Criterion B: Design Overview

Contents

[Internal Structure 2](#_Toc116456358)

[Data Dictionary 2](#_Toc116456359)

[Class Diagrams 3](#_Toc116456360)

[UML Diagram 5](#_Toc116456361)

[Overall Structure 6](#_Toc116456362)

[Entity Relationship Diagram 6](#_Toc116456363)

[Home Page Navigation 6](#_Toc116456364)

[System Flowchart 7](#_Toc116456365)

[Screen Designs 9](#_Toc116456366)

# Internal Structure

## Data Dictionary

The tables below display all the data that will be stored by the program. The data dictionary displays the field name, data type, access, default values and indicates if the field is required to have a value.

**(1) Class: *User***

This table displays all the user information that will be stored. All values will be entered by the user whilst creating their account.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Access** | **Default Value** | **Required?** | **Unique?** |
| email | String | private | NA | Yes | Yes |
| userName | String | private | NA | Yes | Yes |
| password | String | protected | NA | Yes | No |
| passwordResetToken\* | String | protected | NA |  | Yes |
| dateCreated\* | Date | protected | NA |  | No |

*\* - System generated*

**(2) Class: *Buyer* extends *User***

This class will inherit all the fields that are a part of the user class but also contain an array, *BidItems,* that stores all the items that a buyer has bid on. Each array element will have the fields outlined in the table below. If a buyer has never bid on an item, the array will be empty.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Access** | **Default Value** | **Required?** | **Unique?** |
| itemName | String | public |  | Yes | No |
| category | String | public |  | Yes | No |
| subject | String | public |  | Yes | No |
| grade | String | public |  | Yes | No |
| vendorUserName | String | public |  | Yes | Yes |
| isbnCode | Double | public |  | No | No |
| bidDate\* | Date | private |  | Yes | No |

\* - *System Generated*

**(3) Class: *Vendors* extends *User***

This class will inherit all the fields that are a part of the user class but also contain an array, *VendorItems,* that stores all the items that a vendor has listed on the website. Each array element will have the fields outlined in the table below. If a vendor has never bid on an item, the array will be empty.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Access** | **Default Value** | **Required?** | **Unique?** |
| itemName | String | public |  | Yes | No |
| category | String | public |  | Yes | No |
| subject | String | public |  | Yes | No |
| grade | String | public |  | Yes | No |
| productCondition | String | public |  | No | No |
| productPurchaseDate | Date | public |  | No | No |
| sellingPrice | Int | public | 0 | No | No |
| dateListed\* | Date | private |  | Yes | No |
| stillAvailable | Boolean | public | Yes | Yes | No |

\* - *System Generated*

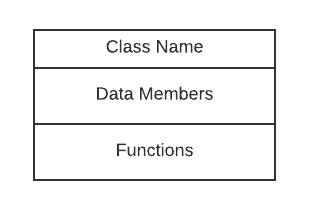
**(4) Class: *Products***

This class will store information about the products listed on CIRCLE. The fields in this class are required to categorize every product on the platform and provide the buyers will all the relevant information about the product. Fields related to the specifics about a product by the vendor will be stored in the *Vendor* class.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Access** | **Default Value** | **Required?** | **Unique?** |
| itemName | String | public |  | Yes | Yes |
| category | String | public |  | Yes | No |
| subject | String | public |  | Yes | No |
| grade | String | public |  | Yes | No |
| isbnCode | Double | public |  | Yes | No |
| imageCover | String | public |  | Yes | No |

## Class Diagrams

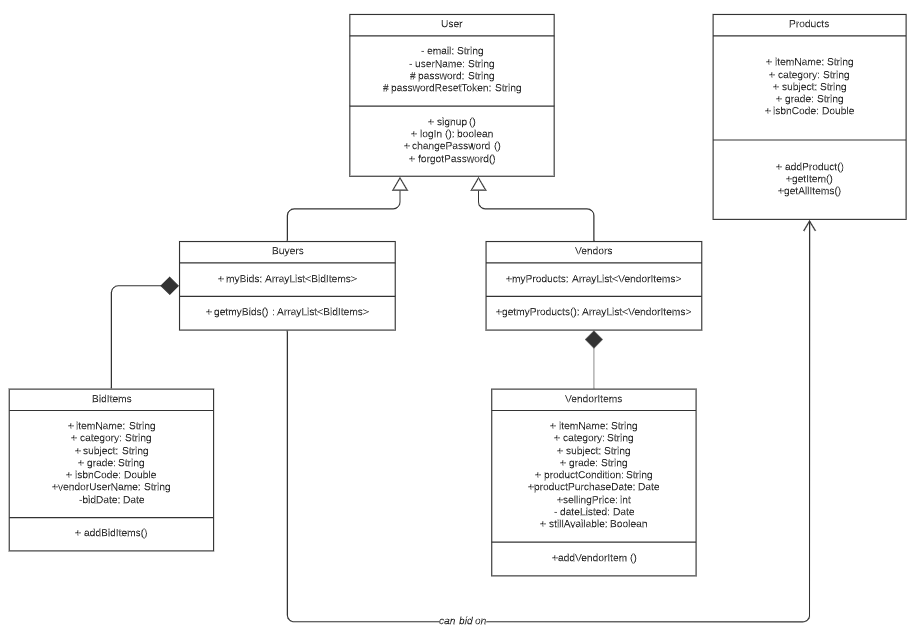
This section contains the class diagrams for the classes mentioned above. These diagrams illustrate both the data and functions that are members of these classes. As mentioned above, *Buyers* and *Vendors* inherit all the data and functions of the *User* class. The class diagram for those two classes only shows the unique elements.

**Key:**

|  |  |
| --- | --- |
|  |  |
|  |  |

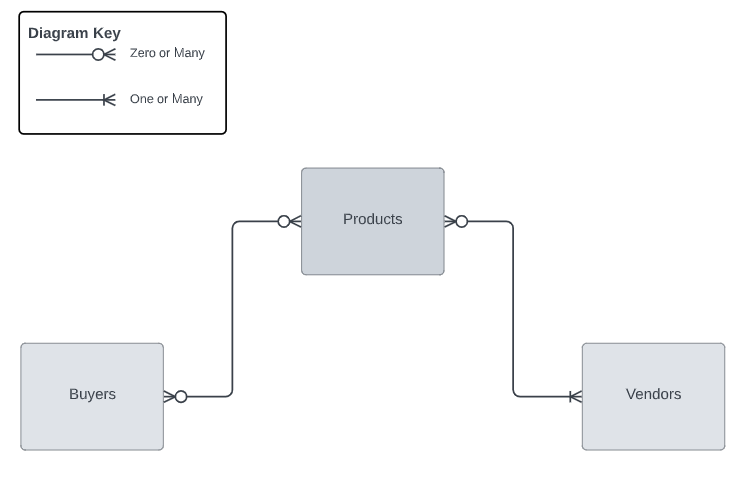
## UML Diagram

The UML Diagram shows the relationship between the different classes*.* Theempty arrow refers to inheritance which means that *Buyers* and *Vendors* inherit all the states and behaviours of *User.* The association between *Buyers* and *Products* is that the buyers can purchase the products. *Buyers* and *Vendors* have arrays, as depicted by the aggregation relationship between the two.



# Overall Structure

## Entity Relationship Diagram

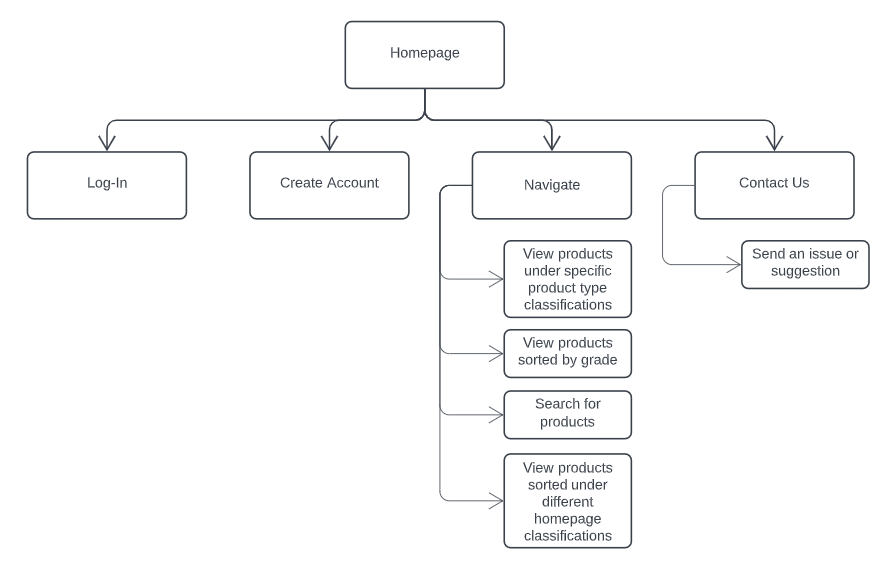


The ERD above shows the relationship between the three main entities in this solution as well as outlines the cardinality.

* *Buyers* can have zero or more P*roducts*
* *Products* can have zero or more *Buyers*
* *Products* must have one or more *Vendors*
* *Vendors* can have zero or more *Products*

## Home Page Navigation

When the user first opens the website, they will land on the home page. The diagram below illustrates the functionalities that a user can access from the home page. Users may log-in to their existing account, create a new account, navigate through products on the site or contact the CIRCLE team with issues/suggestions. These have been outlined in the diagram below.

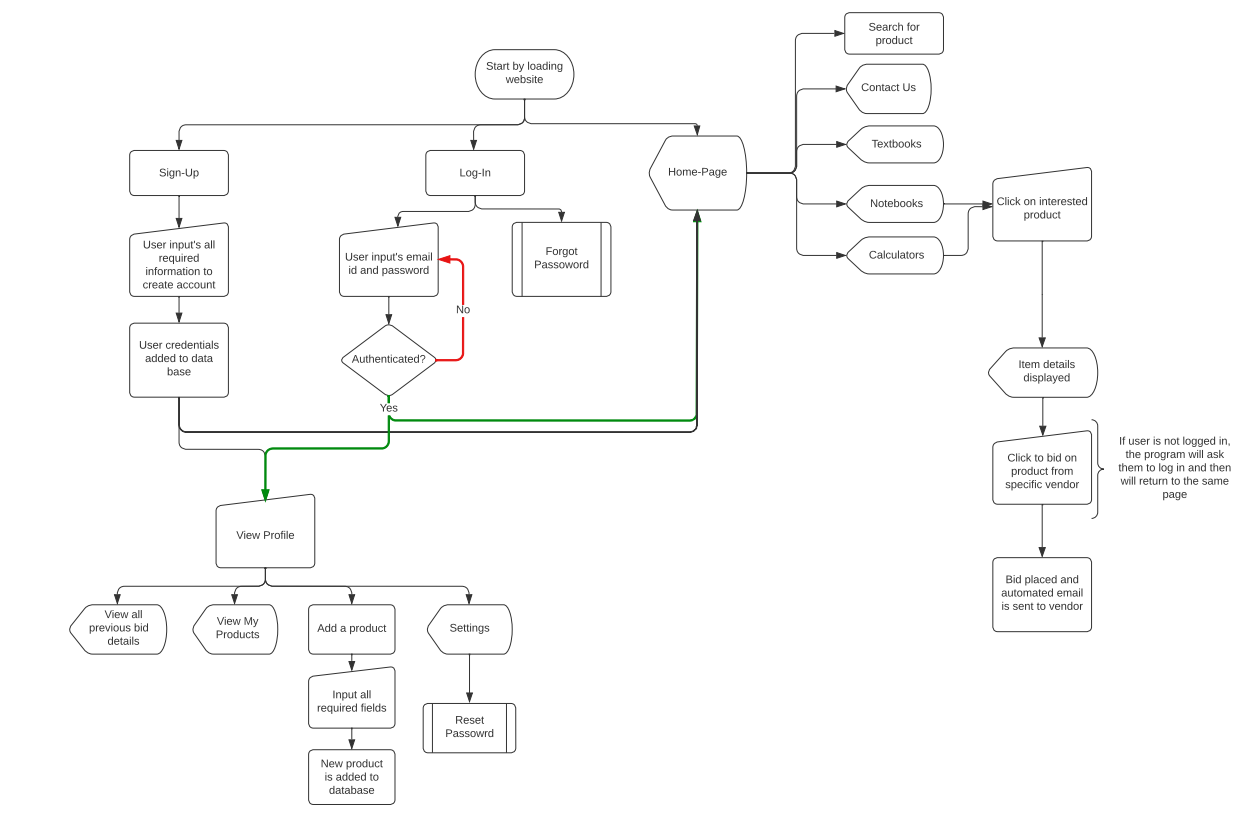


## System Flowchart

The system overview has been represented as a system flowchart.

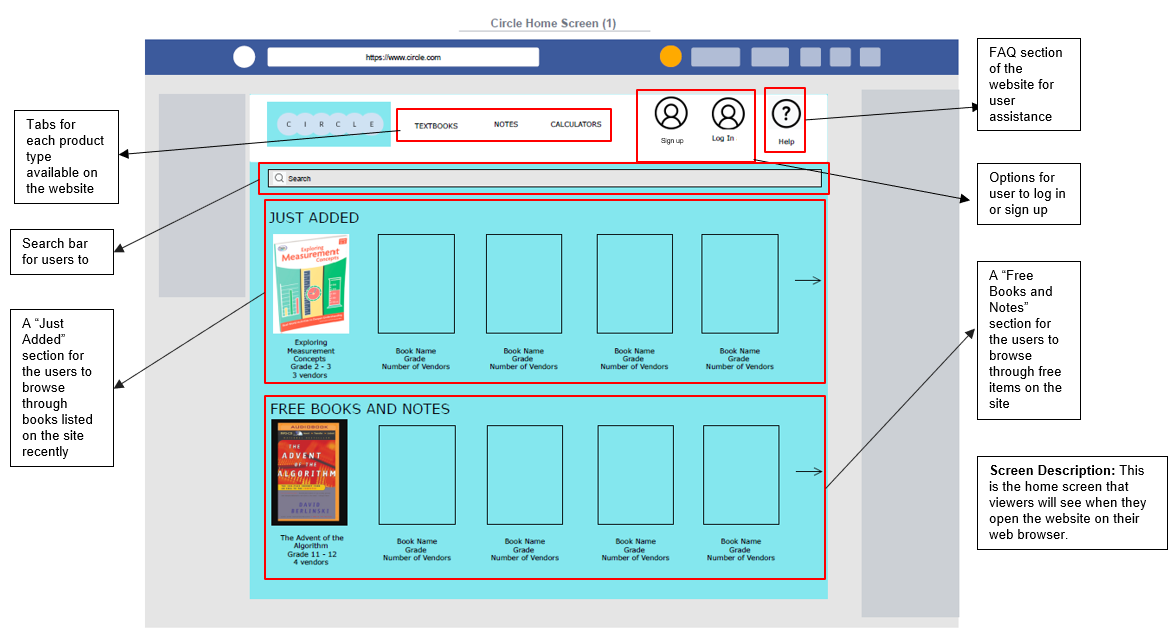
**Key:**

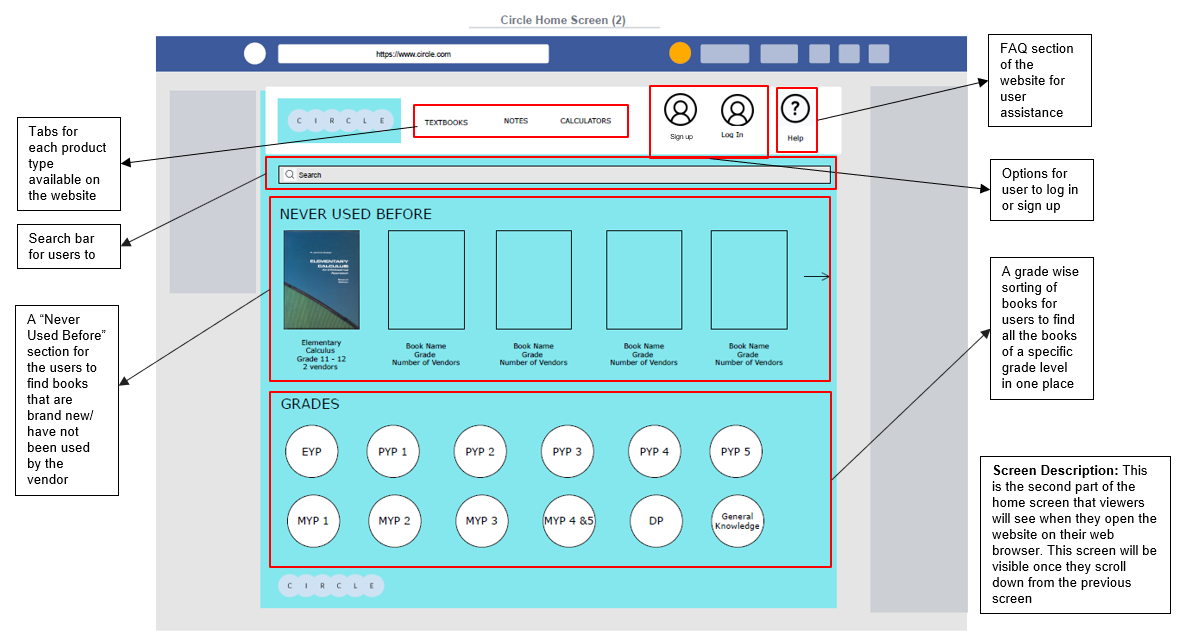
|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
|  | Executor/Terminator |
|  | Manual Input |
|  | Decision |
|  | Defined Process |
|  | Display |
|  | Process |

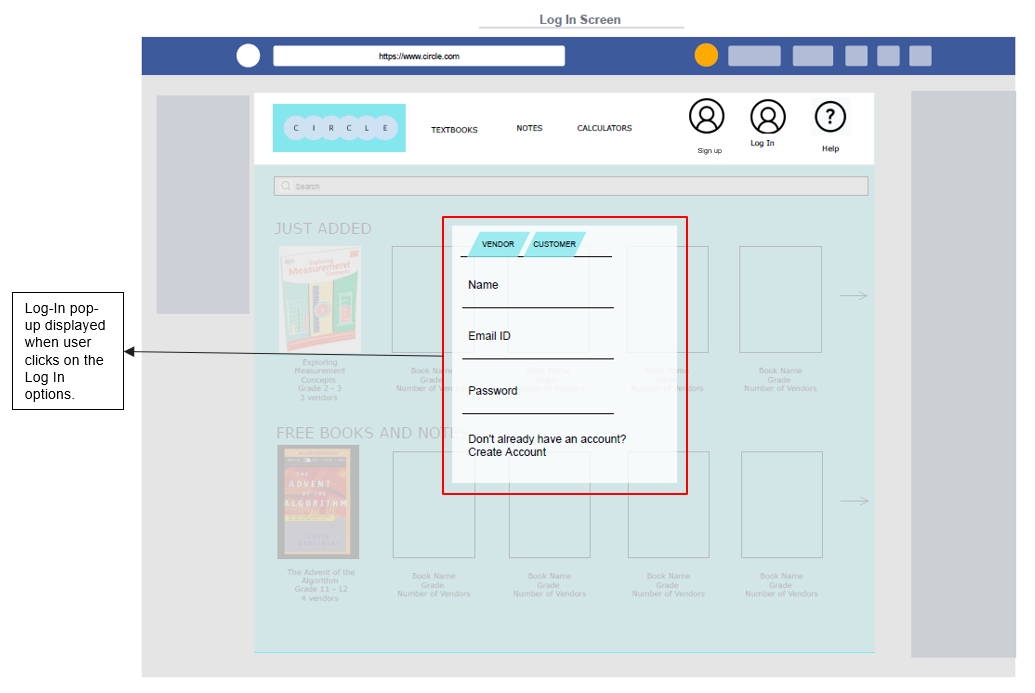


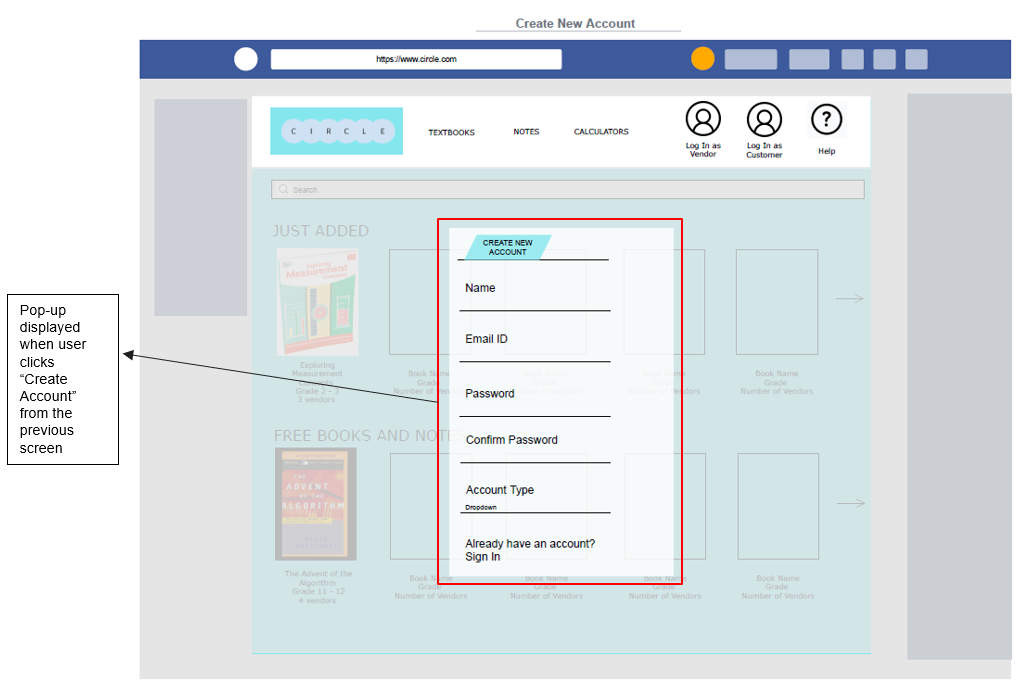
# Screen Designs

The following images show the screen designs I created for the product. Each screen a title and has been annotate to indicate its functionalities.









# Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Test No.** | **Test Case** | **Data** | **Expected Result** |
| 1 | Test of startup page to ensure users are able to browse products without logging in  Links to success criteria **1** | NA | Users are able to surf through the home page, view products and their details without logging in |
| 2 | Test of sorting algorithm on the website to be able to sort products in any category by price  Links to success criteria **9** | Test:  Sort price low to high | The free products must appear at the top of the list. Other products must be listed in ascending order of price |
| 3 | Test that program is able to categorize products into different categories based on product type  Links to success criteria **10** | Test:  Product Type: Textbooks | Only products classified as textbooks should appear in this category |
| 4 | Test that program is able to categorize products into different categories based on grade  Links to success criteria **10** | Test:  Grade: MYP5 | All products for MYP5 students (including textbooks, student notes and calculators) should appear in this category |
| 5 | Test that program is able to categorize products into different categories based on subject  Links to success criteria **10** | Test:  Subject: Physics | All physics textbooks and notes should appear in this category |
| 6 | Test for the search functionality that ensures a user can search for the product based on product name  Links to success criteria **3** | Test:  Normal: Search “*MYP Physics by Concept*”  Abnormal: Search *“MYP Physics without Concept”* | Normal case: Corresponding product must appear on user’s screen post the search.  Abnormal case: Message “Product not Found on CIRCLE” must be displayed on user’s screen. |
| 7 | Test for *Create New Account* functionality. User email must be on the school domain (i.e @oakridge.in). Email id and username must be unique  Links to success criteria **5** | Test:  Normal: Email id: firstname\_lastname @oakridge.in  Username: Firstname123  Abnormal (1): Email id: firstname\_lastname @gmail.com  Username: LastName123  Abnormal (2): Email id: firstname\_lastname @oakridge.in  Username: Firstname123 (given that this email id or username is already registered in the database) | Normal case: Account is successfully created and user is logged in  Abnormal case (1): Error message displayed indicating that official school email id must be used. User is not created.    Abnormal case (2): Error message displayed indicating that the email id or username is not unique and must be changed. User is not created. |
| 8 | Test for password in the *Create New Account* functionality. The password must be at least 8 characters long. Additionally, the *password* and *confirm password* fields must match.  Links to success criteria **5** | Test:  Normal:  Password: “abcd1234”  Confirm Password: “abcd1234”  Abnormal (1):  Password: “abc”  Confirm Password: “abc”  Abnormal (2):  Password: “abcd1234”  Confirm Password: “1234abcd” | Normal case: The password and confirm password fields are red until at least 8 characters are inputted. Once a valid password is entered, these fields turn green. User account is successfully created and user is logged in.  Abnormal case (1): The password and confirm password fields stays red. Error message indicating that password must be longer is displayed once the *signup* button is clicked. User account is not created.  Abnormal case (2): The password and confirm password fields stays red. Error message displayed stating that password and confirm password fields are not identical. User account is not created. |
| 9 | Test that automated email is sent to the vendor after bid is placed.  Links to success criteria **13** | Test: Click “bid” on a product by a specific vendor | Email should have been sent to the vendor with the bid request and the contact details of the person requesting the product |
| 10 | Test for the *edit password* functionality for the user’s accounts  Links to success criteria **14** | Test  Normal case:  Current Password: abcd1234 (as stored in the database)  New password: pqrs5678  Confirm New password: pqrs5678  Abnormal case (1):  Current Password: abcd1234 (different from database)  New password: pqrs5678  Confirm New password: pqrs5678  Abnormal case (2):  Current Password: abcd1234 (as stored in the database)  New password: pqrs5678  Confirm New password: 5678pqrs  Abnormal case (3):  Current Password: abcd1234 (as stored in the database)  New password: pqr  Confirm New password: pqr | Normal case:  The current password is verified. Password and confirm password fields are red until at least 8 characters are inputted. Once a valid password is entered, these fields turn green. User password is updated.  Abnormal case (1):  The password and confirm password fields stays red. Error message displayed stating that password and confirm password fields are not identical. User password is not updated.  Abnormal case (2): The password and confirm password fields stays red. Error message indicating that password must be longer is displayed. User password is not updated.  Abnormal case (3): The password and confirm password fields stays red. Error message indicating that password must be longer is displayed. User password is not updated. |
| 11 |  |  |  |