

Summary of Milestone 3 ZOOM meeting review with Prof. Petkovic and plans for further development

Team number: 03

Meeting date: April 13, 2022

Summary of feedback on UI (list if items, note that details are available in ZOOM recording):

1. Prof. says it's unusual to have tabs above the search box, but it's ok.
2. We should change Login to Login/Signup.
3. 'About' button should be on the left of Post.
4. On the top of the page, we shall have the requirement of spec 17 in smaller letters.
5. We can leave copyrights on the bottom of the page.
6. Prof. did not like the frame UI, it is distracting.
7. We need to make the page responsive when shrinking the frame. It has to show on mobile.
8. Regarding the Latest posts, we should list many, not just one.
9. When clicking on search, the upper-left corner should show the number of items in the search.
10. Filter should be changed to price low to high instead of latest. Don't say filter just put price low to high on the pulldown menu.
11. The nav bar:
 - a. Should have a different color to separate from the rest of the page.
 - b. It needs to be more visible.
 - c. Can put a line to separate it from the rest of the page.
12. When clicking on an image:
 - a. It shall show product details, and should have the same effect as when the contact seller button is pressed.
 - b. Make contact seller button a positive color; e.g. Yellow like Amazon.
 - c. Need a pulldown menu from the contact seller that would prefill product details and provide a text box that says: " Please enter your contact info..."
13. When clicking on post: the order should be all in one column in the following order:
 - a. Title
 - b. Category
 - c. Price
 - d. Description
 - e. Upload file (replacing Choose file button)
14. Then the asterix on the bottom shall say Mandatory Fields, and should be on its own column.

15. Need to add a cancel button to the left of the Post Item button.
16. Under the Post Item button, display a text that says it may take up to 24 hours for the post to be approved.
17. When the user clicks Post, we shall enforce registration or login.
18. The Login text should be consistent and not alternate between "Login" and "Sign in".
19. On the signup page:
 - a. Required fields shall be on the upper right, not the bottom.
 - b. Choose one color for the buttons that perform positive actions (use green/yellow for contact seller, post, and message. Use gray for the cancel button).
20. On the tab that says "My Messages":
 - a. Call it "My Page" instead.
 - b. By clicking on it the user shall be forced to login.
 - c. It shall have two inner tabs named "Messages" and "My Posts" respectively
21. On the My Messages page,
 - a. Just show received messages and remove the "sent" and "received" tabs
 - b. The message should contain product title, date received, and text. Or make it clickable to display this info.
22. On the My Posts page:
 - a. Show product title and date it was received.
 - b. Instead of having a "View Post" button, we shall put the title of the product and make it clickable.
 - c. Have a delete button where we can delete the post.

Summary of feedback on code and architecture

1. Use a header comment with name of owner, input, output, error messages, etc.
2. Use comments on top of lines of code as well.
3. Put the name of the owner in all code that comes to the master branch.
4. There shall be up to 40 alphanumeric characters on the search.
5. Focus and the user UI not the admin UI.

Summary of feedback on github usage

1. On github page:
 - a. Set rule that only team lead and github master can merge code.
 - b. The Github structure should have a development branch and a master branch.
 - c. Establish a master branch with test commits before merging.
 - d. Commit comments are good, industry and management looks at it. Keep comments descriptive.
 - e. Name folders that contain code carefully so that we can find it easily.
 - f. There needs to be comments above the code: "input is this, output is that", etc.
 - g. There shall be a boolean set to false describing whether a post has been approved and visible to users. The admin approval process shall be flipping the boolean to true by admin to approve the post

Summary of feedback on DB

1. The database:
 - a. Passwords must be encrypted. Do it early.
 - b. Shall have a foreign key to categories, which we already have.
 - c. Pathname shall be relative.
 - d. Do not encode the pulldown menu by hand, it should be from category via foreign key.

Summary of feedback on teamwork

1. Teamwork has been efficient thus far.
2. Continue using Trello to assign tasks to team members.
3. Continue following Agile methodology by continuously evaluating our application and adapting to any necessary changes.
4. Continue enforcing Scrum by reflecting on team progress multiple times per week.

Any other comments and issues

No other comments given.

Check Point (CP) if given, DUE:

No Check Point given.

Agreed upon P1 list of features for final delivery

Anonymous Users:

- (3) Anonymous users shall be able to search for items to buy.
- (3.1) Anonymous shall also have access to advanced search functions such as sorting of results.
- (4) Anonymous users shall be able to view all the categories of posts on the website and navigate to their preferred section.
- (5) Anonymous users shall click on any post and see its details, pricing, etc.
- (5.1) The option to contact the seller shall also be visible to all users.
- (6) Anonymous users shall be able to view Gator Express's about page.
- (7) Anonymous users shall be able to view the map of SFSU for reference.
- (18) Registration form: required for anonymous users to register.
- (18.1) Shall contain name, email id.
- (18.2) Shall contain phone number as an optional field.
- (18.3) Shall contain terms and conditions.
- (19) Anonymous users shall only register using their university's email id (.edu email).
- (19.1) Anonymous users registering with any other email shall be

rejected immediately.

Registered Users:

- (8) Registered users shall be able to login to use the website and create posts.
- (9) Registered users shall be prompted to enter a password to get access to their account.
- (9.1) Registered users shall not be able to see their password echoed as they are entering it.
- (10) Forgot Password: Registered Users shall use this option to regain access to their account.
- (13) Post Creation: Registered users shall create posts from their account and make it public for buyers to see.
- (14) Registered users interested in purchasing the item from any post shall send the seller their contact information.
- (15) Sellers shall rely on the buyer to send them their contact information via one-way chat on the website.
- (16) Registered users shall create posts to sell items.
- (17) Registered users shall be able to view messages sent from other registered users.
- (17.1) However, Registered users shall not be able to respond to said messages.
- (28) Registered users shall be able to view their own posts.

Admin:

- (23) Admin shall manage the posts.
- (24) Admin shall view requests to add posts.
- (25) Admin shall be able to delete inappropriate posts.
- (26) Admin shall be required to approve all posts before they go live.

Plan of action

First, our Github master will establish a development branch, so that our main branch can function as the “Golden” branch which has tested code. This must be done immediately, so that both Front and Back-End teams can revert to the Golden branch if the application is unexpectedly broken.

The Front-End team will first focus on implementing any Functional Requirements that haven’t already been met. We can start with adding Non-Functional Req. 17 as it is as simple as just inserting text above the header. Then we will implement basic UI for any remaining pages, so that all Use Cases can be performed. We will then optimize the resizing of the application in order to meet Non-Functional Req. 3. After that, there are no serious Front-End issues that violate P1 Functional or Non-Functional Requirements, so we can implement the other suggestions made by the Professor in ascending order of implementation cost.

The Back-End team will first encrypt all passwords so as to meet Non-Functional Req. 14. Then, functionality will be added for remaining features which have not already been implemented. We will prioritize adding functionality for Use Cases whose UI has already been developed, so that QA can begin testing those Use Cases even if the entire application is not complete.