**CEN 4010 Principle of Software Engineering, Spring 2018**

**Final Project Portfolio**

**Perry’s Parts Pavilion Access Center**

*Team Five: Grupo Fivo*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Neil Maniquis**  *Team leader*  nmaniquis2017@fau.edu | | **Noah Leach**  *Scrum master*  nleach2013@fau.edu | | **Diego Segura**  *Team organizer*  dsegura2015@fau.edu | |
|  | **George Bechtel**  *Product owner*  gbechtel2013@fau.edu | | **Franklin Carrillo**  *Web developer*  fcarrillo2012@fau.edu | |  |

Project website: <http://lamp.eng.fau.edu/~CEN4010_S2018g05/>

|  |  |
| --- | --- |
| **Date** | **Changes** |
| 2/19/18 | Original proposal |
| 3/16/18 | Created history table, described database tables, added definitions on price |
| 3/17/18 | Added database diagram made using balsamiq, updated tables |
| 3/25/18 | Updated system architecture |
| 5/2/18 | Modified peer evaluation, added search function, changed file layout |

Date: 5/2/18

**Product summary**

Our website is called Perry’s Parts Pavilion Access Center.

Committed functions:

* List all active products: <http://lamp.eng.fau.edu/~CEN4010_S2018g05/links.php>
* Search by product name or keyword: <http://lamp.eng.fau.edu/~CEN4010_S2018g05/>
* Customer account creation
* User login
* Staff login
* Staff can manually add item information

PPPAC supports partial word search. ex searching for “ohm” will bring up all products with “ohm” in the product name or keyword.

When creating an account, there is a realtime update to show if the passwords match. This is to help insure that the password was entered without error.

After logging in, will remember user until logout or 1 hour of inactivity.

Google analytics tracks our pages.

Only logged in staff are able to add new items, customers will receive an error message if they attempt to access the page.

Customers can:

* create an account
* view items
* search items

Vendors can:

* create an account
* view items
* search items
* list items for sale

**Milestone documents**

1. Executive Summary:

Perry’s Parts Pavillion Access Center will allow students to access lab equipment and electronic parts with an account created using their Z-number. There will be a database which contains a list of electronic parts available to students in the lab. The admin (Perry) and staff have control over this database. This product would be suitable for electronics labs in other academic institutions.

1. Competitive Analysis:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Shipping | Rentals | Bidding | Item Listings | Laser Cutting/3D Printing |
| PPPAC | ✘ | ✔ | ✘ | ✔ | ✘ |
| Amazon | ✔ | ∼ | ✘ | ✔ | ✘ |
| eBay | ✔ | ✘ | ✔ | ✔ | ✘ |
| Craig’s List | ∼ | ✘ | ✘ | ✔ | ✘ |

|  |  |
| --- | --- |
|  | Legend |
| ✔ | Yes |
| ✘ | No |
| ∼ | Somewhat |

Amazon, eBay, and Craig’s List are extremely large online stores that appeal to a mainstream audience while leaving out specialized enthusiasts. Our small, lightweight platform allows us functionality that other stores do not. In addition, we have a physical location, allowing users to rent items (but not through the web store), an impossible feat to perform with a solely online store. Staff can list items for sale which can be purchased at the physical location. Our small size will allow individual items to be easier to find, since they aren’t going to be buried in thousands of other listings.

1. Data Definition:

Perry’s Parts Pavillion Access Center – The name of our product.  
Item – Products to be made available to customers.  
Shopping list – A list of intended items to later purchase.  
Check out – The collection of items about to be purchased.  
Homepage – The introductory page of the website.  
Kits – Several related items sold as a bundle for an overall reduced price.  
Z-number – The primary key used to identify student user accounts. Each student at FAU has previously had one assigned.  
EE 96 Room 205 – The location of electronics parts, tools, 3d printer, laser cutter, and desk.  
Cost – The average acquisition dollar amount for one unit. Refers to the average value of the item.  
Price – The dollar amount for one unit when selling.   
Retail – Price for individuals.  
Bulk – Price for large orders. May be used for individuals or groups

1. Overview, Scenarios and Use Cases:

Customers can:

1. visit the website

2. search for an item in different categories

3. read product information

4. select quantity of the items to be ordered

5. create an account or log in

Staff can:

1. add vendor information

2. add item information

3. create kits

4. update inventory

1. High-Level Functional Requirements

Priority Level:

1 - Must Have

2 - Desired

3 - Opportunistic as Defined in the Class

Priority: 1 A GUI for the user to use Perry’s Parts Pavillion Access Center.

Priority: 1 A database with user accounts (student, staff), a catalog of all the electronic parts available to students

Priority: 2 List of tools available for rental, and laser cutting/3d printing jobs.

Priority: 2 Ability of students to request tickets for advice/troubleshooting (mentoring invoice) and orders.

Priority: 2 Support for barcode scanner used by staff and admin for adding items to the catalog, checking out items, and distributing kits to students.

1. List of Non-Functional Requirements

1. Ability to go to any item page from the homepage within 1 minute – DONE

2. Ability to create a new user account under 5 minutes – DONE

3. Ability to add a new item for sale under 8 minutes – DONE

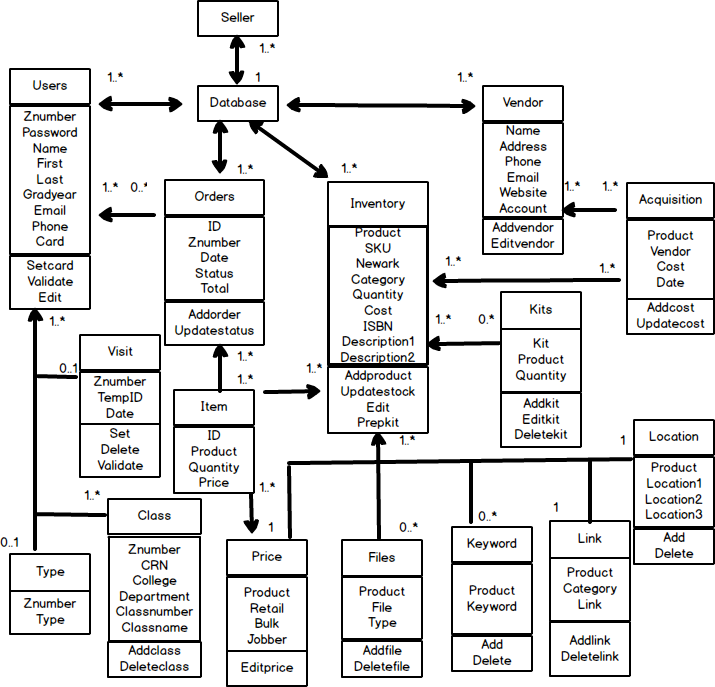
4. Ability to update the inventory of 10 items in under 5 minutes – ISSUE – Requirement is related to a non-committed function

5. Passwords are not stored in plaintext and identical passwords will be stored differently for different users – DONE

6. Usability of website for customer on a mobile device should be within 1 minute of desktop experience – DONE

1. High-Level System Architecture and Database Organization

Database class diagram



**Database tables:**

Users(Name varchar(20), First varchar(25), Last varchar(25), Znumber int(8), Email varchar(50), Password varchar(275), Phone varchar(20), Gradyear int(4), Card varchar(25), Extra1 varchar(250), Extra2 varchar(250), Extra3 varchar(250))

Type(Znumber int(8), Type varchar(10))

Visit(Znumber int(8), TempID varchar(10), Date varchar(25))

Class(CRN int(5), College varchar(5), Department varchar(5), Class varchar(5), Classnumber varchar(5), Classname varchar(50))

Order(ID int(8), Znumber int(8), Date varchar(20), Status varchar(15), Total decimal(19,4), Extra varchar(250))

Item(ID int(8), Product varchar(75), Quantity int(8), Price decimal(19,4))

Vendors(Name varchar(50), Address varchar(250), Phone varchar(20), Email varchar(50), Website varchar(50), Account varchar(50), Extra1 varchar(250), Extra2 varchar(250), Extra3 varchar(250))

Price(Product varchar(75), Retail decimal(19,4), Bulk decimal(19,4), Jobber decimal(19,4))

Inventory(Product varchar(75), SKU varchar(25), Newark varchar(25), Category varchar(25), Cost decimal(19,4), Quantity int(6), Descriptionshort varchar(350), Descriptionlong varchar(900), ISBN varchar(25), Extra1 varchar(250), Extra2 varchar(250), Extra3 varchar(250))

Location(Product varchar(75), Location1 varchar(25), Location2 varchar(25), Location3 varchar(25))

Kit(Product varchar(75), Kit varchar(25), Quantity int(8))

Files(Product varchar(75), File varchar(200), Type varchar(25))

Acquisition(Product varchar(75), Vendor varchar(25), Cost decimal(19,4), Date varchar(20))

Keyword(Product varchar(75), Keyword varchar(25))

Link(Product varchar(75), Category varchar(25), Link varchar(250))

**Media storage:** in file system

Image and PDF files will be stored in their corresponding item folder. The database will contain url for files

**Keyword search:**

Each keyword and product pair will occupy a row on the Keyword table. This allows for a variable amount of keywords for a single product.

Keyword search is done by searching the table for the matching word (SELECT Products from Keyword WHERE Keyword = keyword). The results will be used individually to find the links in the Link table.

Software/Tools:

1. Brackets

2. Putty

3. Filezilla

4. Git Gui/Git bash

5. LAMP

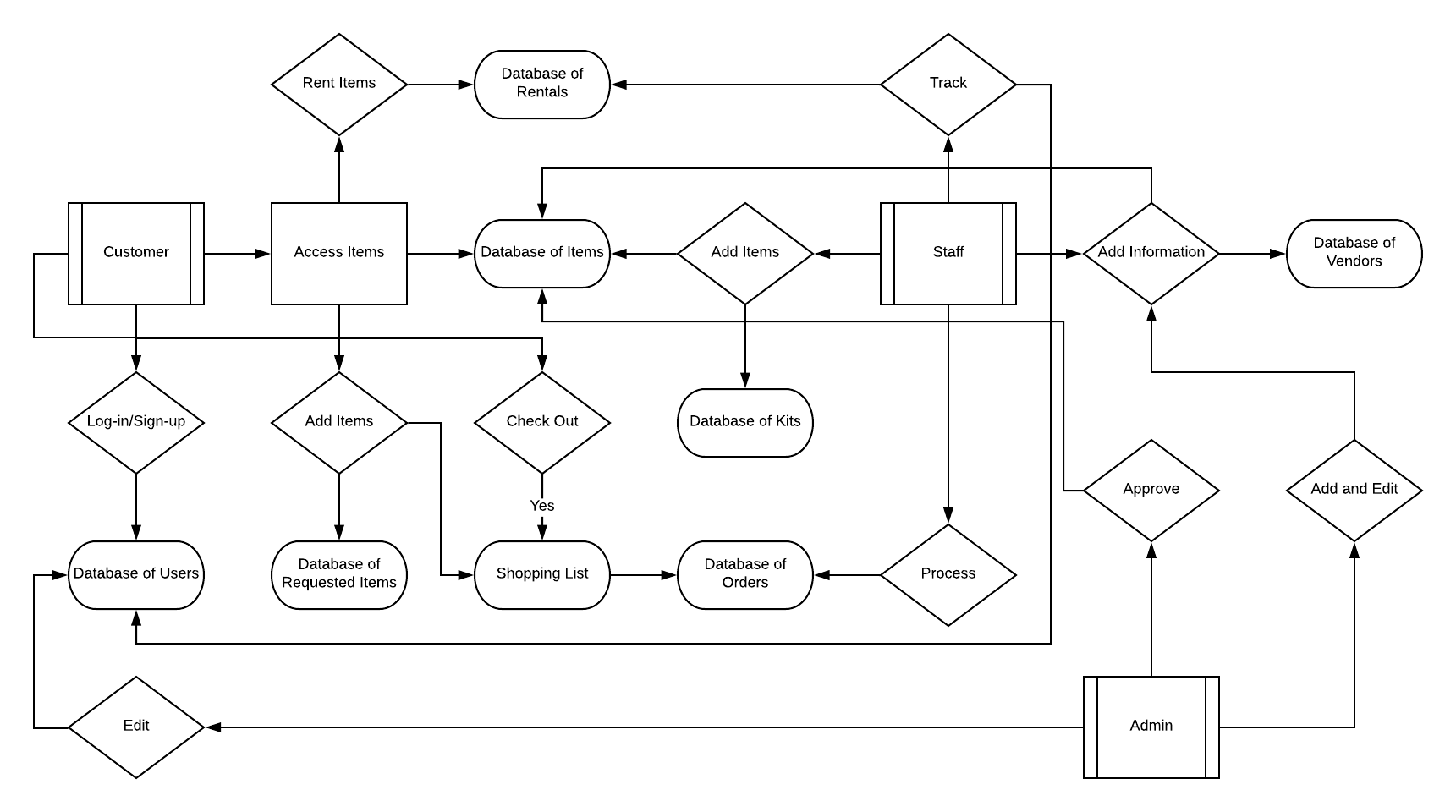
Languages:

1. HTML/CSS
2. Javascript
3. PHP
4. MySQL

Supported browsers:

1. Firefox
2. Chrome
3. Safari
4. Microsoft Edge

1. High-Level UML Diagrams



1. Key Risks

1. Skills risks: 3 out of 5 team members have no previous experience with web development. Only 1 has experience with backend.

2. Schedule risks: Our priority 1 requirements should be able to be implemented. Likely, our priority 2 requirements, and less likely, priority 3.

3. Technical risks: The CSV file has products with % in the name. This generates a URL that results in a bad request.

Item pages are created through PHP. There is a possibility for malicious code to be saved to the server.

4. Teamwork risks: All of our team members are distance learning students, meaning we do not meet physically on a daily basis.

5. Legal/content risks: We do not expect to encounter any legal or content risks.

|  |  |
| --- | --- |
| **Date** | **Changes** |
| 2/19/18 | Original proposal |
| 3/16/18 | Created history table, described database tables, added definitions on price |
| 3/17/18 | Added database diagram made using balsamiq, updated tables |
| 3/25/18 | Updated system architecture |
| 5/2/18 | Modified peer evaluation, added search function, changed file layout |

**Instructor Feedback**

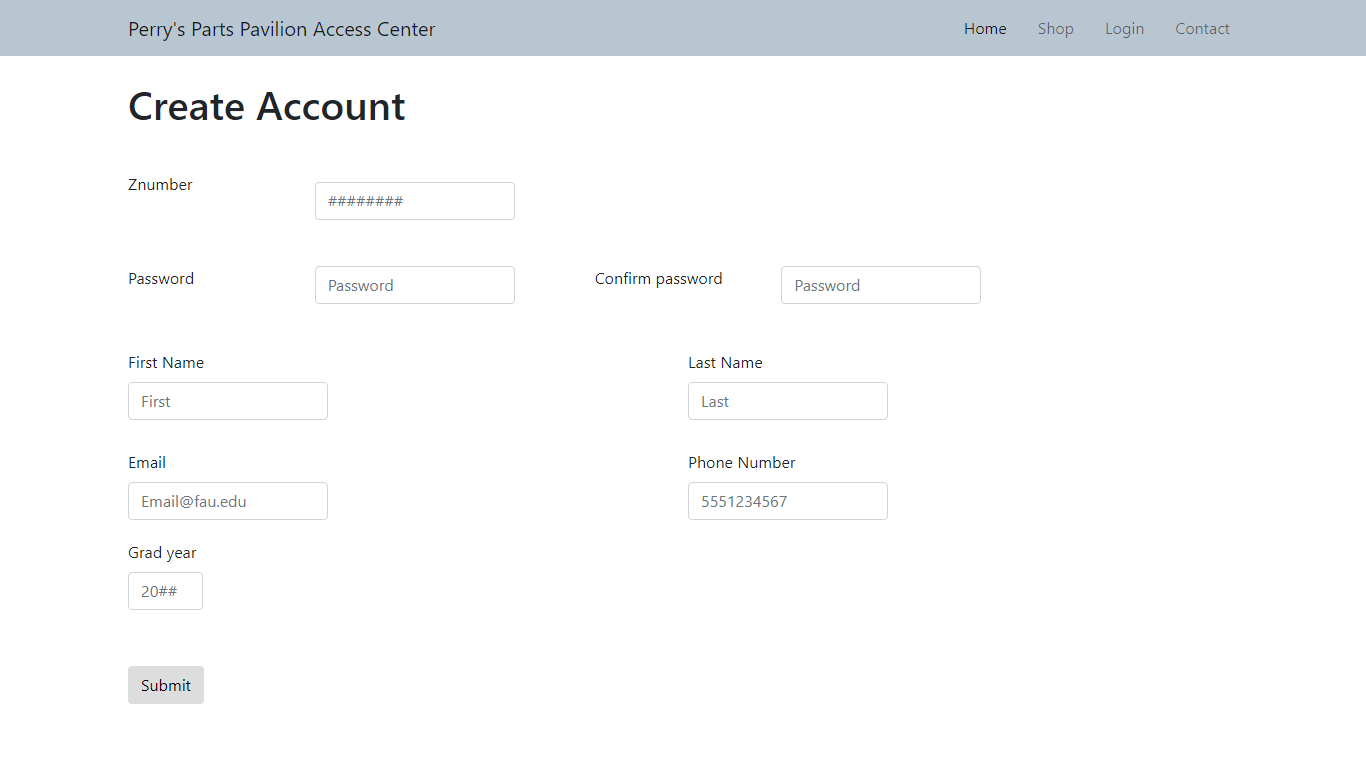
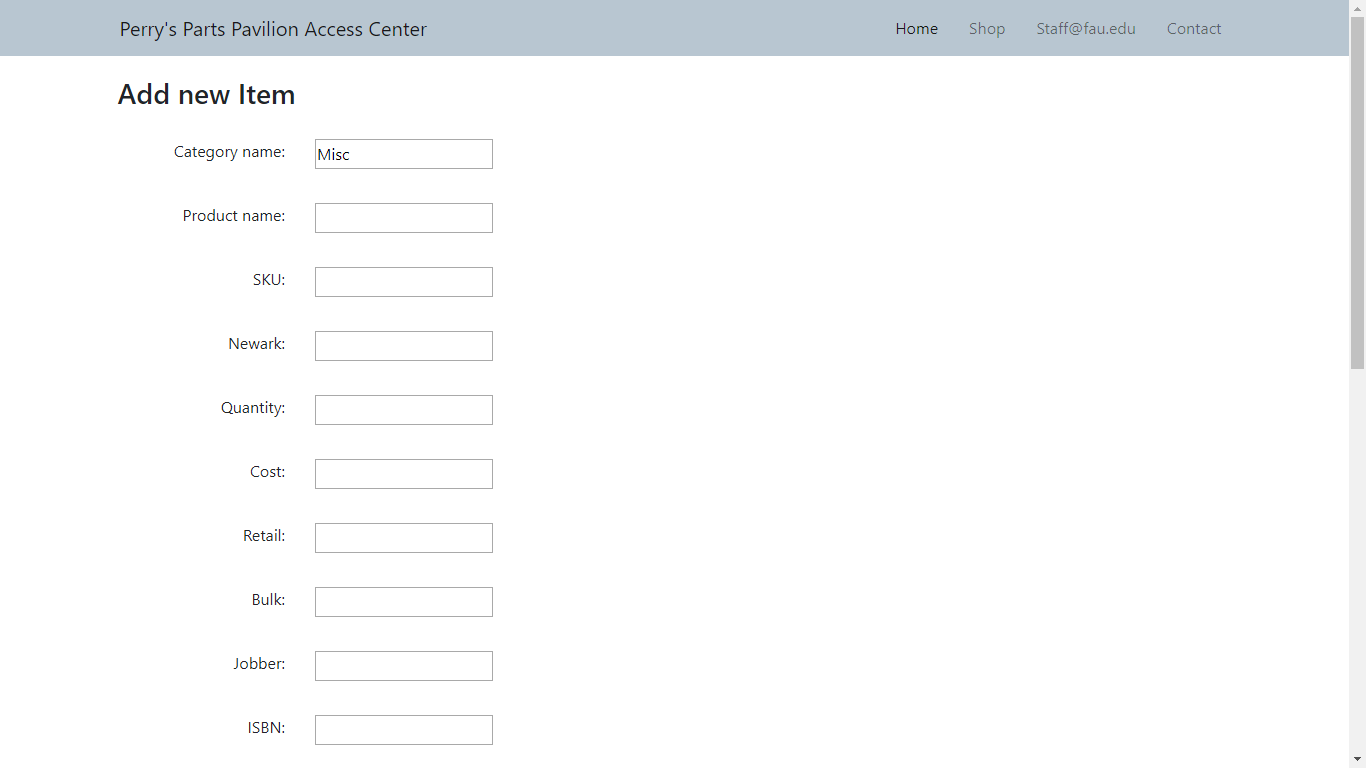
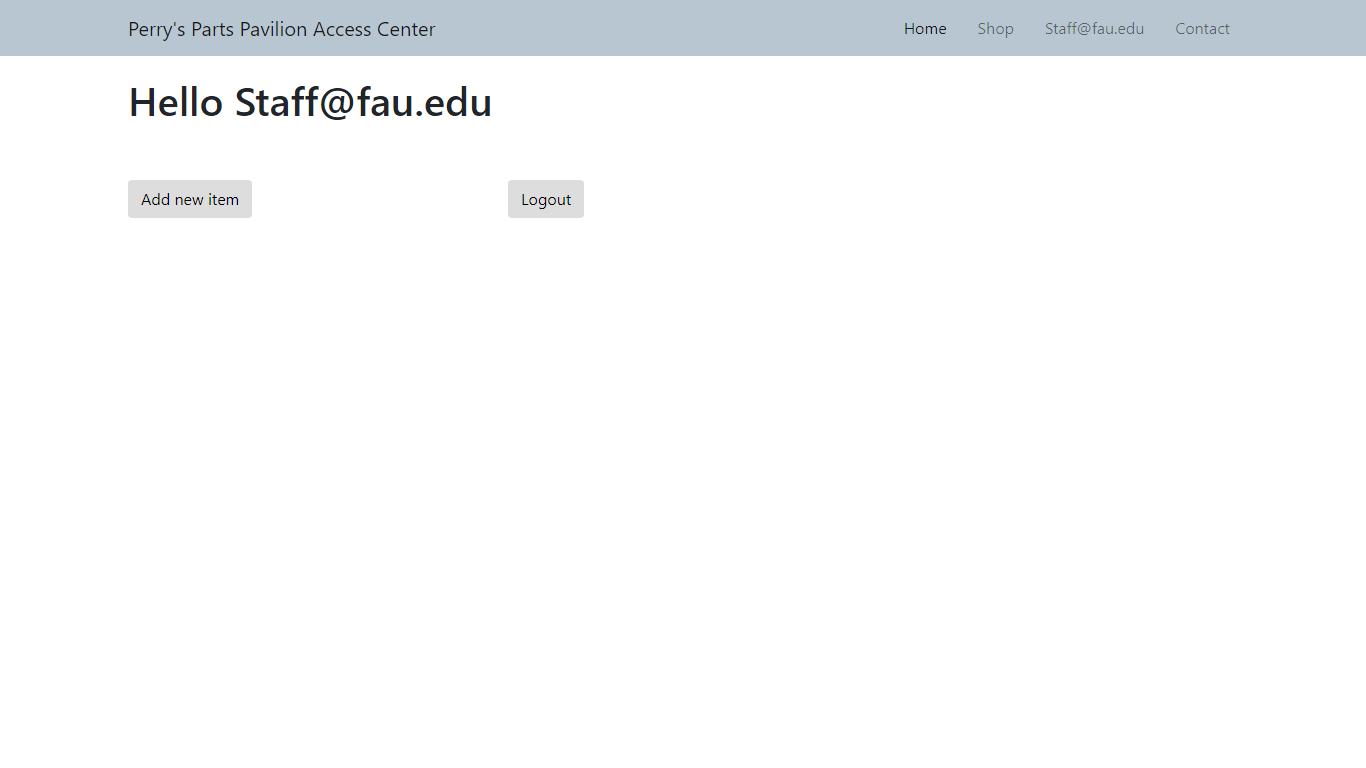
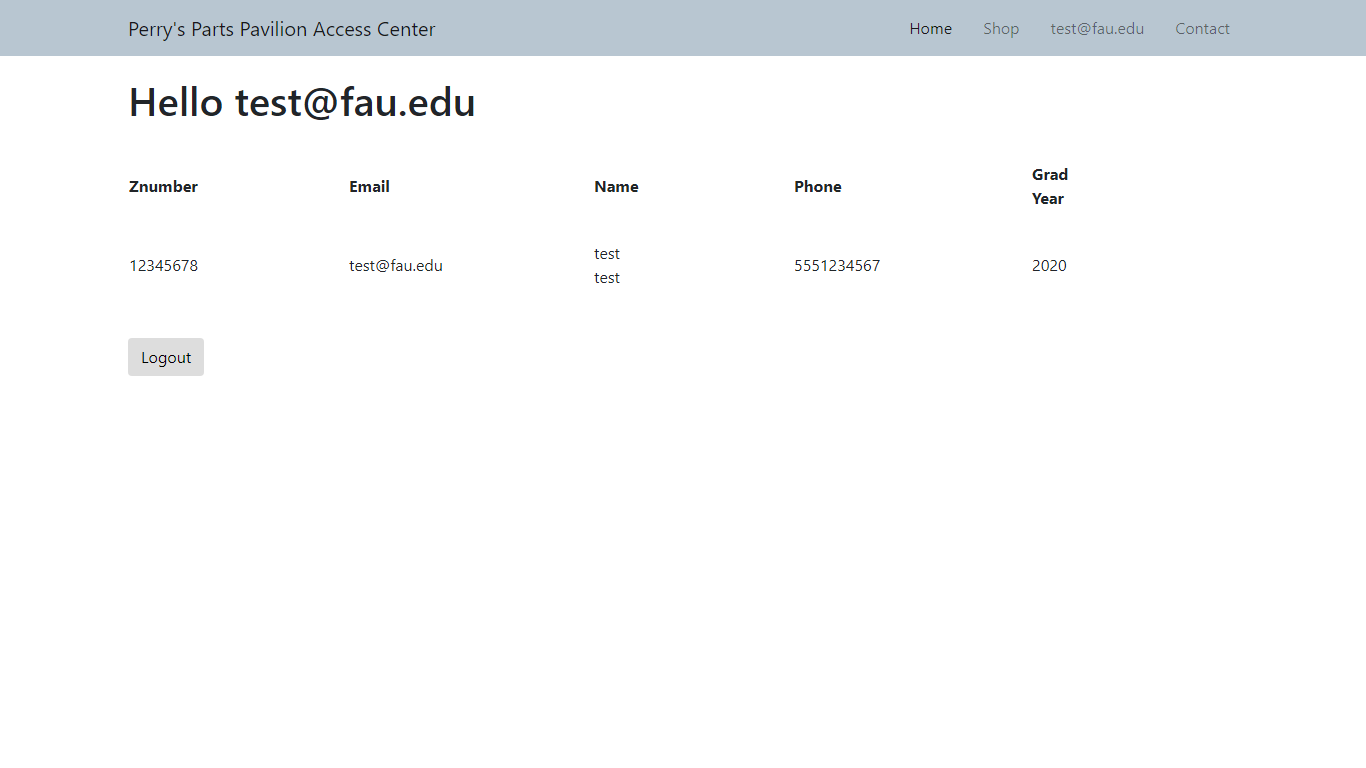
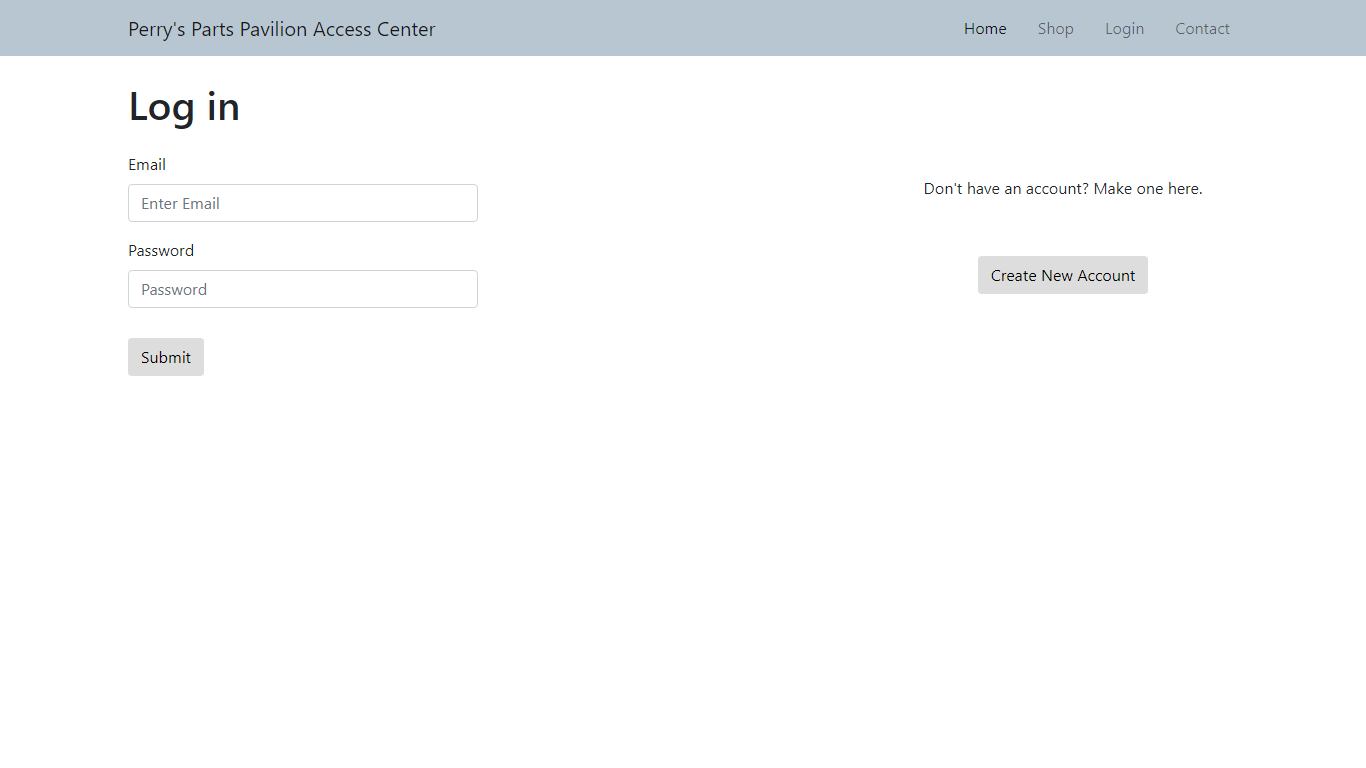
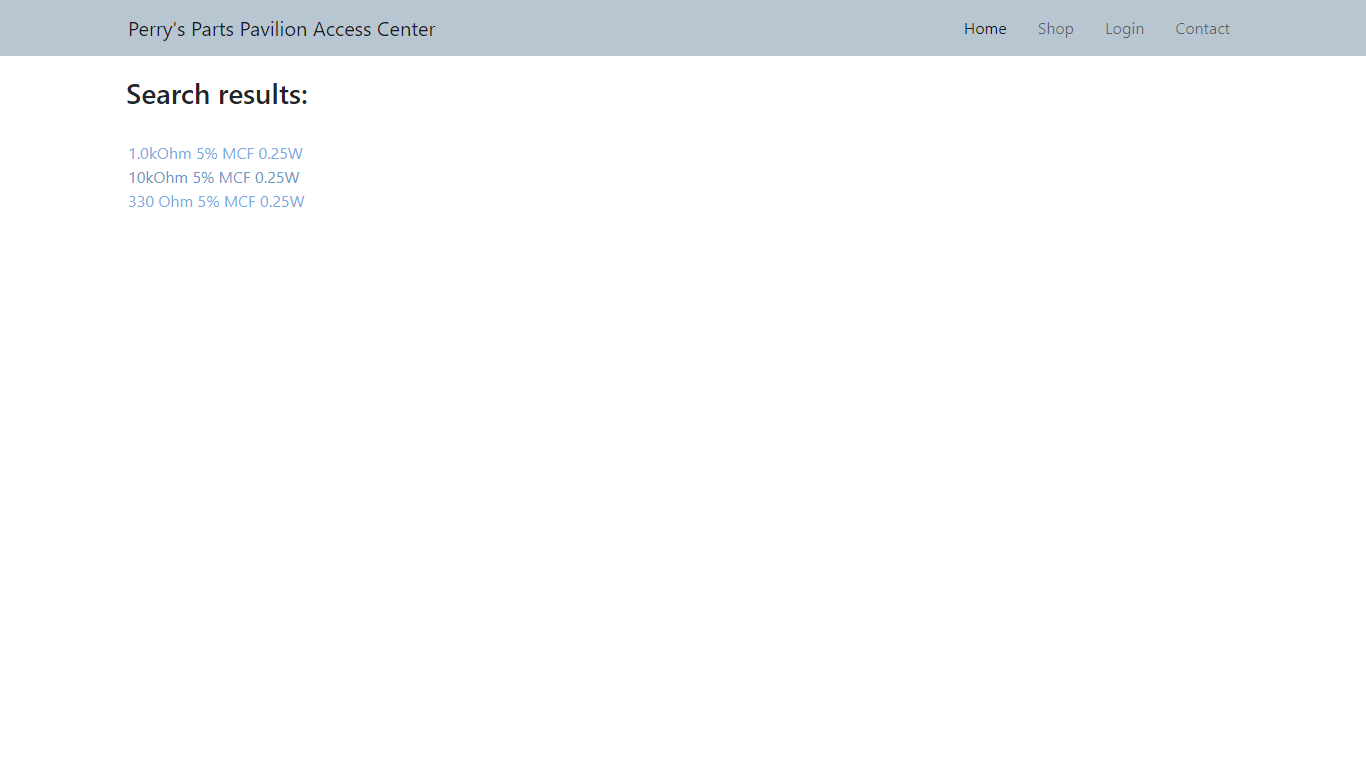
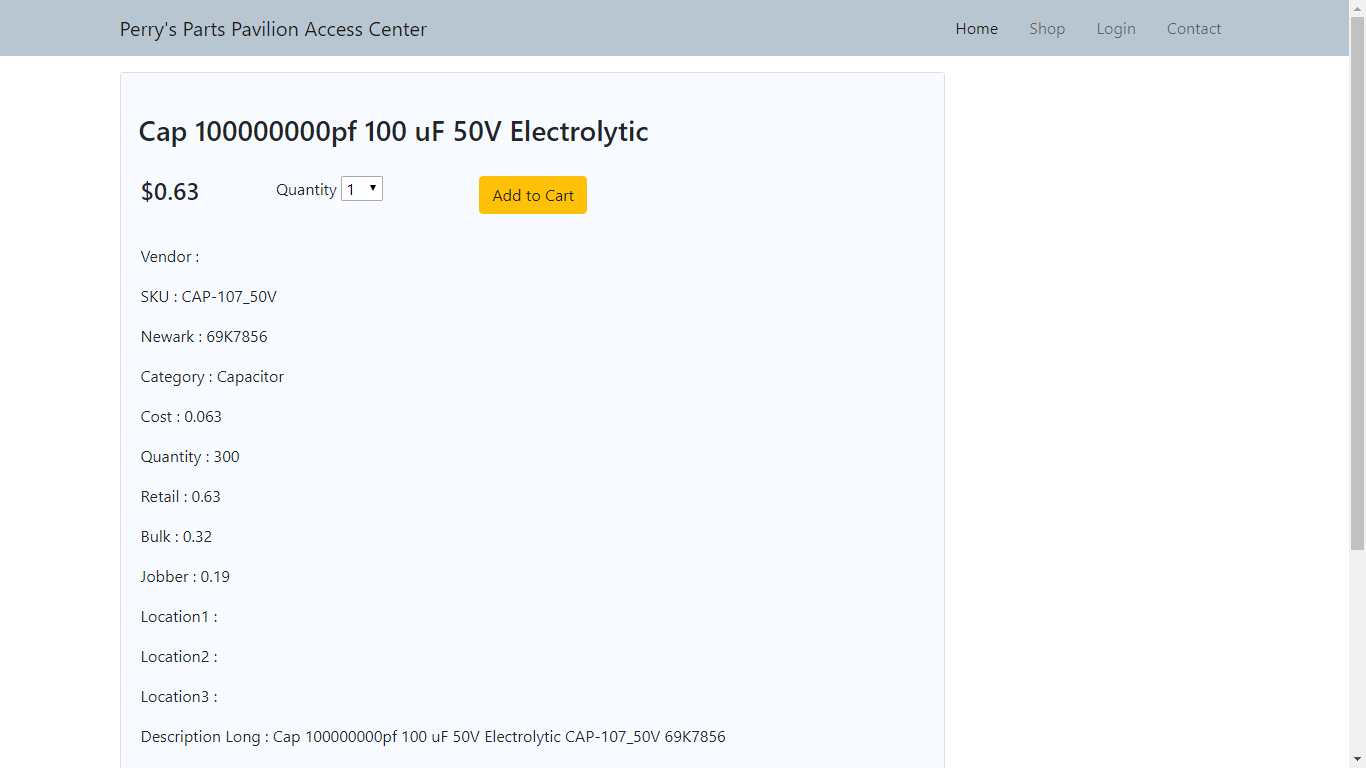
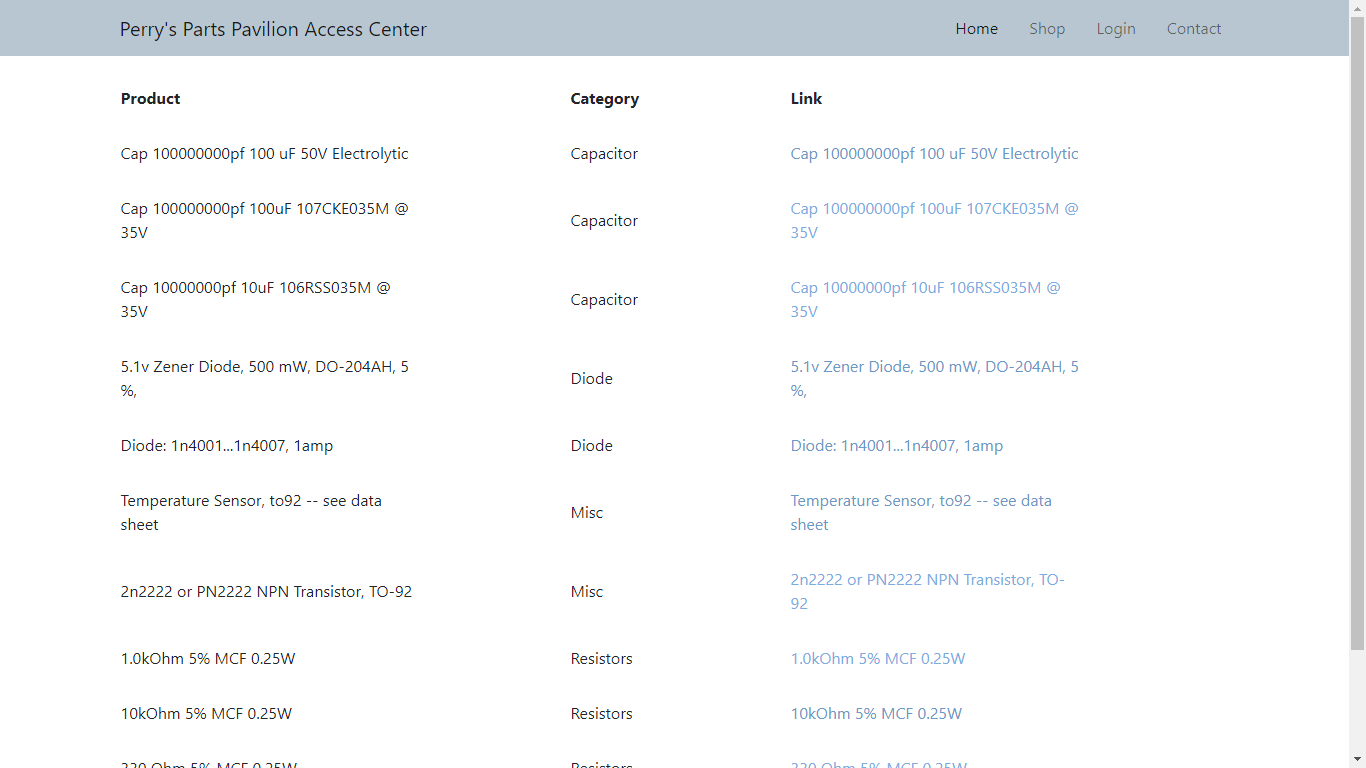
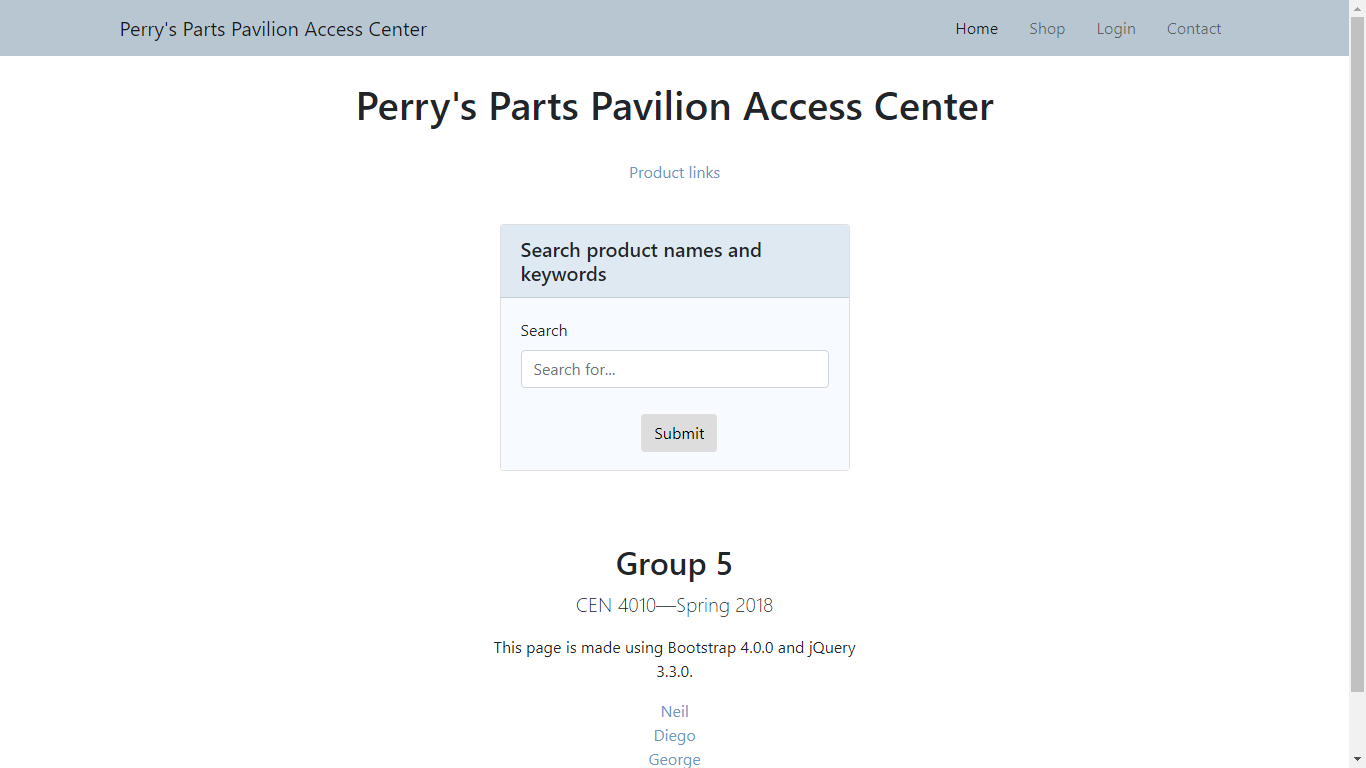
**Milestone 3:** “i can not access to the link for milestone 3 in github. I can not find the link for your vertical demo.” -Haicheng Tao, Apr 1 at 9:40pm

We were using branches in GitHub and the branch containing the documents had not been merged with master at the time.

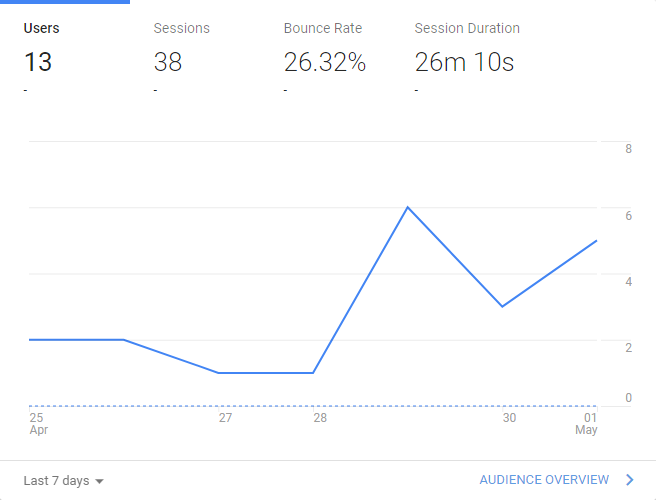
**Milestone 4:** “1. Explain clearly in peer evaluation file. 2. The 'search' function is available. 3. The file layout could be fancier.” -Haicheng Tao, Apr 23 at 12:52pm

The peer evaluation has been updated below under Team Contribution. Added the search function to the document. The file layout, assuming this refers to the product list on our website, has been cleaned up using CSS.

**Product Screenshots**



**Google analytics plot**



**Tracking ID**

UA-117910499-1

**Team members contribution**

|  |  |  |
| --- | --- | --- |
| **Name** | **Points** | **Contribution** |
| Neil Maniques | 36 | Assisted in all milestones.  Developed database structure.  Developed all the backend code including login, account creation, item creation, and item/keyword search.  Github 27 commits. |
| George Bechtel | 25 | Assisted in all milestones.  Created Github and Youtube account.  Researched using cookies for login verification and importing CSV files.  Did the CSS, modified and added HTML.  Github 20 commits. |
| Noah Leach | 25 | Assisted in all milestones.  Created UML diagrams.  Github 10 commits. |
| Franklin Carrillo | 13 | Assisted in milestone 2 and 3.  Last message on Discussion board was Mar 12, 2018.  Github 3 commits. |
| Diego Segura | 1 | Assisted in milestone 1.  Created the group Trello account.  Last message on Discussion board was Feb 19, 2018.  Github 3 commits. |

**Post-project analysis**

For this project, the main challenges faced are time and experience. In terms of experience, I was the only member that previously used javascript, php, and mysql. Only one other member had experience with html and css. In terms of time, we have other classes that have their own homework and projects. This limits the amount of time we can put into the project per week.

For the problem of experience, the only way to correct this is to study and practice. However this requires both a willingness to learn and time. During the first month, I posted programs and resources to help facilitate obtaining the skills necessary to build a website. It appears that these resources were not used by the people who needed it the most. Because backend programming is so intertwined with the requirements, I wrote the majority of the code that our group created. I did write some code that could be reasonably changed to fit another requirement, for example, add item information could be modified to add vendor information, but this was not done due to the constraints.

For the problem of time, I believe that other members underestimated the complexity of building a website. After milestone 0 and 1, I noticed a pattern that some people would only be available the weekend before or on the day that the milestone was due. While I hoped that their grade would be enough motivation to change this pattern, this was not the case. As the lead programmer, I knew that the requirements could not be programmed in a weekend, so I started long before the milestone was due. Even if they are only available to work on the weekends, I would have given them a head start on completing the task.

Perhaps better communication and delegation would counter some of these challenges. Ultimately, this does not address the issue that some members are unwilling or unable to their portion of work.

For this project, I developed:

1. Account creation function – adds users to database while performing checks for some inappropriate or duplicate accounts

2. Account validation function - prevents normal users obtaining access to staff functions, and remembers users until they log off or are inactive

3. Item creation - stores item information in the database and generates a link and a file to access the information

4. Item list/search - displays links to items after completing a two sided wildcard search for both product names and keywords

Planned features:

1. Keyword update – allows unlimited addition of keywords to an item

2. Item update – update item inventory and information

3. Dynamic items – item pages pull their information from the database and shows relevant information; customers will not see cost of items or location in store

4. Upload files – support the addition of files (pdf and images) which will be linked on the corresponding item page

**Final Project Demonstration**

<https://youtu.be/XPF6axP98JQ>