CS 361

TMW

Tomorrow (formerly known as Right Now)

Prepared By:

Mariam Ben-Neticha, Raymond Lo, Donald Morgan, Christiano Vannelli, John Watts

XP: Cycle 1

User Stories

Selection of specific country to aid

User can select a country in need of aid from an exhaustive menu of items

View items country is in need of

The user can browse the selection of items available to be bought/shipped to the selected country

View in-depth explanation of product

User can select an item and view an in-depth description on both the product, as well as it's intended purpose

Purchasing of aid products

User can pay for item on a payment page which prompts user for payment information and confirms that the user wishes to complete the order before viewing a receipt of the order.

Contribution towards "Action Items"

User can view "Action Items" on the home-page of the website that indicate the goal amount as well as the donated amount for each item.

Selection of intended contribution amount

User can select the "Action Item(s)" the user wants to contribute towards and enters the desired contribution amount as well as payment information.

Update on amount contributed towards "Action Item"

"Action Item" donated amount is updated after every donation to be displayed on user screens.

Update on available stock

Item availability (in stock, etc.) is updated after every purchase and makes known to the user if an item is currently out of stock.

Viewing of shipping information

User can use confirmation number(s) to track items purchased or donated towards.

XP: Cycle 1

Task and Effort Estimate (per user story):

Selection of specific country to aid

Task: HTML (Handlebars) - Website Layout /Node.js - routes for web pages, etc
Effort Estimate: 6hr

View items country is in need of

Task: HTML (Handlebars) - Website Layout /Node.js - routes for web pages, etc
Effort Estimate: 4hr

View in-depth explanation of product

Task: HTML (Handlebars) - Website Layout /Node.js - routes for web pages, etc.
Effort Estimate: 1hr

Purchasing of aid products

Task: HTML (Handlebars) - Website Layout /JavaScript -Cart Table creation & other functions /Node.js - get/post requests/MySQL - selection and creation queries
Effort Estimate: 6hr

Contribution towards "Action Items"

Task: HTML (Handlebars) - Website Layout /JavaScript - event-listener functions /Node.js - get/post requests/MySQL - selection and creation queries
Effort Estimate: 2 hrs

Selection of intended contribution amount

Task: HTML (Handlebars) - Website Layout /JavaScript - functions /Node.js - get/post requests/MySQL - selection and creation queries

Effort Estimate: 2 hrs

Update on amount contributed towards "Action Item"

Task: HTML (Handlebars) - Website Layout /JavaScript -functions /Node.js - get/post requests/MySQL - selection and update queries
Effort Estimate: 3hr

Update on available stock

Task: HTML (Handlebars) - Website Layout /JavaScript -Cart Table creation /Node.js - get/post requests/MySQL - selection and creation queries
Effort Estimate: 3hr

Viewing of shipping information

Task: HTML (Handlebars) - Website Layout /JavaScript -Cart Table creation /Node.js - get/post requests/MySQL - selection and creation queries
Effort Estimate: 3hr

XP: Cycle 1

Priority List

1. User can browse the selection of items available to be bought/shipped to that country
2. User can select an item and view a description of that item as well as the cost.
3. User can pay for item on a payment page which prompts user for payment information and confirms that the user wishes to complete the order before viewing a receipt of the order.
4. Item availability (in stock, etc.) is updated after every purchase and makes known to the user if an item is currently out of stock.
5. User can select the "Action Item(s)" the user wants to contribute towards and enters the desired contribution amount as well as payment information.
6. "Action Item" donated amount is updated after every donation to be displayed on user screens.
7. User can select a country in need of aid from a menu
8. User can view "Action Items" on the home-page of the website that indicate the goal amount as well as the donated amount for each item.
9. User can use confirmation number(s) to track items purchased or donated towards.

XP: Cycle 1

Group Work Summary

Front-End

Mariam and Christiano joined together to form the front-end pair programming duo. This pair met every Monday and Wednesday for 2+ hours per session. Together they screen shared while working in text editors such as Sublime Text 3 and used Bootstrap templating to put together a skeleton for the website. Throughout the first cycle, they continued to add more content and functionality to the application; building up the views directory with .handlebars web pages.

Once the web pages were laid out, they added imagery and button functionality, to make the website an interactive experience for the user. They then performed unit testing for Cycle 1, utilizing the Selenium IDE extension for Firefox. They were able to determine points of weakness and ensure that the web pages were functioning as intended.

Back-End

John and Raymond joined paired up to program the back-end. This group met during various days of the week for 2+ hour sessions. They used screen share, ssh, putty and coding in Notepad++. For the first cycle, a MySQL database was created on the ONID server, and accessed using PHPMyAdmin. The first cycle consisted mostly of constructing and testing the database schema, along with basic manual item creation queries.

For the second cycle, the database was migrated to an Amazon EC2 instance running Ubuntu Server and MYSQL to allow for global access. Some minor flaws were uncovered in the schema, which was simplified to improve the frontend user experience. A SQL procedure was created to update the active cart, and additional queries were added to improve cart functionality.

XP: Cycle 1

Unit Test Summary

Front-End

Unit testing was completed utilizing the Selenium IDE extension for Firefox. Our unit tests determined the user's ability to successfully view an item, traverse through the website, use the carousels, and interact with the imagery. Below are some of the unit tests that we ran via Selenium.

Cycle 1 Tests	
click	link=India
click	link=Congo
click	link=About
click	link=Tracking
click	link=Community Events
click	link=TMW
click	css=ol.carousel-indicators › li
click	//div[@id='carousel-example-generic']/ol/li[2]
click	//div[@id='carousel-example-generic']/ol/li[3]
click	css=span.glyphicon.glyphicon-chevron-left
click	css=span.glyphicon.glyphicon-chevron-right

The “click” on the “css=span.glyphicon.glyphicon-chevron-right” element produced an error which was resolved after inspection of our html.

Back-End

Unit testing was done manually by running each query through PHPMyAdmin, and later using the MySQL CLI on the Amazon EC2 instance, utilizing feedback from each query’s execution. We used Notepad++ to create queries for tables. We then connected to PHPMyAdmin or the MySQL CLI to run these queries. We had some syntax errors when running the queries, but MySQL informed us where these errors were and we quickly corrected them. We then iterated this process until each query executed successfully.

XP: Cycle 1

Acceptance Testing Summary

Our client for this project happened to also be a group member. As a result, we had his input during our acceptance testing. We utilized the black box acceptance testing method; the input values used were the mouse clicks as well as inputting the proper URLs for the different web pages.

Beginning on the home-page, we first started by clicking on the carrousel pictures, which redirected the client to the 'India' and 'Congo' pages respectively. Next, we tested the buttons on the left of the home page for 'India' and 'Congo', which loaded each of those pages as expected. We then tested links on the navigational bar, 'About', 'Tracking' and 'Community Events.' These pages loaded as expected, the URLs for the 'India' and 'Congo' pages as well as the 'About', 'Tracking' and 'Community Events' pages were input and loaded properly.

At the end of this acceptance testing cycle, the team agreed that the 'Community Events' page, which was meant to have a listing of all the major 'Action-Items' for each Nation in Need, would be better distributed amongst the Countries for which the 'Action-Item' would be directed. This change would improve visibility for the 'Action-Items', which have a great importance to the project as well as improve the layout of the website as a whole.

XP: Cycle 2

Changes Summary

Front-End

During the second cycle, many changes were made to the website. A 'cart' and 'checkout' page were created, where the 'cart' page could be accessed from the navigational bar and the 'checkout' from within the 'cart' page. Within the 'India' and 'Congo' pages, item pages for 'water' and 'first-aid' were created for both countries. The 'Action-Item' pages for each country were also added to the website. This provided some changes to the layout as a result of the acceptance testing. Instead of having a 'Community Events' page for the 'Action-Items', we simplified the website layout and included the 'Action-Items' of each website within the Nation's page.

During this second cycle, we also added an update functionality to the cart and applied functionality to the items listed for addition to the cart. At the end of the cycle, we decided we did not have enough time to complete the Tracking user-story and removed the 'Tracking' page from the website. The unit testing was completed in the same manner as XP-Cycle 1 using Selenium IDE. Acceptance testing for the changes made during this cycle were also completed in the same manner as XP-Cycle 1. These tests helped to identify any programming errors within our program.

Back-End

During the second cycle, we had to change databases because the local website had a hard time connecting to the ENGR database. We created a AWS EC2 instance with MySQL installed and an FTP server. We filled the tables with objects that were required from the site, such as each page's item, purchaser's form, cart and purchase confirmation.

XP: Cycle 2

Group Work Summary

Front-End

The team remained in the same pair-groups formed during XP-Cycle 1 since the pair-members were familiar with the work they had done on either end. Christiano and Mariam continued to meet every Monday and Wednesday for 2+ hrs per session to work on the project together using Google Hangouts Screenshare. They also maintained contact through Google Hangouts Chat to update the other pair-member on any additional work they completed on their own.

XP-Cycle 2 was more difficult to work through since the majority of the tasks we worked on were dependent on the database and merging the Front-End work with the Back-End work. In addition, the functionalities for the features the client wanted to implement on the website were fairly complex. With the number of pages the website contained, this became an increasingly unrealistic task for the time-frame given. As a result, the team revisited the user-stories and renegotiated the tasks that the team would be able to complete by the end of the deadline.

Back-End

During the second cycle, we had a very hard time trying to connect to the ENGR database. We were running these queries coupled with the code from the website and it would give us a handshake error. We tried different ways to connect to the server, but were able to come up with a solution by visiting the Piazza discussion board. Here we found out that the database has trouble with locally built web-pages. We then decided to create an AWS EC2 instance with MySQL installed and an FTP server. We had success connecting to the database, but we ran into some trouble uploading files to the server. After some troubleshooting we were able to connect to the FTP to upload files. We then ran into trouble with being able to access the files we uploaded with the website and found out that UNIX sets its own permissions to files and was able to change these to get everything up and running.

Reflection

XP Preferable to Waterfall

The XP method allowed the project to move forward at a faster pace. Although it was not as clearly laid-out as the Waterfall method, the XP method allowed the programmers and the client some liberties in terms of changing the features or tasks of the project to bring about a better result, especially given the time constraints inherent to this project.

Waterfall Preferable to XP

The waterfall method seemed much more organized and had very clear cut goals, whereas although the goals in XP are defined, sometimes the extra freedom causes disorganization. Communication of low-level decisions was somewhat difficult without structured documentation to share between team members.

Strengths

The XP method allows for a good amount of freedom and the ability to move into developing at an accelerated pace. This is a great method for getting a working product in a short period of time and relatively low-cost.

The Waterfall method is very structured and the architectural design is much more thought out. This is a great method for getting a well-functioning and polished product that will have minimal bugs.

Weaknesses

The XP method can become disorganized and sometimes the pace becomes too quick for one group or the other when splitting into pairs, when one pair is relying on the work of another pair.

The Waterfall method is a structural behemoth and sometimes creating diagram after diagram doesn't seem to be as effective as diving straight into the development. It can also be very cost-inefficient.