Iteration 0 Presentation

Team 3

Team 3

Team Leader, Implementation Leader : **Eli Curme** (ecurme@bu.edu)

Backup Team Leader, Configuration Leader : **Dinara Tiyekbayeva** (dtiyekba@bu.edu)

Requirement Leader : **Jay Hwang** (jw0508@bu.edu)

QA Leader : **Pelin Akbiyik** (pakbiyik@bu.edu)

Security Leader : **Chenghao Feng** (fengche2@bu.edu)

Project Overview

ZICZAC

Purpose

- To encourage **Buy/Sell/Trade** pre-owned items among people in local area for used recycling and reducing waste.
- To become a community platform for nearby neighbors.

Potential Users

- Users who want to sell or trade their items.
- Users who want to buy pre-owned items with an affordable price.

Related Work

Most Popular Buy and Selling Web Application in US

1. OfferUp
2. 5miles
3. Mercari
4. OFFERit
5. Carousell
6. eBay

Similar Web Application in Korea

- Daangn

The Key Feature of ZICZAC

- **Direct Transaction** (Buy / Sell / Trade)
- **Communicate with Chat System** between buyers and sellers.

Project Requirements

(**1 point)   Sign Up**

“As a new user, I want to create a new account, so that I can access my orders or post an item for sale.”

(2 points) Post Item for Sale

“*As a seller, I want to post an item for sale, so that it can be purchased by another user.”*

(2 points) Rating System

“As a user, I want to rate the other users after dealing, so that users with low rating get penalties on service usage. ”

(3 points) Chat System

“As a buyer, I want to communicate via message (live chat) to a seller for direct transaction, so that we can decide a price of item and time for dealings.”

(3 points) Search Engine

“As a buyer, I want to search items based on name and location, so that I can check items in local area.”

Management Plan - Process Model

- Influences:
  - Agile
  - Scrum
  - DevOps
- Priorities:
  - Deliver working software early, build further as project progresses
  - Cover essential requirements first
  - Team collaboration to overcome technical skill discrepancies
  - Use meeting times to clearly define responsibilities

Management Plan - Risk Management

Risk	Mitigation Strategy
Team member drops	Distribute work to team
Improper use of powerful git commands	Experimental work done in alternate branch
Requirements change / scope creep	Allow scope to evolve, provided essential features are in place

QA

Metrics:

- **Product metrics**: The software is designed to provide platform for users to sell and buy products. The user would create an account, which consists of username, secure password, location and contact information. The user would list the products that they are willing to sell. The user would upload a picture of the product and write the description of the product. Users who are interested in purchasing products would indicate the location and the type of product that they are interested in. The user would be able to see the reviews for each seller and contact the seller that they prefer. The seller and the buyer would then communicate through the messaging feature on the website and negotiate the price of the product. Once the user and the buyer agree, they would meet at a location to complete the transaction.
  So far, Account, Order and Item classes have been created and we have two Python files. One to store user data and one to store inventory.
  The method for the project is using Python pickle module in order to store data in a file. The credentials on the website will be checked against this file storage. Same method will be used for an inventory of items for sale.

- **Process metrics**: Unit test, integration test and system test will be applied to analyze and improve both frontend and backend.

- **Project metrics:** Total of five software developers are working on the project. Each software developer is responsible for one of the aspects of the project, however teamwork plays an important role and each developer participates in all aspects of the project. Assignments, deadlines and productivity play an important role. At least once a week, team meeting is conducted to discuss the project. The goal is to use time efficiently because the team has to accomplish a lot in a short period of time. User friendliness and security plays an important role in this project. The website has to be attractive to get as many user as possible, at the same time it has to be secure in order to protect user information and avoid scams. To achieve this, we put password criteria in place and disqualify sellers who receive points below a certain number. In addition, users can contact the customer service in order to complain about a seller.

QA

- **Product quality metrics:** Consists of defect density and customer satisfaction. Defect density is defect per unit of code. It is calculated as number of defects divided by the number of lines in the code. Our goal is to have zero or low defect density. Customer satisfaction indicates the product quality. Our goal is to have minimum customer complains and create a user friendly and secure website. To achieve this, we ask users to submit a survey about their experience with the website and make improvements based on their answer.

Standard

- We use HTML/CSS/JS for frontend and Python Flask for backend.

Inspection/Review Process

- Review will be done mainly by the QA leader, however the entire team will also review. The quality and user friendliness of the software will be reviewed after completing the framework of the software. Improvements will be done based on the result. Inspection on the software will be done frequently to make sure there are no defects or logic errors in the code

Testing:

- QA will conduct the testing mainly, however all the team members will participate. Testing will be conducted each time the code is updated. Each time the test is conducted, the results will be documented to a separate file with the date of the testing. Tests will consist of debugging and fixing the errors.

Defect Management

- If the code does not execute properly, that means it is defected. Debugging will be the tool to manage defects.

Configuration Management Plan

1. Source and configuration codes location: Github
2. Github structure:
    a. Main branch
    b. Each user’s individual branch

3. Github updates notifications: automated messages in slack

4. Github file/folder naming convention:

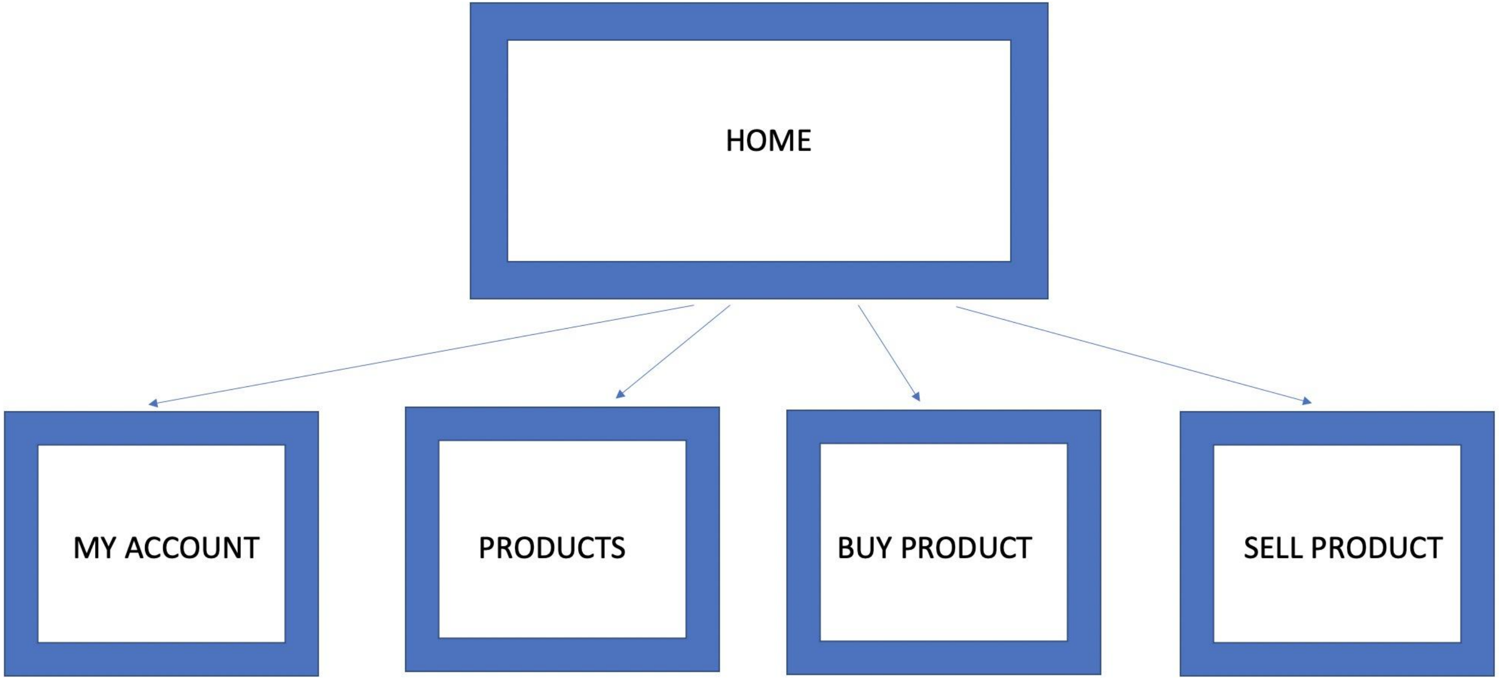
    a. File_name_v_version.file_extension
    b. Folder_name_letter

        where letter represents how often file needs to be backed up
        A - once a year
        B - once a month
        C - once a week
        D - daily

5. Github commit rules:

    a. Clear git commit message which outlines what changes have been made
    b. Commit message has link to testing document that describes what testing was performed on changed code

Website workflow



HOME

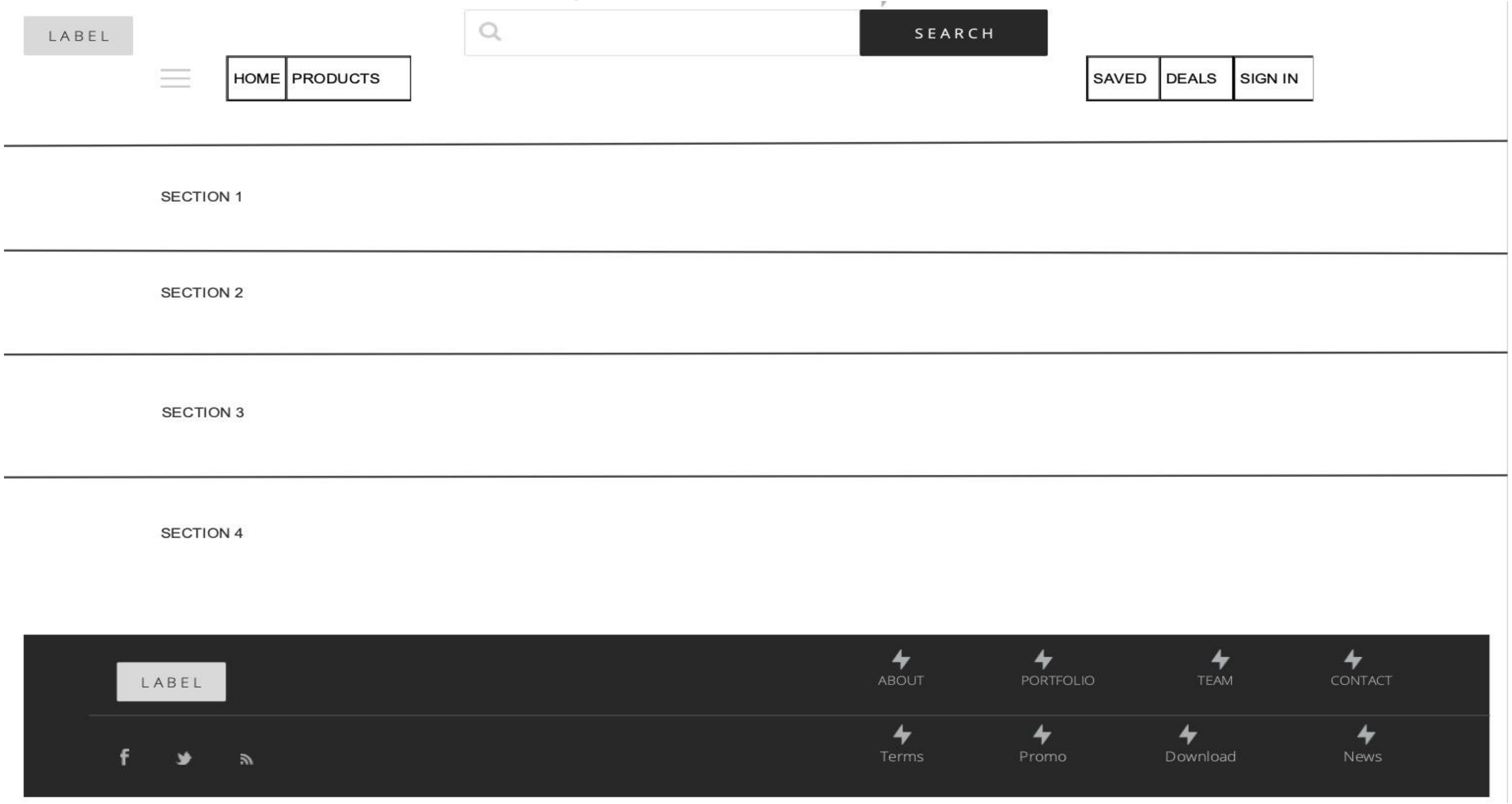
MY ACCOUNT

PRODUCTS

BUY PRODUCT

SELL PRODUCT

Website initial design



Front end technical stack

Building from scratch with:

HTML
CSS
JS
5
3
JS

Incorporating CSS framework Bootstrap and JS framework such as JQuery or React, other frameworks will be decided as the project progresses.

Password Security

- Secret = Password
- Verification
- Keep safety. How?
  - Encryption
  - Strength of Password

Change them often, keep them private and never share them with anyone.