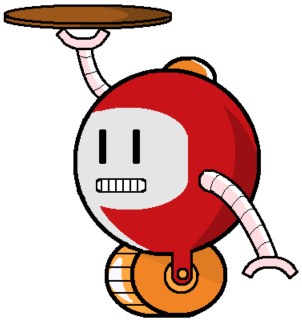

LazyBots

McMaster University

Draft Verification and Validation

SE 4GA6 & TRON 4TB6

Group 9

Karim Guirguis	001307668
David Hemms	001309228
Marko Laban	001300989
Curtis Milo	001305877
Keyur Patel	001311559
Alexandra Rahman	001305735

0 Contents

1 Revisions **3**

2 Purpose **4**

3 Scope **4**

4 Background **4**

5 Test Cases **4**

5.1 Testing Plan and Testing Factors . . . 4
5.2 Alfred Client Application . . . 5
5.3 Server System . . . 7
5.4 Alfred Manager System . . . 9
5.5 Alfred Drive Train System . . . 10
5.6 Alfred Image Processing System . . . 12
5.7 Alfred Pumping System . . . 14
5.8 Integration Testing . . . 15

6 Beyond Testing **16**

6.1 Code Walk-through and Code Reviews . . . 16

7 Supporting Material **16**

8 Traceability Matrices **16**

0 List of Tables

1 LazyBots Table of Revisions . . . 3
10 Alfred Client Application Traceability . . . 16
11 Server System Traceability . . . 16
12 Alfred Manager System Traceability . . . 16
13 Alfred Drive Train System Traceability . . . 17
14 Alfred Image Processing System Traceability . . . 17
15 Alfred Pumping System Traceability . . . 17

1 Revisions

Date	Revision Number	Authors	Comments
February 17th, 2018	Revision 0	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	-
April 12th, 2018	Revision 1	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	Updated some test case results and changed the traceability matrix to be multiple ones, for clarity.

Table 1: LazyBots Table of Revisions

2 Purpose

The purpose of the document is to help support the development process of the drink dispensing autonomous robot known as Alfred. In particular, this document will describe what tests have been performed to ensure that Alfred will be able to perform the act of navigating to and pouring drinks correctly.

3 Scope

This document will focus on different software systems that have been discussed within the system design document. This document will focus on performing tasks on the system as if it was a black box, meaning the important aspects are the inputs and outputs of the system rather then the internal mechanics. For smaller sub-features, depending on the implementation, the code may have been considered when creating test cases to ensure the system is robust.

4 Background

Alfred is designed to work with various different software systems, so that it can receive the desired drinks of the user, and be able to determine based on internal knowledge of the restaurant to be able to navigate by itself to the correct table. Alfred will then have internal containment units for beverages that will allow the user to put there cup into the drink dispenser to have Alfred pour there drink for them.

Alfred will be able to receive the drink that the user that the user request by requesting the next drink from a designated server. This server will story any incoming drink requests form the mobile application. This mobile application is designed to be used on a tablet or mobile device, where the servers could sign in and give it a table identification number to allow Alfred know which table has sent this order.

Management staff will be able to view Alfred's status from the computers within the back. This system will give the management the ability to modify tables, view the errors that the robot has sent, and will allow the management staff to call Alfred back to the home base.

5 Test Cases

5.1 Testing Plan and Testing Factors

Testing was performed based on the systems and sub systems that were defined within the software. Testing was down within a bottom up approach where small unit functions were first tested. Depending on the complexity of this unit, either white box or black box testing would be chosen. If the unit was complex, white box testing would be used based to help testing based on branch coverage. Black box testing would then be used for larger integration testing. Testing was done over the following modules:

Component	Test Plan Test Factors
Client Application	The main goal of testing the client application is to ensure that the experience is seamless for the customer. This means testing was done beyond just correctness testing. The bulk of the testing for the client application consisted of Usability and user interface testing. The server test cases tested the sequence in which API endpoints are accessed. Negative test cases for the client make sure that only the correct sequence of API access can be exercised. Test numbers 01-05 focus on the pages which the staff will be interacting with (Login page and Settings page). Test number 06 tests that the customer cannot access the settings page without credentials. Test numbers 07-14 focus on testing all the components (buttons, number pickers, alerts and warnings) which the customer will be interacting with.

Server	The main goal for testing the server is to make sure that all endpoints are functioning properly and return the correct results.The tests were split up into groups, based on the users of the server. Test sets within the entire test suite focused on enpoints to be used by the admin, the customers, and the robot. Another important part of testing the server was the sequence in which the endpoints were exercised. With the use of tokens and passwords, it was important to run those tests first and ensure that they went smoothly in order to proceed forward. In order for a customer to place an order, the table at which they are sitting must authenticate with the server and will be provided with a token to be used in later communication.
Alfred Manager	Testing on this system was focused on unit operations working together. Test Cases were focused on Algorithms that would be used in terms of navigation. Other factors that were considered were the communication from the server to the robot as well as communication to the pumping system
Alfred Drive-train	The main goal of testing with the drive-train system is to ensure reliability for motion in terms of navigation. Do to this, there has been a heavy amount of testing with the encoder and the open loop PWM voltage that goes to the motor, to ensure that motion is reliable and that the feedback signal is correct. From this, components were tested within a feedback loop to ensure that proper movement could be performed. Finally, the ultra sonic sensors were tested both on and off the moving robot to ensure that the system is safe if the user or something is in front of it.
Alfred Image Processing	The focus of the image processing system testing is within detection of circles and there proper location.
Alfred Pumping System	The Alfred Pumping system focused on the mechanics of the pumps being able to receive and dispense the correct drinks, as well as detecting errors. In particular, the communication between the raspberry pi and the arduino, taking these values and dispensing the correct drink and ensuring that we are not out of liquids or that the liquids are at a non safe temperature.

5.2 Alfred Client Application

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
01	Testing Login with server down	NFR32	"admin:admin" as credentials	"Could not connect to server" pop up	"Could not connect to server" pop up	pass
02	Testing Login with server running	NFR32	"wrongUser: wrongPass" as credentials	"invalid credentials" pop up	"invalid credentials" pop up	pass
03	Testing Login with server running	NFR32	"admin: admin" as credentials	settings page is launched	settings page is launched	pass

04	Testing device-table mapping change (server down)	NFR5	Select different table number and click "Apply"	"Table change unsuccessful" pop up and table number reverts to old value	"Table change unsuccessful" pop up and table number does not revert to old value	pass
05	Testing device-table mapping change (server running)	NFR5	Select different table number and click "Apply"	Drinks list page is launched	Drinks page is launched	pass
06	Testing access to settings page after initial setup	NFR32	Press back after drinks page is launched	No observed change	no observed change	pass
07	Testing drink amount selection	NFR35 NFR21 NFR8 NFR5	Change the amount of sprite from 0 to 2	the number picker value will display 2	the number picker value displays 2	pass
08	Testing cart info	NFR5 NFR21 NFR30 NFR32	On the drinks list page select 1 Sprite and 2 cokes and click on "Go to cart"	current cart page is launched and the summary of items shows 1 sprite and 2 cokes	current cart page is launched and the summary of items shows 1 sprite and 2 cokes	pass
09	Testing sending order to server (server down , empty cart)	T01 T02 NFR15	have 0 drinks selected and click "send order" on the cart page	"Empty cart!" pop up	"Empty cart!" pop up	pass
10	Testing sending order to server (server down , non-empty cart)	T01 T02 NFR15	have some drinks selected and click "send order" on the cart page	"Order placement unsuccessful" pop up	"Order placement unsuccessful" pop up	pass

11	Testing sending order to server (server running , empty cart)	T01 T02 NFR15	0 drinks selected and click "send order" on the cart page	"Empty cart!" pop up	"Empty cart!" pop up	pass
12	Testing sending order to server (server running , non-empty cart)	T01 T02 NFR15	have some drinks selected and click "send order" on the cart page	"Order has been received by Alfred" pop up and place in line is displayed	"Order has been received by Alfred" pop up and place in line is displayed	pass
13	Testing cart reset after order placed(drink selection page)	N/A	send successful order and press "back" to go to drinks page	all drink amounts reset to 0	all drink amounts reset to 0	pass
14	Testing cart reset after order placed(cart page)	N/A	send successful order and press "back" to go to drinks page and press "go to cart"	cart is empty	cart is empty	pass

5.3 Server System

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
1	Test GET /drinks endpoint	N/A	N/A	res.body == drinkTypes	res.body == drinkTypes	pass
2	Test GET /numOfTanks endpoint	N/A	N/A	res.raw_body == 3	res.raw_body == 3	pass
3	Test POST /updateCreds endpoint	NFR33	New admin username/password pair	200	200	pass

4	Test POST /login endpoint	NFR33	Admin username/password pair	200	200	pass
5	Test POST /map endpoint	AD1/AD2	Text file containing map contents	200	200	pass
6	Test GET /map endpoint	AD1/AD2	N/A	File contents == Map received	File contents == Map received	pass
7	Test DELETE /drinks endpoint	N/A	Drink type	Missing COKE	Missing COKE	pass
8	Test POST /drinks endpoint	N/A	Drink type and tank number	COKE in body	COKE in body	pass
9	Test POST /table endpoint	NFR33	Table ID	"token" and "token_type" in body	"token" and "token_type" in body	pass
10	Test POST /placeOrder endpoint	TO1/TO2	Table ID and order contents in JSON	200	200	pass
11	Test GET /placeInLine endpoint	AF14	Order ID and table ID	res.raw_body == 1	res.raw_body == 1	pass
12	Test DELETE /cancelOrder endpoint	N/A	Order ID and table ID	200	200	pass
13	Test POST /placeOrder endpoint	TO1/TO2	Table ID and order contents in JSON	200	200	pass
14	Test GET /checkToken endpoint	NFR33	N/A	Must return validity of token	Includes validity of token	pass
15	Test GET /map endpoint	AF3	N/A	File contents == Map received	File contents == Map received	pass
16	Test GET /nextOrder endpoint	AF4	N/A	Order should have all required keys	Order has all required keys	pass

5.4 Alfred Manager System

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
1	Testing the ability to avoid obstacles and reach 2 tables 2x4 matrix 2 obstacles	AF3, AF10	map1.txt	[[(0, 1)], [(0, 2), (1, 2)]]	[[(0, 1)], [(0, 2), (1, 2)]]	pass
2	Testing 3 routes to a table and 1 route to another table 3x3 matrix 1 obstacle 2 tables	AF3, AF10	map2.txt	[[(1, 0), (2, 0), (2, 1), (2, 2), (1, 2)]	[(1, 2)]], [[(1, 0), (2, 0), (2, 1), (2, 2), (1, 2)], [(1, 2)]]	pass
3	Testing multiple routes to same table with no other tables. 3x3 matrix with 1 obstacle and 1 table	AF3, AF10	map3.txt	[[(0, 1), (0, 2)]]	[[(0, 1), (0, 2)]]	pass
4	Testing the order of reaching tables Tables traversed by row priority and column relative to home 4x5 matrix with 3 tables and 8 obstacles	AF3, AF10	map4.txt	[[(0, 1), (0, 2), (0, 3)], [(0, 2), (1, 2), (2, 2)], [(3, 2), (3, 1)]]	[[(0, 1), (0, 2), (0, 3)], [(0, 2), (1, 2), (2, 2)], [(3, 2), (3, 1)]]	pass
5	Testing whether an exception is thrown if table is unreachable. 2x3 matrix with 1 unreachable table, 2 obstacles	AF3, AF10	map5.txt	KeyError(" could not reach table")	pass	
6	Testing complex flow and tables in corners. 4x6 matrix with 3 tables and 7 obstacles	AF3, AF10	map6.txt	[[(1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (1, 3), (0, 3), (0, 4)], [(0, 3), (1, 3), (1, 2)], [(2, 2), (3, 2), (3, 3), (3, 4), (2, 4)]]	[[(1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (1, 3), (0, 3), (0, 4)], [(0, 3), (1, 3), (1, 2)], [(2, 2), (3, 2), (3, 3), (3, 4), (2, 4)]]	pass

7	Testing complex flow, tables in corners, multiple obstacles and tables 1 spot away from each other. 5x10 matrix with 6 tables and 18 obstacles	AF3, AF10	map7.txt	[[(1, 0), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (2, 7), (1, 7), (0, 7)], [(0, 8), (0, 9)], [(1, 9)], [], [(0, 9), (0, 8), (0, 7), (1, 7), (1, 6)], [(1, 7), (2, 7), (3, 7), (3, 6), (3, 5), (3, 4), (3, 3), (3, 2), (4, 2), (4, 1), (4, 0), (3, 0), (2, 0)]]	[[(0, 1), (1, 1), (2, 1), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (3, 2), (3, 3), (3, 4), (3, 5), (2, 5), (1, 5), (1, 6)], [(0, 6), (0, 7), (0, 8), (0, 9)], [(0, 8), (0, 7), (0, 6), (1, 6)], [(0, 6), (0, 7), (0, 8)], [(0, 9), (0, 10)], [(0, 9), (0, 8), (0, 7), (0, 6), (1, 6), (1, 5), (2, 5), (3, 5), (3, 4), (3, 3), (3, 2), (4, 2), (4, 1), (4, 0), (3, 0), (2, 0)]]	fail
8	receiving an order	A1	An order of Coke and Diet Coke on the User Application	Alfred receives a drink that corresponds to tank 1 and 2	Alfred receives a drink that corresponds to tank 1 and tank 2	Pass

5.5 Alfred Drive Train System

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
1	Testing Positive Change of Left Encoder	AF3	movement of encoder	60= 60+1	61	pass
2	Testing Negative Change of Left Encoder	AF3	movement of encoder	60= 60-1	59	pass
3	Testing Positive Change of Encoder	AF3	movement of encoder	60= 60+1	61	pass
4	Testing Negative Change of Encoder	AF3	movment of encoder	60= 60-1	59	pass

5	Open Loop Test of Movement going forward	AF3	10% duty cycles	Both wheels spin forward	Both wheels spin forward	pass
6	Open Loop Test of Movement for turning	AF3	10% duty cycles with one pin on to specify backward motion	Wheels spining in different directions	Wheels spining in different directions	pass
7	Ultra Sonic distance test at 1.5m	AF10	a wall 1.5m in front of robot	Nothing blocking	Nothing blocking	pass
8	Ultra Sonic distance test at 0.35m	AF10	a book blocking the path	Blocked	Blocked	pass
9	Speed is slewed on start	AF3	A 30% in duty cycles	Printing showing a duty cycle increase	Printing showing a duty cycle increase to 30%	pass
10	Closed Loop Test of Movement going forward	AF3	reference of 20 encoder counts/sec	Both wheels spin forward at 20 encoder counts/s	Both wheels spin forward at 20 encoder counts/s	pass
11	Closed Loop Test of Movement going forward	AF3	reference of 10 encoder counts/sec	Both wheels spin forward at 10 encoder counts/s	Left side failed to correctly start	fail - the software worked correctly but there was mechanical issues at slow speeds
12	Closed Loop Test of Movement going forward with unequal resistance	AF3	reference of 20 encoder counts/sec	Both wheels spin forward at 20 encoder counts/s	Both wheels spin forward at 20 encoder counts/s	pass

13	Closed Loop Test of Movement for turning	AF3	reference of 20 encoder counts/sec	Wheels spinning in different directions at 20 encoder counts/s	Wheels spinning in different directions at 20 encoder counts/s	pass
14	Ultra Sonic Stop	AF3	reference of 20 encoder counts/sec then something blocks its path	Something is blocking the path while moving so Alfred stops	Alfred Stopped	pass
15	Turn of 90deg	AF3	A reference of 90 degrees	Alfred will turn 90deg and stop	Alfred will turn 90deg and stop	Pass

5.6 Alfred Image Processing System

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
9	Able to take picture	AF3	software trigger to take the picture	A picture	a Picture	pass
10	Able to detect circles from an image	AF3	Image with a circle	Position of the circle	Position of the circle	pass
17	Image processing stop with node close	AF3	an image close up to camera while moving	Has reached the next node so Alfred stops motion	Alfred Stopped	pass
18	Image processing stop with node at ceiling away	AF3	An image further away at ceiling height distance	Has reached the next node so Alfred stops motion	Alfred Stopped	Conditional pass-needed lighting
19	Image processing False Positive 1	AF3	Camera staring at a brick wall with no circles	Alfred should continue motion	Alfred Continued motion	Pass

20	Image processing False Positive 2	AF3	Camera staring at the track within MARC	Alfred should continue motion	Alfred Stopped	Conditional Pass- Alfred noticed the contours within the track. This is deemed as an out of scope problem

5.7 Alfred Pumping System

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
1	Testing Serial communication to Arduino by sending d1d2x sent from Computer	AF1	d1d2x via serial signal	reading d1d2x read by USB	reading d1d2x read by USB	pass
2	Testing Serial communication from RPI by sending d1d2x sent from RPI	AF1	d1d2x via serial signal	reading d1d2x read by USB	reading d1d2x read by USB	pass
3	Testing Serial communication from RPI by receiving 11110000x sent from Arduino	AF1	11110000x via serial signal	reading 11110000x read by RPI USB	reading 11110000x read by RPI USB	pass
4	Testing receiving a value of 20C water with the temperature sensor	AF1	Temperature sensor in 20C water	20	19.8	pass
5	Testing receiving a value of 1Kg with the load sensor	AF1	1kg on load cell	1kg	0kg	Fail
6	Able to receive from the correct tank based on drink order (Tank 1)	AF2	Order for tank 1	Tank 1 pours	Tank 1 pours	pass
7	Able to receive from the correct tank based on drink order (Tank 2)	AF2	Order for tank 2	Tank 2 pours	Tank 2 pours	pass
8	Able to receive from the correct tank based on drink order (Tank 3)	AF2	Order for tank 3	Tank 3 pours	Tank 3 pours	pass
9	Under Weight error when tank is empty	AF2	No weight	Under Weight Error	Nothing	Fail need amplifier for weight sensor
10	Over Temperature error when tank is empty	AF2	a hot liquid	Over Temperature Error	Over Temperature Error	Pass
11	Not over Temperature error when tank is empty	AF2	a cold liquid	No Over Temperature Error	Nothing	Pass

5.8 Integration Testing

Test Number	Description	Requirement Reference	Inputs	Expected Outputs	Actual Outputs	Results
1	Alfred is able to pull from server and receive the next order	AF3	An order from the client application	Json indicating the order	Json indicating the order	pass
2	Alfred is able to pull from server and receive a modified map	AF3	A change from the administrative system for a change within the map of the restaurant	A text based version of the map	A text based version of the map	pass
3	Alfred is able to display text in regards to the directions that must be travelled to get to table 1	AF3	The map/ order from previous steps	Turn left-> Forward-> Turn Left -> Forward -> Turn Right -> Forward -> Turn Right -> Forward -> Forward	Turn left-> Forward-> Turn Left -> Forward -> Turn Right -> Forward -> Turn Right -> Forward -> Forward	pass
4	Alfred is able to build a message and send an order to the pumping system	AF1	The order from the previous test	d1d1d1 d2d3d3x	d1d1d1 d2d3d3x	pass
5	The pumping system is able to parse this message and poor to a cup based on the corresponding order	AF1	The message from the previous test	Dispenses 3 drinks from tank 1, 1 from tank 2 and 2 from tank 3 and replies with 0 for errors and ! for done	Dispenses 3 drinks from tank 1, 1 from tank 2 and 2 from tank 3 and replies with 0 for errors and ! for done	pass

6 Beyond Testing

6.1 Code Walk-through and Code Reviews

Code reviews were performed with at least two of the software team that had not wrote the code in particular. When a new piece of code would be pushed. A review request would be made and the person who wrote the code would help explain the functionality of the software to the other group members. From there, group members would be advised to ask questions about the software and would suggest changes if needed.

7 Supporting Material

Please refer to previous design documents.

8 Traceability Matrices

The following is the Traceability Matrix for the functional and non-functional requirements according to their test case classification:

Test Cases:	**Functional and Non-Functional Requirement:**
Alfred Client Application	Table Ordering Functional Requirement 1
	Non-Functional Requirement 5
	Non-Functional Requirement 8
	Non-Functional Requirement 15
	Non-Functional Requirement 21
	Non-Functional Requirement 30
	Non-Functional Requirement 32
	Non-Functional Requirement 35

Table 10: Alfred Client Application Traceability

Test Cases:	**Functional and Non-Functional Requirement:**
Server System	Alfred Functional Requirement 3
	Alfred Functional Requirement 4
	Alfred Functional Requirement 14
	Table Ordering Functional Requirement 1
	Table Ordering Functional Requirement 2
	Non Functional Requirement 33

Table 11: Server System Traceability

Test Cases:	**Functional and Non-Functional Requirement:**
Alfred Manager System	Alfred Functional Requirement 1
	Alfred Functional Requirement 3
	Alfred Functional Requirement 10

Table 12: Alfred Manager System Traceability

Test Cases:	**Functional and Non-Functional Requirement:**
Alfred Drive Train System	Alfred Functional Requirement 3
	Alfred Functional Requirement 10

Table 13: Alfred Drive Train System Traceability

Test Cases:	**Functional and Non-Functional Requirement:**
Alfred Image Processing System	Alfred Functional Requirement 3

Table 14: Alfred Image Processing System Traceability

Test Cases:	**Functional and Non-Functional Requirement:**
Alfred Pumping System	Alfred Functional Requirement 1

Table 15: Alfred Pumping System Traceability