Test Plan:
On-Premises Utilities Management Dashboard

Nicole Hessner
IP03, CS504 – Software Engineering,
Masters of Science in Computer Science,
School of Computing and Technology, City University of Seattle
hessnernicole@cityuniversity.edu

Executive Summary

This test plan is for an on-premises utilities management system, intended for industrial applications with on-site utility generation. On-site utilities can include steam, hydronic heating water, chilled water, and compressed air, in addition to usage of utilities generated off-site, such as electricity and natural gas. To maximize value to the customer, software development will be based on a Test-Driven Development approach, as pre-developing the test cases will allow the software development team to remain focused on the original software requirements, as interpreted through the tests. The philosophy of this test plan loosely follows that of James Whittaker's "10 Minute Test Plan" on the Google Testing Blog. The ideas generated from the 10 Minute Test Plan are then integrated into more traditional test case tables, focusing broadly on Unit Testing, Integration Testing, System Testing, and User Acceptance Testing.

Keywords: software testing, Test-Driven Development, 10 Minute Test Plan, user acceptance testing {5}, {6}, {7}

1. TESTING OVERVIEW

Approach
Whittaker's 10 Minute Test Plan from Google acts as the initial brainstorming phase for this test plan. His three major categories are Attributes, Components, and Capabilities. His approach to test plan development is analogous to Agile versus Waterfall for software project management: having a plan is important, but getting stuck in the details of documenting a plan often results in wasted effort. Requirements change, especially for a project like this one, which includes interfacing with several existing data streams, not all of which are self-contained in a readable database. Since requirements are likely to change as this project evolves, it is prudent to focus on the core work without getting too deep into the details (Whittaker, 2011).

Testing for this software will best be accomplished using a combination of Path Analysis and Error Prone Analysis, in addition to some intuitive test requirements. This piece of software has a moderate degree of complexity, so using Path Analysis will help ensure that all desired functionality is adequately tested (Tsui et al., 2022).

There will also be some Error Prone Analysis, due to a high likelihood that, during the development process, the users and software team will find that existing systems that were expected to interface with the software are not adequate to provide the functionality initially envisioned. One of the major requirements of this project is that additional metering cannot be installed, so this is a likely risk to project scope and schedule. Due to the ages and types of existing monitoring, the integrity of the data will need to be validated to determine its ultimate usefulness. Using the review process and inspections

to determine which requirements continuously contain defects will be useful in identifying which requirements may need to be altered to create the best possible functionality and user experience from existing data streams (Thumati et al., 2020).

Finally, some of the testing will be in the form of code inspections by a team of software developers and testers. They will focus on finding defects, but not fixing them. Some errors will be seeded into the software, in order to approximate how many wild bugs still remain in the system (Tsui et al., 2022).

Philosophy

This test plan will first and foremost utilize Whittaker's 10 Minute Test Plan (2011). I want to ensure that the focus of testing remains on capabilities, as driven by overall program attributes and components. This ethos will be combined with Test-Driven Development (TDD), to ensure that the features that are developed reflect the requirements of the end-user (Tsui et al., 2022). When I was participating as a user in a similar development project, a common occurrence was for other groups within the business to try to insert themselves into the project and insert their own requirements. This lengthened the development process significantly. It also resulted in a program that partially met the requirements of several user groups, instead of having a few core functions that received adequate focus during the development process (Thumati et al., 2020). For that reason, I hope that using TDD will act to minimize scope creep and speculative development by reinforcing the need for a change management process when the customer comes with new, possibly unrelated requirements (Tsui et al., 2022).

2. TEN MINUTE TEST PLAN

Attributes

- User experience: simple, easy to navigate, clear graphics, English language descriptions for menus. Display information, not data.
- Timely interfacing: data must be up-to-date and meaningful.
- Have the raw data available for download for technical users.

Components

CLASSES

- Priority: display priority list, calculate priority
- Parameters: set cost parameters, set energy conversion parameters, set master priority parameters
- Cost: calculate cost, display sorted cost list
- Utility Monitoring Data: import utility monitoring data
- Equipment Monitoring Data: import equipment monitoring data
- Maintenance Data: import maintenance data
- Utility Usage: calculate cost, display utility usage list
- Dashboard: view webpage

MODULE NAMES

- Data Integration: Maintenance Data, Utility Usage, Equipment Monitoring
- Cost Module: maintenance costs, electricity costs, natural gas costs, steam costs, chilled water costs, compressed air costs
- Utility Module: maintenance time, electricity usage, natural gas usage, steam usage, chilled water usage, compressed air usage
- Reporting Module: get raw data, get reporting slide

Capabilities

USER ACTIONS

- Go to Dashboard
- Check priority list
- Filter priority list by: estimated maintenance hours saved, money saved, utilities saved
- Set cost parameters
- Set utility efficiency parameters
- Set unit conversion rate parameters

- View parameters
- Generate and share savings report
- Download raw data

2. UNIT TEST CASES

The template for the test cases that follow are from Hamilton (2022).

Test Scenario	Test Case	Test Data	Expected Result	Actual Results	Pass/Fail
TU01	Open the dashboard and view homepage.	URL	Dashboard opens and has a homepage		
TU02	Calculate individual utility cost	1. Cost_per_MMBtu =$4.50 2. NG_MMBtu=10,000	$45,000.00		
TU03	Display utility consumption	Test spreadsheet of utility data	Display utility data pulled from static spreadsheet		
TU04	Calculate maintenance cost	1. Maintenance job list (time job opened) 2. Cost/maintenance hour = $150	Time in hours since the job was open times $150.		
TU05	Display cost list	1. URL for cost list 2. Cost data	Cost list that can be filtered.		
TU05					

3. INTEGRATION TEST CASES

Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Results	Pass/Fail
TI01	Import Maintenance Data	1. Connect to maintenance database. 2. Verify that data is available for download. 3. Check methods that use maintenance data (cost priority). 4. Write to spreadsheet. 5.	Maintenance database.	Maintenance data is successfully pulled and downloaded from dashboard in a human-readable format.		
TI02	Import Equipment Monitoring Data	1. Connect to HVAC monitoring system 2. Pull temperature setpoints by HVAC unit. 3. Pull schedules by HVAC unit. 4. Write to spreadsheet.	HVAC monitoring system	Equipment monitoring data is successfully pulled and downloaded from dashboard in a human-readable format.		

TI03	Import Utility Monitoring Data	1. Connect to compressed air meter database. 2. Pull data for last two hours. 3. Write to spreadsheet.	Compressed air meter data.	Compressed air data is available for download and is from the last two hours.		
TI04	Cost Module	1. Calculate natural gas cost. 2. Calculate electricity cost. 3. Calculate average maintenance hours for similar work from maintenance data. 4. Add results	1. maintenance job list 2. electricity cost 3. natural gas cost	Costs are available by job and can be sorted.		
TI05	Utility Module	1. Pull data from available utilities. 2. Pull maintenance jobs. 3. Calculate utility usage for each utility for each hour the job has been open. 4. Sort jobs by utility selected.	1. utility data 2. maintenance job time opened	Utility usage is available by job and can be sorted.		
TI06	Reporting Module	1. Generate summary. 2. Write summary to spreadsheet. 3. Click button to download summary.	1. priority list 2. utility usage 3. utility cost	Spreadsheet contains summary of top 10 maintenance jobs, utility usage for each utility, and cost per utility and total		

4. SYSTEM TEST CASES

Test Scenario	**Test Case**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Results**	**Pass/Fail**
TS01	Live Data Pull	Get and display live data from equipment monitoring and utility metering systems.	1. Compressed air monitoring data 2. Maintenance data 3. Steam plant data 4. Natural gas meters	Data displays live on the dashboard for the systems it is available for		
TS02	Calculate priority	Ensure all data and parameters are integrated to create a single priority list.	All cost and utility usage data.	Priority list is generated showing top ten maintenance jobs and sort.		

5. ACCEPTANCE TEST CASES

Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Results	Pass/Fail
TA01	UI Ease of Navigation	1. Have user navigate to dashboard. 2. Ask user to identify and interact with menu. 3. Go to each data display page. 4. Test raw data downloads with Energy Manager.	Dashboard	Satisfaction with menu and user interface.		
TA02	Set Parameters	1. Change static parameters	User input parameters	Parameters successfully update in the formulas that use them.		

7. REFERENCES

Hamilton, T. (2022, April 2). How to Write Test Cases: Sample Template with Examples. Guru99. Retrieved May 22, 2022, from https://www.guru99.com/test-case.html#the-format-of-standard-test-cases

Thumati, B. T., Siva Subramania, H., Shastri, R., Kalyana Kumar, K., Hessner, N., Villa, V., Page, A., & Followell, D. (2020). Large-scale Data Integration for Facilities Analytics: Challenges and Opportunities. *2020 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/bigdata50022.2020.9378440

Tsui, F., Karam, O., & Bernal, B. (2022). Testing and Quality Assurance [E-book]. In Essentials of Software Engineering (3rd ed., p. 199). Jones & Bartlett Learning. Retrieved May 22, 2022, from https://learning.oreilly.com/library/view/essentials-of-software/9781449691998/chapter-14.html

Whittaker, J. (2011, September 1). *The 10 Minute Test Plan*. Google Testing Blog. Retrieved May 22, 2022, from https://testing.googleblog.com/2011/09/10-minute-test-plan.html