University of Windsor Food Bank

Group – 11

Course – Advanced Software Engineering

Jay Pandya
University of Windsor
pandya31@uwindsor.ca

Toral Patel
University of Windsor
patel6i@uwindsor.ca

Richa Kansara
University of Windsor
kansara1@uwindsor.ca

Henil Parikh
University of Windsor
parik11n@uwindsor.ca

ABSTRACT

Managing system brings more users to system and removes manual interaction and managing of records for years. It provides easy retrieve and modifications of entry to Food Bank system and help students with available product details and notifies students about out of stock products when they are in stock.

I. INTRODUCTION

Easy process for students to get free food products from University of Windsor is best way. To ease the process this platform will help them. Managing manual records of inventory is tedious and autonomous task so it can be removed using this. The full-time students find it difficult to go to food bank and select items and waiting in long queues occupies their time, so students avoid taking food from food bank and prefers local grocery stores over it. This platform well manages suppliers and volunteers along with easing students' access. Many people want to donate food, but they are sometimes unaware where can they donate and above this which product is very much in demand or which item is out of stock. So, this system. This website will help full-time students to select the food items from Food Bank as per their needs and can add to cart and acquire free products every two weeks. System works same as manual food bank working at Canterbury College where manual handling of product items is done, also manual entry of students is getting maintained so that each student can get free food every two weeks and one item from each category.

II. OBJECTIVE

Objective is to provide website which handles inventory of items. Students can select product one each category and on checkout they can select pickup time. Feature to add favorable food item into wish-list is available for students. Moreover, they can edit their Wishlist by changing the food item or deleting it. So complete flow for student will be, selecting products, view product details, and adding items to cart and at checkout they can schedule date and time for picking up products from physical food bank. Also, students can directly walk-in to physical food bank and can get product, but volunteer need to make an entry for that students and number of items they have taken into a system so that inventory gets maintained properly.

Our important concern for our website is to make flexible and user friendly. Furthermore, the main aspect is to make our website authentic and secure so that students can trust blindly.

The above all statements are related technicality of our website but the food product which is provided in the food bank is donated by the donors and suppliers. Also, system will contain notification feature which will notify students if they have products added to Wishlist and those products are out of stock and notify them when they are in stock. Also, separate UI will be managed for suppliers, volunteers and admin where suppliers can see products out of stock.

III. DELIVERABLES

Food Bank web application will contain login pages for all users. Separate landing page after login for users Admin, Suppliers, Volunteers and Students will be managed. Full-Time students need to register before login and for remaining users, admin will add user to site and system will send mail with login credentials which is added by admin to system. They will also have common features like Change Password and Forgot Password.

Volunteers will have access to feature that can scan barcode of products acquired by students while walk-in in Food Bank. So that using that barcode of product, quantity gets updated in database. For this volunteer need to login to system and open Manage Product page where he can select product and user if exists and if not then user can be added by volunteer and acquired product items of users. Also managing time availability can be managed by system where volunteers add time they are available so accordingly shift will be given by admin.

Admin will have the super role among users i.e. can accepts user's registration by matching studentid from database and, also matching the course and whether it is part-time or full-time so accordingly can avail benefits. Admin will also update via website to suppliers about which item is running out of stock so that it can be delivered by them. Volunteers will be added by admin and their shift plans will be managed according to availability provided by them.

Database Operator is also one of the role that is present in website. User with this role can add, edit, update and remove products, categories, ingredients, and manages inventory. So when physical products get delivered to food bank, database operates add it to database so that it will get reflect on site to all users.

Users will be able to filter, sort according to price, alphabets, food category and many other choices. We would be displaying them the ingredients of every food item on the details page of the food item. The other functionality which we would be providing to users is facility of Walk-In in which if user don't have access to application and wishes to directly go to the food bank during open hours so he/she would be able to talk to the volunteer and will be able to take the food items they are wishing to take it.

IV. Problems Faced With Existing System

Before this system developed, the students faced difficulty in getting items which they want as they didn't have any idea about which items were being present in the physical food bank. So, when they went to the Food bank, they ended up getting items which were present in the food bank rather than the items which they wanted. The other problem was that most students were unaware about the locations of Food banks in the city.

Due to this, they were not able to take advantage of the University food bank service. Due to lack of communication between Students and donors, donors were not able to identify the needs of students. Due to which, students were not able to get items of their needs If student was away from the city in the days when food bank items were being provided, they were not able to get those items. IN traditional system, the data about students who took food and available food was kept in Ms Excel, so it was difficult to find details about particular students and food when it is required.

V. Project Architecture

University of Windsor Food Bank (UWFB) project architecture is composed of different layers which interact with each other to help university students in accessing food items from the food bank in evident manner.

The User Interface of UWFB is imposed on ReactJS which is one of the high-reaching technology these days. Moreover, the other technologies which may be used are HTML5, CSS3, BOOTSTRAP4 and JavaScript.

Unit test cases are implemented using React Testing Library and Jest. The backend of UWFB is implemented using Python language. Django framework of the python will be used to fulfill the functionality of different modules. The database operations are done using MySQL.

VI. Drawback And Mitigation Plan

When barcode fails to get match from the database entry using available library. As a solution to this problem, selecting product from available list will also be provided to volunteer, so that volunteer can select product manually from site and save quantity count if barcode fails to do so.

If student transfer the course from full-time to part-time, in future, university database request will be made so that it can be updated accordingly on this site so whenever student changes enrollment, then student will be denied availing free products.

If user is ended with the graduation, then also he/she can acquire the facility of food bank. This problem will be solved by same way that is access to university database in future so that this update is accessible so that after completion of course, student should not have right to avail to this benefit so this will refrain them from doing same.

VII. Test Strategy

We would be following Test Driven Development approach as a testing strategy [3]. According to this strategy, we are writing test cases first and then start development. Few scenarios will be handled in test cases and if it gets pass then that code will be used in development and then unit test cases will be executed with complex scenarios to verify the code works properly and as expected.

Also automation scripts will be written to test complete website and then that it will be deployed on Jenkins job so when project development is done, Jenkins job will take update from git server and creates build of project and further Jenkins job will execute test cases before creating build and automation script on deployable build and accordingly pass and fail test case, deployment will be decided. Testing will be coded by JUnit library, and method wise test cases will be written with different values as input. So above mentioned approach is known as Software Testing Life Cycle which will be followed.

- Regression Testing:- After system is build, regression testing of all features will be performed.
- Cross-browser Testing:- System will be tested in all browser and also mobile browsers.
- Usability Testing:- Tools: This system test will be used in completely test website user interface design. To check all the pages of site are user friends and easy to understand and no additional knowledge is required and proper messages are displayed and no error message is displayed which is difficult for user to understand.This will be tested by all team members. Test cases will be written for all pages with expected value for particular behaviour and after testing actual values will be mentioned and accordingly application will be modified to meet expected value of written test cases.

- Performance Testing:- We will use tools such as PageSpeed Insights, Apache, Jmeter or webpagetest.org.
  In our system there will be a situation when multiple users would try to access the website at that time the data should get load in proper response time, speed, stability, efficiency, data transfer rate, proper throughput should be maintained. We will test our website by placing link and check responsiveness, stability and efficiency on webpagetest.org.
- Stress Testing:- We will be using tools such as Arachni or Nogotofail in this type of testing.
  Security testing will be used to identify the threats in the system and measure its potential vulnerabilities, so the system does not stop functioning or is exploited. It also helps in detecting all possible security risks in the system and help developers in fixing these problems +through coding.We will conduct the text by querying database all the records are encrypted so that no user data is revealed to the user

VIII. Product Documentation Draft

Requirement Documentation:
Functional:
This website includes various features like: Searching, sorting, filtering, adding product to cart, adding product to Wishlist in addition to this user can provide ratings and comments after purchase of products. The highest level of data security and data prevention is provided to the user.

Users - The user would be able to view the food items, select the food items, they can add the food product to the cart, if any item isn't in the stock then they can add that product to the Wishlist so in future if stock comes they will get a notification of the product. After that user have to go to food bank and take the item.

Volunteers - Volunteer would be having privilege to scan the bar code of the products which user wants to take. Volunteer can edit, update, delete and add the details of the products.

Donors - Donors would be able to add, edit, delete food product in the website.
Admin - This person will be having all the rights to system.

All users, donors, volunteers, admin would be able to sign up, sign in, and log out to the system.

Non-Functional Requirement:

Reliable backup of the client database on the cloud. The client's information needs to be encrypted to maintain confidentiality. The system should not crash in between the office operation hours so that the company does not lose valuable time. Users must change the initially assigned login password immediately after the first successful login. Moreover, the initial should never be reused.

It should be capable enough to handle 20 million users with affecting its performance.

User Documentation:

User will be able to give us feedback and as per their feedback we will try to fix the errors. Secondly, there is quality analyst in our team who would be testing the quality of the system.
Different operations such as displaying items available in Food bank, add items to wish list, add to cart, place an order and generating unique Order Id and Requesting unavailable items will be performed in the website. Apart from this, it will generate page not found error if device is not connected to internet but it will automatically be directed to requested page once it gets connected.
Customer will able to perform following activities on website: -
1)register if user is new
2)Login
3)Adding product to cart
4)Adding product to wish list

IX. Workflow

Till now, we have mentioned the developing tasks which are completed:

- Created Project Structure of Back-End using Python Language Django Framework and Frond-End project using NPM package explorer, ReactJs and Bootstrap. For accomplishing this step, we have added required tools in system and imported projects in respective IDEs.
- Looked for library that reads barcode which is in python and have created prototype using that library. Single class is created in which path to image is provided in class and result is displayed in new window with image and its name.
- Created react-bootstrap app and created admin dashboard with header, footer, navigation and graphs which shows sample data of users logged in, products utilized, highest selling product
- In JIRA, we have assigned individual tasks to every single member.
- Created GitLab repository and have added all code into it.

X. References

References
[1] University of Waterloo Campus Food Bank https://campusfoodbank.com/
[2] University of Alberta Campus Food Bank http://foodbanksalberta.ca/locations/campus-food-bank-society-university-of-alberta/
[3] Testing Strategy - https://www.guru99.com/how-to-create-test-strategy-document.html