored in a human readable format on the database. Furthermore, consideration was made towards easy to implement and viable prevention of SQL injection, and cross browser site scripting. The users picture was simply stored on the publicly accessible folder of the application. Major assets included user, and user data.

List of non-functional requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). **Done**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers. **Done**

3. Selected application functions must render well on mobile devices. **Done**
4. Data shall be stored in the team's chosen database technology on the team's deployment server. **Done**
5. No more than 50 concurrent users shall be accessing the application at any time. **Done**
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **Done**
7. The language used shall be English. **Done**
8. Application shall be very easy to use and intuitive. **Done**
9. Google analytics shall be added. **Done**
10. No email clients shall be allowed. **Done**
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **Done**
12. Site security: basic best practices shall be applied (as covered in the class). **Done**
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. **Done**
14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). **Done**