Homework 7

Assistant for Recycling and Waste Management to Reduce Recyclables in Landfills

Noah Beach, Kevin Christensen, Emmet Cooke, Jacob Leno, Caleb Scott
CS361 Group 18 - Spring 2018

1. Contributions Summary

The work was divided among our group for HW7 as follows: Noah Beach - User Story #15/UML Diagram Usefulness/Future Schedule Planning, Kevin Christensen - User Story #7 and #15, Emmet Cooke - User Story #7/Integration Testing, Jacob Leno - User Story #4 and #6/ Integration Testing, Caleb Scott - User Story #4 and #6.

2. Software Description

2.1 Week Two User Story Work

Week One Software URL

The software for week two work can be used at the following URL: http://cs361group18.noahbeach.com

User Story 4 Work Summary

Jake Leno and Caleb Scott worked in a pair to complete this user story. The user story took about 3 hours to complete but was estimated to take around 4. A lot of the code was similar to code used previously on the machine control page so the basic structure of this code was reused and modified to fit this situation. A router was created for each button on the reports page. Each time a button was pressed it would send information to the server via a GET request. The appropriate router would handle the GET request and call the getReports function with arguments corresponding to the information requested. The getReports function would then query the database for the information and return it the the calling function.

The SQL query was validated by testing it directly in the database with values that would simulate a request from the server. The javascript executing these queries was tested by sending GET requests and outputting the results to the results to the console for verification.

Problems were encountered when trying to add new recyclables to the database. Two of the tables had their foreign keys created incorrectly. For each table the ‘id’ attribute of the table was referencing the ‘id’ attribute of another table. The structure of each table had to be changed so that the attribute intended to be a foreign key reference was pointing to the ‘id’ attribute of another table.

Currently user story 4 is implemented and a supervisor would be able to view this page to see information on what had been recycled in the past day, week, month and year.

User Story 6 Work Summary

Jake Leno and Caleb Scott worked in a pair to complete this user story. The user story took about 3 hours to complete but was estimated to take around 2. A lot of the code was similar to code used previously on the reports page so the basic structure of this code was reused and modified to fit this situation.

The SQL query, the router functions, and the getReports function used on this page were the same as what was used for user story 4. User story 6 and user story 4 have very similar uses and therefore this code was able to work for both user stories.

Initially on the public reports page we wanted the page to display without the navigation bar but we ran into some issues getting the page to use a different layout. The website is set up to use a default handlebars layout in main app.js function. We were able to get the router to use a different layout than the default but were not aware of it because the {{{body}}} tag associated with the handlebars engine was left out of this other layout. This caused us to troubleshoot the program in several ways that were not related to the problem and more time. We also had some confusion because the links in the buttons were set to call the routers for the reports page.

Currently user story 6 is implemented and a member of the public would be able to navigate to this page to see the reports on what has been recycled by the recycling assistant.

User Story 7 Work Summary

Kevin Christensen and Emmet Cooke worked in a pair to complete this user story. The user story implementation took around two and a half hours to implement, but it was estimated to take around ten. With the use of pair programming, we were able to massively cut down on the amount of time required for this user story. A new table was added to the public reports page, which was done in a short amount of time, and then the table was filled with data, which required a bit more effort to get the correct results.

To validate the SQL queries, each query was compared against the correct results to verify that they were returning the correct data. The javascript executing these queries was tested by sending GET requests with valid and invalid query results that the code would respond correctly to.

Problems encountered included issues generating the correct SQL query so that the correct number of tagged recyclables were counted. Once the issue with generating the correct query was sorted, there were no issues with appending the new graph to the webpage.

The current status of the user story is that it is implemented. A user is able to go to the public reports page and view the correct data in the implemented table.

User Story 15 Work Summary

Noah Beach and Kevin Christensen worked in a pair to complete this user story. The user story implementation took approximately 9 hours to complete, and was estimated to take 7 hours total. Writing/testing the SQL queries took approximately 4 hours. The increase in time was mainly due to the requirement to add an additional table to the database to handle employee tracking. In addition to the requirement to design a new table, the queries for this user story were more complex than other user stories, requiring sql queries that join across three tables to achieve the desired information for the user interface. Designing the user interface took approximately 5 hours, this time included debugging.

To validate the SQL queries, each query was tested in phpMyAdmin with the correct inputs to verify that the queries were correct and would return the correct data. The javascript executing these queries was tested by sending GET requests with valid and invalid query parameters to verify that the code would respond with the correct data.

Problems encountered included issues with getting the query that joined the employee, machine, and facility tables together took quite some time to get correct. Another problematic area was developing the formula that determined if a machine was adequately staffed or not. The final formula decided upon was:

$$\text{Inadequate Staff} = \frac{(100 - Tagged\ Waste\ Percentage)}{\#\ of\ Staff\ Assigned\ to\ Machine} > 20$$

Though we feel this is a good starting point, this formula will have to be tweaked based on real world statistics once a machine is put into production.

The current status of the user story is implemented from a user interface perspective. The percentage of tagged waste is currently randomly generated by a function since the underlying hardware/hardware interface for the system is not implemented but the control interface is ready to accept it.

2.2 UML Diagram Discussion

The UML Sequence diagrams developed during the previous week (HW6) were useful for the actual implementation of the user stories. Since the purpose of the UML sequence diagram is to show the interactions of objects in sequential order it allowed the pairs of developers to imagine the steps required in the user interface to complete the user story.

The diagrams also gave the developers a birds eye view of what the user story would look completed.

Finally the diagrams provided the development team with something to present to the customer (had that been a necessity), allowing the customer to provide feedback on the different aspects of the interactions the user would have with the system for that particular story.

2.3 Integration Testing

User Story 4: Integration of Database + Reports + UI

Test Case	Input Values	Driver Functions	Expected Outcomes	Observed Outcomes

Test the function of the /day router	Click the day button	getReports(), router.get('/day',	The page should display all the recorded recyclables from the past day	The recyclables from the last day are displayed
Test the function of the /week router	Click the week button	getReports(), router.get('week',	The page should display all the recorded recyclables from the past week	The recyclables from the last week are displayed
Test the function of the /month router	Click the month button	getReports(), router.get(['/', '/month']	The page should display all the recorded recyclables from the past month	The recyclables from the last month are displayed
Test the function of the /year router	Click the year button	getReports(), router.get('year',	The page should display all the recorded recyclables from the past year	The recyclables from the last year are displayed

While creating the reports package there were several issues with the database. Two of the tables had their foreign keys created incorrectly. For each table the ‘id’ attribute of the table was referencing the ‘id’ attribute of another table. The structure of each table had to be changed so that the attribute intended to be a foreign key reference was pointing to the ‘id’ attribute of another table.

User Story 6: Integration of Database + Public Reports + UI

Test Case	**Input Values**	**Driver Functions**	**Expected Outcomes**	**Observed Outcomes**
Test the function of the /day router	Click the day button	getReports(), router.get('/day',	The page should display all the recorded recyclables from the past day	The recyclables from the last day are displayed

Test the function of the /week router	Click the week button	getReports(), router.get('week',	The page should display all the recorded recyclables from the past week	The recyclables from the last week are displayed
Test the function of the /month router	Click the month button	getReports(), router.get(['/', '/month']	The page should display all the recorded recyclables from the past month	The recyclables from the last month are displayed
Test the function of the /year router	Click the year button	getReports(), router.get('year',	The page should display all the recorded recyclables from the past year	The recyclables from the last year are displayed

While creating the router for the public reports page there were some issues with getting the navigation bar to display correctly as well as issues with the links associated with the buttons. Originally we did not intend to have a nav bar on the public reports page and we tried several methods to remove it from the page. Initially when the user would navigate to the page none of the information was loaded or displayed. We discovered that the cause behind this was that the {{{body}}} tag associated with the handlebars engine was left out of the layout. In the end we decided that this we changed it to look like the rest of the website/UI.

User Story 7: Public Reports + UI

Test Case	**Input Values**	**Driver Functions**	**Expected Outcomes**	**Observed Outcomes**
Test the /day table	Click the day button	getReports(), router.get(‘/day’,	The page displays a table at the top of the page outputting the number of items sorted with tags from the last day	The table is correctly displayed at the top of the page
Test the /week table	Click the week button	getReports(), router.get(‘/week’,	The page displays a table at the top of the page outputting	The table is correctly

			the number of items sorted with tags from the last week	displayed at the top of the page
Test the /month table	Click the month button	getReports(), router.get([‘/’, ‘/month’],	The page displays a table at the top of the page outputting the number of items sorted with tags from the last month	The table is correctly displayed at the top of the page
Test the /year table	Click the year button	getReports(), router.get(‘/year’,	The page displays a table at the top of the page outputting the number of items sorted with tags from the last year	The table is correctly displayed at the top of the page

User Story 15: Supervisor Warnings

Test Case	**Input Values**	**Driver Functions**	**Expected Outcomes**	**Observed Outcomes**
Test the /supervisor page	Click the supervisor on nav bar	getEmployees(), getMachines(), router.get(‘/’,	The understaffed machines are displayed at the top	The correct machines are displayed
Check correct machines are showing	Page load	getEmployees(), getMachines(), router.get(‘/’,	The machines are updated	The new machines that are understaffed are displayed

2.4 Refactoring

The refactoring for this week mostly involved reorganizing the database so that the new webpages implemented by this weeks user stories could correctly access the data. These changes to the database allowed us to correctly gather the data so that we could display the relevant information.

There were multiple foreign keys among a few different tables that had to be reassigned, to clean up the relationships and allow the database to be more easily used. Initially, idRecyclable in statistics referenced the id of recyclable and idType in recyclable referenced the id of types. These relationships were removed and new foreign keys were established to more clearly state the relationships between these tables.

On User Story 7, a small change was made to the view created by some of the earlier user stories from this week. A small table was appended to the top of the view, so that User Story 7 could be correctly displayed.

With User Story 15, a new table called Employees was added to the database to include the information pertaining to employees and their relationships with their location of employment. This table was used in User Story 15 so that an employee could be assigned to a location. This relationship was then used to determine whether a location had the correct number of employees assigned to it.

3. Week Three (and Beyond) Implementation Plan

In the following weeks the application will be developed further into a fully functional system. This will include executing the remainder of the user stories as well as implementation and testing of the hardware. The following user stories are left to be developed.

User Story Eight:

As the operator of the sorting facility, I want the sorting machine to distinguish tagged recyclables from other waste so that I can use fewer people to hand-sort.

The task will be broken into the following steps:

- Create all packages pertaining sensors, conveyors, robots and the sorting gate
- Create a hardware IO package and RFID database
- Set up a server and allow it to interface and control all of these packages

User Story Nine:

As a member of the public, I want an easily understood infographic detailing what materials are recyclable and examples of everyday products

The task will be broken into the following steps:

- Collect information on what brands/types of products have RFID chips

- Create a website for the public about RFID recyclable sorting technology
- Publish information on the website

User Story Ten:

As a supervisor, I want the ability to easily track what changes have been made to the database and by whom those changes were made

The task will be broken into the following steps:

- Create the Operator Controls package and give it the ability to query the RFID database and display the requested information
- Create the User Interface package with the ability to navigate to a database editing page populated with data from the Operator Controls package
- Create a Login package and track logins and corresponding changes in the database
- Create the Main Display Page package with a supervisor sub page and populate it with information about changes made to the DB

User Story Eleven:

As an IT professional I want to be sure that all of the data is secure and that critical systems are not accessible from outside the facility

The task will be broken into the following steps:

- Find network firewall vendor.
- Implement vendor firewall to block any anyone without permission from accessing critical systems

User Story Twelve:

As a technician I want to be able to disable and depower individual stations for maintenance without shutting down the entire line.

The task will be broken into the following steps:

- First set up the Conveyor Motion package
- Create the Hardware IO package and give it the ability to send signals to the Conveyor Motion package so it can start and stop the motion of the belt.
- Create the User Interface package with the OperatorControls sub package
- Give the OperatorControls package buttons to disable and depower individual stations.

User Story Thirteen:

As a technician I want to be able to track data from actuators to be able to identify potential failures and make repairs before shutting the line down

The task will be broken into the following steps:

- Create the Diagnostics display package and route all alarms received from the system to variables contained in the package

- Create the User Interface package with the ability to navigate to a Diagnostic page that would give information about current alarms populated with data from the Diagnostics Display package

4. Customer Interaction

4.1 Customer Communication and Interaction

No customer interaction was attempted this week.