MCMASTER UNIVERSITY

CAS 4ZP6

TEAM 9

CAPSTONE PROJECT 2013/2014

PORTER SIMULATION

Test Report Revision 0

Authors:
Vitaliy Kondratiev - 0945220
Nathan Johrendt - 0950519
Tyler Lyn - 0948978
Mark Gammie - 0964156

Supervisor:
Dr. Douglas Down

March 18, 2014

CONTENTS

1 REVISION HISTORY

Revision #	Author	Date	Comment
1	Nathan Johrendt	March 17	Test Report Template Complete
2	Vitaliy Kondratiev, Nathan Johrendt, Tyler Lyn, Mark Gammie	March 17	Test Plan Updates
3	Vitaliy Kondratiev, Nathan Johrendt, Tyler Lyn, Mark Gammie	March 18	Test Plan Updates

2 INTRODUCTION

This testing report is the realization of the test plan written for the Porter Simulation. It contains descriptions and output data for the most important tests from the test plan. The majority of the following test cases are dynamic; many have static versions found within the test plan, but their results are not as valuable, and so they have been omitted.

3 SYSTEM TEST REPORTS

3.1 INPUT/INITIALIZATION CORRECTNESS 1

Once input values are loaded by the simulation, it has become initialized with modified values. Before beginning simulation, the simulation will re-print the information it just imported to confirm that it has found the right information. A simple check to ensure data is being handled correctly during import. While there are not many variables to be tested now, several will be added before the final implementation is complete. This test deals primarily with the Input Module which traces to requirements section 8.1.

 (i) **Initial State:** Uninitialized Simulation

 (ii) **Input:** Simulation input variables from interface, as well as a file location for both data.csv and schedule.csv for importing job data and porter shift information respectively. See Figure 3.1: User Interface below.

 (iii) **Expected Output:** Simulation outputs variable data identical to that defined by the interface.

 (iv) **Actual Output:** duration = 86400.0
    statsource = "C:\ Users\ Nathan\ Documents\ GitHub\ Porter-Simulation\ Simulation Prototype"
    shiftsource = "C:\ Users\ Nathan\ Documents\ GitHub\ Porter-Simulation\ Simulation Prototype"
    outputloc = "C:\ Users\ Nathan\ Documents\ GitHub\ Porter-Simulation\ Simulation Prototype"

3.2 EVENT LIST CORRECTNESS 1

As the simulation runs, when an event is dispatched to a porter, it will then have fewer jobs left to assign. With 30 porters on a 24 hour schedule, no jobs will remain undone, and any that remain will represent an error to examine. This test involves the simulation core, dispatch and export modules.

 (i) **Initial State:** Simulation event list is generated by initializing the simulation through the interface

 (ii) **Input:** Event list is generated from jobs stored in the data.csv file. The simulation will be configured to add jobs for one day, with 30 porters scheduled 24 hours a day. See Table 3.1: Porter Schedule in schedule.csv below.

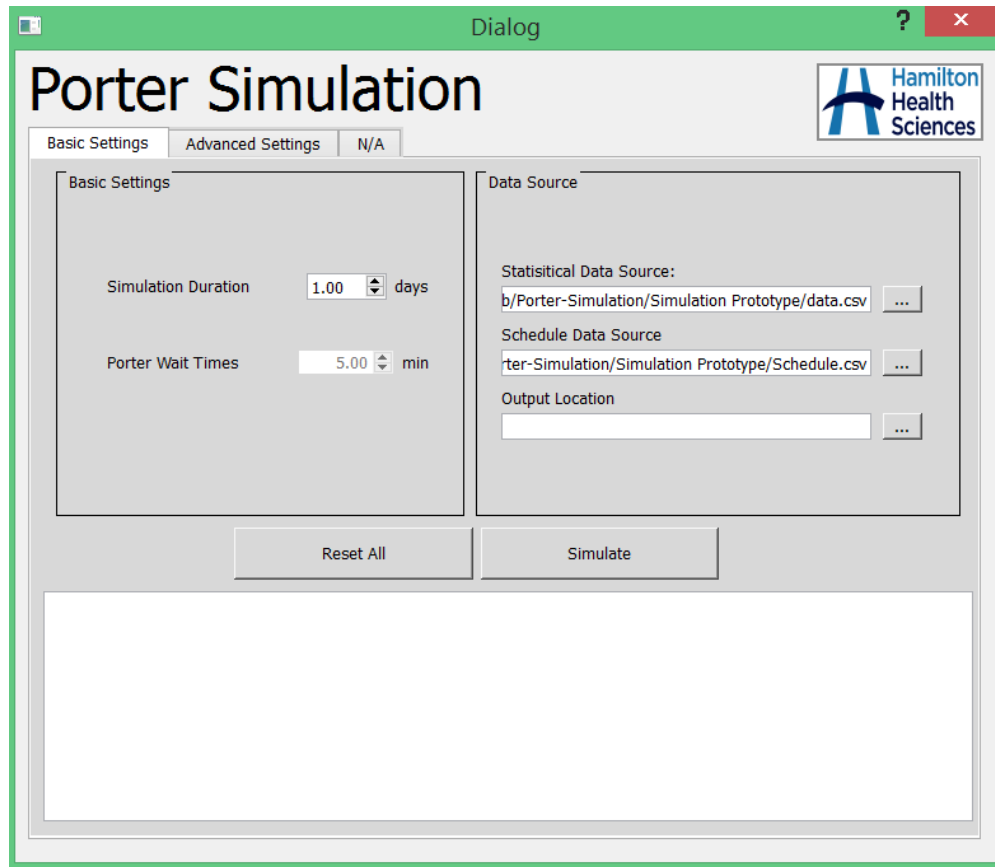 (iii) **Expected Output:** Conclude that no incomplete jobs remain

Figure 3.1: User Interface

Table 3.1: Porter Shift Information in schedule.csv for Event List Correctness 1

Shift ID	StartTime	EndTime	Porter Ids	Day
0	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	0
1	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	1
2	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	2
3	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	3
4	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	4
5	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	5
6	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	6

Table 3.2: Porter Shift Information in schedule.csv for State Change Correctness 1

Shift ID	StartTime	EndTime	Porter Ids	Day
0	12:00 AM	11:59 PM	0,1,2,3,4	0

(iv) **Actual Output:**
Job145: E2 Orthopedics Room 07 Bed 03 -> E3 Nurse Station (3497)
Porter28 is dispatched Job147: Emergency Department - GRIDLOCK (2286) -> E3 Nurse Station (3497)
Porter28 is in-progress Job147: Emergency Department - GRIDLOCK (2286) -> E3 Nurse Station (3497)
Porter20 is complete Job148: Emergency Department - GRIDLOCK (2286) -> F4 Surg Oncology (3501)
Job144: E2 Orthopedics Room 15 Bed 02 -> E3 Nurse Station (3497)
Porter29 is dispatched Job145: E2 Orthopedics Room 07 Bed 03 -> E3 Nurse Station (3497)
Porter1 is pending
Porter29 is in-progress Job145: E2 Orthopedics Room 07 Bed 03 -> E3 Nurse Station (3497)
Porter24 is complete Job147: Emergency Department - GRIDLOCK (2286) -> E3 Nurse Station (3497)
Job150: Emergency Department Bed 01 -> F4 Surg Oncology (3501)
Porter24 is dispatched Job144: E2 Orthopedics Room 15 Bed 02 -> E3 Nurse Station (3497)
Porter28 is in-progress Job144: E2 Orthopedics Room 15 Bed 02 -> E3 Nurse Station (3497)
Porter24 is pending
Porter28 is complete Job144: E2 Orthopedics Room 15 Bed 02 -> E3 Nurse Station (3497)
Porter29 is complete Job145: E2 Orthopedics Room 07 Bed 03 -> E3 Nurse Station (3497)
Porter25 is pending
Porter28 is pending
Porter29 is pending
*****SIMULATION COMPLETE*****

3.3 STATE CHANGE CORRECTNESS 1

With only five Porters working, they will be required to change state frequently as they complete all of a regular day's tasks understaffed. Any problems, beyond the obvious delays, will result in a Porter getting stuck with a job at a particular state. This will be recorded in the output data and can be addressed. This test deals exclusively with the simulation core, dispatch and export modules.

(i) **Initial State:** Simulation event list is generated by initializing the simulation through the interface

(ii) **Input:** Event list is generated from jobs stored in the data.csv file. Porter schedule is generated from the schedule.csv file. The simulation will be configured to add jobs for one day, with five porters scheduled 24 hours that day.

(iii) **Expected Output:** All Porters complete jobs dispatched to them successfully until the simulation duration expires

(iv) **Actual Output:**
Job6: CDU Express Bed 05 -> X-Ray Core - B1 (2245)
Job10: Ultrasound (2276) -> F3 Medicine Room 12 Bed 02
Porter2 is dispatched Job6: CDU Express Bed 05 -> X-Ray Core - B1 (2245)
Porter2 is in-progress Job6: CDU Express Bed 05 -> X-Ray Core - B1 (2245)
Porter1 is complete Job3: X-Ray Core - B1 (2245) -> C3 Oncology Room 26 Bed 01
Porter4 is pending
Porter0 is complete Job4: F4 Surg Oncology Room 12 Bed 02 -> Ultrasound (2276)
Porter2 is complete Job6: CDU Express Bed 05 -> X-Ray Core - B1 (2245)
Job38: E3 Medicine Room 02 Bed 04 -> MRI (2778)
Porter4 is dispatched Job10: Ultrasound (2276) -> F3 Medicine Room 12 Bed 02
Job37: Ultrasound (2276) -> C3 Oncology Room 31 Bed 01

Table 3.3: Porter Shift Information in schedule.csv for Porter/Event Linkage Correctness 1

Shift ID	StartTime	EndTime	Porter Ids	Day
0	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	0

Porter3 is dispatched Job38: E3 Medicine Room 02 Bed 04 -> MRI (2778)
Job33: C3 Oncology Room 07 Bed 01 -> X-Ray Core - B1 (2245)
Porter1 is dispatched Job37: Ultrasound (2276) -> C3 Oncology Room 31 Bed 01
Job39: F4 Surg Oncology Room 05 Bed 02 -> PACU (3415)
Porter0 is dispatched Job33: C3 Oncology Room 07 Bed 01 -> X-Ray Core - B1 (2245)
Porter3 is in-progress Job38: E3 Medicine Room 02 Bed 04 -> MRI (2778)
*****SIMULATION COMPLETE*****

3.4 STATE CHANGE CORRECTNESS 3

(i) **Initial State:** Initialized simulation

(ii) **Input:** porterUnit.py unit testing file

(iii) **Description:** Through the use of UnitTest in python, this test checks that a porter starts in a pending state and is able to iterates through all states when a porter is given a job.

(iv) **Expected Output:** The porter is in the pending state after work is complete.

(v) **Actual Output:** Run the python module: python porterUnit.py

.
———————————————————————————-

Ran 1 test in 0.001s

OK

3.5 PORTER/EVENT LINKAGE CORRECTNESS 1

Every time an Event moves from pending to dispatched, a Porter must be linked to that event. This test is to ensure that all pairings of job and Porter are unique, and that after completing a job, a porter will return to "available" and continue accepting new jobs. This test involves the simulation core, dispatch and export modules.

(i) **Initial State:** Simulation event list and porter schedule are generated by initializing the simulation through the interface

(ii) **Input:** Event list is generated from jobs stored in the data.csv file. Porter schedule is generated from the schedule.csv file. The simulation will be configured to add jobs for one day, with 30 porters scheduled 24 hours that day.

(iii) **Expected Output:** Porter and Event are linked together uniquely

(iv) **Actual Output:** Porter23 is complete Job13: CDU Express Bed 05 -> Ultrasound (2276)
Porter17 is pending
Porter27 is complete Job12: E3 Medicine Room 02 Bed 01 -> Ultrasound (2276)
Porter21 is pending
Porter26 is pending
Porter24 is pending
Porter28 is complete Job18: CT - B1 (3023) -> E3 Medicine Room 01 Bed 01 Porter25 is complete Job26: Ultrasound (2276) -> E3 Medicine Room 02 Bed 01 Porter23 is pending
Porter29 is complete Job24: Ultrasound (2276) -> CDU Express Bed 05 Porter27 is pending

Table 3.4: Porter Shift Information in schedule.csv for Task Pool Correctness 1

Shift ID	StartTime	EndTime	Porter Ids	Day
0	12:00 AM	11:59 PM	0	0

Porter25 is pending
Porter28 is pending
Porter29 is pending
*****SIMULATION COMPLETE*****

3.6 TASK POOL CORRECTNESS 1

With only a single porter active, the majority of the jobs added by the simulation will simply be added to the dispatcher and wait. This test is to ensure that the dispatcher holds incomplete jobs correctly and that they are reported after the simulation completes. This test involves the simulation core, dispatch and export modules.

 (i) **Initial State:** Simulation event list and porter schedule are generated by initializing the simulation through the interface

 (ii) **Input:** Event list is generated from jobs stored in the data.csv file. Porter schedule is generated from the schedule.csv file. The simulation will be configured to add jobs for one day, with one Porter scheduled 24 hours a day.

(iii) **Expected Output:** Simulation output properly stores many incomplete job entries.

(iv) **Actual Output:**
Job134: X-Ray Core - B1 (2245) -> F5 ALC Room 01 Bed 01
Porter0 is pending
Porter0 is dispatched Job33: C3 Oncology Room 07 Bed 01 -> X-Ray Core - B1 (2245)
Porter0 is in-progress Job33: C3 Oncology Room 07 Bed 01 -> X-Ray Core - B1 (2245)
Job137: CT - B1 (3023) -> Intensive Care Unit [ICU] (3486)
Job141: Emergency Department Bed 10 -> C4 Haematology Nurse (3396)
Job135: X-Ray Core - B1 (2245) -> E4 Room 06 Bed 01
Job117: E4 Room 06 Bed 03 -> X-Ray Core - B1 (2245)
Job142: C4 Haematology Room 52 Bed 01 -> CT - B1 (3023)
Job146: Emergency Department - GRIDLOCK (2286) -> E3 Medicine CTU (9300)
Job143: Emergency Department - GRIDLOCK (2286) -> E3 Medicine CTU (9300)
Porter0 is complete Job33: C3 Oncology Room 07 Bed 01 -> X-Ray Core - B1 (2245)
Job148: Emergency Department - GRIDLOCK (2286) -> F4 Surg Oncology (3501)
Job147: Emergency Department - GRIDLOCK (2286) -> E3 Nurse Station (3497)
Job145: E2 Orthopedics Room 07 Bed 03 -> E3 Nurse Station (3497)
Porter0 is pending
Porter0 is dispatched Job39: F4 Surg Oncology Room 05 Bed 02 -> PACU (3415)
Job144: E2 Orthopedics Room 15 Bed 02 -> E3 Nurse Station (3497)
Porter0 is in-progress Job39: F4 Surg Oncology Room 05 Bed 02 -> PACU (3415)
Job150: Emergency Department Bed 01 -> F4 Surg Oncology (3501)
Porter0 is complete Job39: F4 Surg Oncology Room 05 Bed 02 -> PACU (3415)
Porter0 is pending
Porter0 is dispatched Job11: E3 Medicine Room 01 Bed 01 -> CT - B1 (3023)
Porter0 is in-progress Job11: E3 Medicine Room 01 Bed 01 -> CT - B1 (3023)
*****SIMULATION COMPLETE*****

Table 3.5: Porter Shift Information in schedule.csv for Termination Correctness 1

Shift ID	StartTime	EndTime	Porter Ids	Day
0	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	0
1	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	1
2	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	2
3	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	3
4	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	4
5	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	5
6	12:00 AM	11:59 PM	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29	6

3.7 TASK POOL CORRECTNESS 3

Through the use of UnitTest in python, this test checks that the dispatcher correctly adds a new job to the Task Pool if there are no available Porters. Three specific variables are tested for: Priority Weight, Appointment Factor and the length of the pending jobs list. This test deals exclusively with the dispatcher module.

   (i) **Initial State:** Initialized simulation

  (ii) **Input:** dispatcherUnit.py unit testing file

 (iii) **Expected Output:** Conclude that the Task Pool contains one additional event as the number of pending jobs has increased by one.

 (iv) **Actual Output:** Run the python module: python dispatcherUnit.py

      ...
      ———————————————————————————————————-

      Ran 3 tests in 0.003s

      OK

3.8 TERMINATION CORRECTNESS 1

After accepting all inputs and initializing, the simulation will compute five days of operational data and output the results. This test involves all modules.

   (i) **Initial State:** Simulation is initialized through the interface

  (ii) **Input:** Event list is generated from jobs stored in the data.csv file. Porter schedule is generated from the schedule.csv file. The simulation will be configured to add jobs for five days, with 30 porters scheduled 24 hours a day.

 (iii) **Expected Output:** Simulation outputs that it has completed each step and has written the results to the dashboard.

 (iv) **Actual Output:**
      *****SIMULATION COMPLETE*****
      *****WRITING OUTPUT*****
      *****SAVING OUTPUT*****
      *****BUILDING DASHBOARD*****
      *****DASHBOARD COMPLETE*****

3.9 COMPATIBILITY TEST 1

The simulation is run and compared to results computed on other operating systems to ensure it is functioning normally in the new environment. The stakeholders have indicated that they will only be using the software on versions of windows. The current implementation utilizes cxfreeze to create windows executable files.

(i) **Initial State:** Unknown Operating System with the simulation accessible on local storage

(ii) **Input:** 'Termination Correctness 1' simulation parameters (details outline previously under 'Termination Correctness 1')

(iii) **Expected Output:** Simulated results are consistent with previously generated values

(iv) **Actual Output:** Currently, the simulation has been tested on a number of systems, including windows 7 and 8, and the executables produced by cxfreeze have performed perfectly. Final output from each simulation compatibility test has been:
*****SIMULATION COMPLETE*****
*****WRITING OUTPUT*****
*****SAVING OUTPUT*****
*****BUILDING DASHBOARD*****
*****DASHBOARD COMPLETE*****

4 NONFUNCTIONAL TEST REPORTS

4.1 USABILITY TEST 1

End user is provided with a set of instructions from the user manual on how to initialize and run the simulation. The success of this test is determined by how much external assistance the End User requires from the development team on their first use of the software.

(i) **Initial State:** Simulation prior to execution

(ii) **Input:** End user with a copy of both the simulation software package and accompanying user manual.

(iii) **Expected Output:** The end user successfully Initializes the Simulation

(iv) **Actual Output:** The simulation is yet to be formally tested with stakeholders in regards to the scope of this test case. However, feedback has been gathered during project demos with a few key stakeholders, and the responses have been positive. This test will become critical during the very final stages of development, as the interface for the project is being tailored to the stakeholder's needs.

4.2 PERFORMANCE TEST 1

The project stakeholders have placed a loose time limit on the duration of execution for a single simulation run of fifteen minutes. No correct simulation execution has taken longer than five minutes, with most between one and two minutes, but ensuring this general timing for the simulation across all applicable systems is the purpose of this test.

(i) **Initial State:** Simulation prior to execution

(ii) **Input:** Any combination of inputs that the interface will accept and begin simulating

(iii) **Expected Output:** The simulation completes execution taking between thirty seconds and five minutes, repeated thrice for consistency.

(iv) **Actual Output:**

Table 4.1: Simulation Execution Times for Performance Test 1

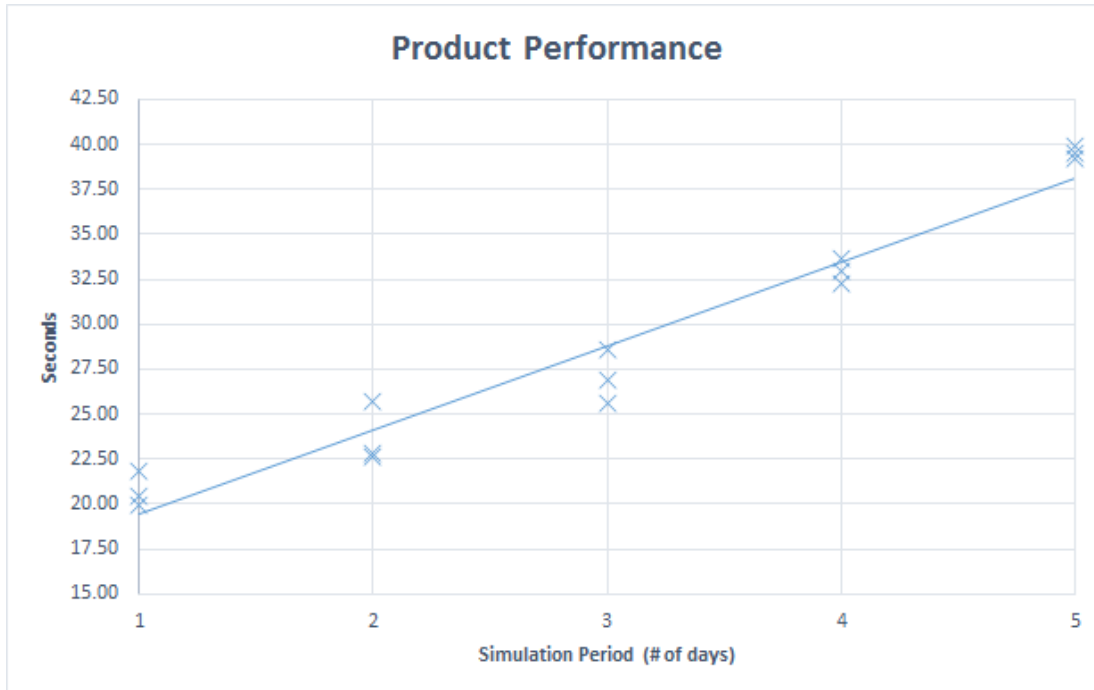| Simulation Run	Simulation Period (# of days)	Duration of Execution (seconds)
1	1	19.945
2	1	21.861
3	1	20.442
4	2	25.713
5	2	22.823
6	2	22.592
7	3	25.569
8	3	28.569
9	3	26.850
10	4	33.673
11	4	32.964
12	4	32.206
13	5	39.469
14	5	39.942
15	5	39.157

Figure 4.1: Performance Report

5 SUMMARY

Over the past few weeks, our implementation has been dramatically overhauled, resulting in brand new methods for importing and information handling. At the current state of development, the previously defined automated testing techniques were no longer applicable, and have been omitted from this report. Approaching the final implementation, the project team anticipates that a number of automated tests will be re-added to the test plan and report.

These test cases, whether performed as a formal testing suite or quick samples during implementation changes, have provided useful information for repairing or improving the simulation software. Many of these tests have produced expected results without fail, but they provide essential security by repeatedly checking the core features of the software, ensuring accuracy throughout development. One test in particular, Task Pool Correctness 1, which overloads a single porter with many tasks to test the dispatcher's ability to store pending jobs, highlighted a major problem with how jobs were dispatched. Jobs from the beginning of a day weren't being handled by a porter until the very end, this led to an examination of the dispatcher module and the resolution of the issue.

6 FIGURES AND TABLES APPENDIX

(a) Figure 3.1: User Interface

(b) Table 3.1: Porter Shift Information in schedule.csv for Event List Correctness 1

(c) Table 3.2: Porter Shift Information in schedule.csv for State Change Correctness 1

(d) Table 3.3: Porter Shift Information in schedule.csv for Porter/Event Linkage Correctness 1

(e) Table 3.4: Porter Shift Information in schedule.csv for Task Pool Correctness 1

(f) Table 3.5: Porter Shift Information in schedule.csv for Termination Correctness 1

(g) Table 4.1: Simulation Execution Times for Performance Test 1

(h) Figure 4.1: Performance Report