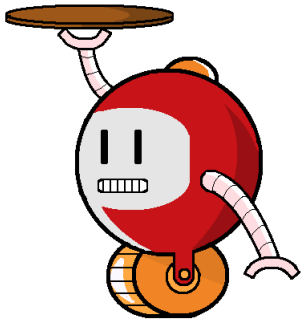

LazyBots

McMaster University

Design Document

SE 4GA6 & TRON 4TB6

Group 9

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

Table of Contents

List of Tables

List of Figures

1 Revisions

Date	Revision Number	Authors	Comments
December 24th, 2017	Revision 0	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	-
March 12th, 2018	Revision 1	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	Added the section to include the assumptions of the project as well as added some traceability between the components and the requirements. Finally, included the components that were likely to change.
March 16th, 2018	Revision 2	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	Combined System Design Document and Component Design Document

Table 1: LazyBots Table of Revisions

2 Purpose

The purpose of this project will be to create an autonomous robot that will navigate to and serve the requested drink to the user who request a drink. Currently in an office setting, workers must leave their offices to get their own drinks. In restaurants, drinks are served by waiters and waitresses, which hinders them from doing other work at that time. Alfred will be designed to make the serving drinks autonomous.

Alfred will allow users to request drinks. These requests will form a queue which Alfred will serve in order using a FIFO protocol. Alfred will go to the table of each user and pour the drinks ordered from that table. Alfred will also have an administrator user which will be able to call Alfred back and override any action that is being taken at the time.

The following document will outline the overall system components, as well as the overall system behaviour, operation and undesired error handling.

2.1 Scope

The system implemented is one that is meant to automate the dispensing of beverages to customers within a restaurant at the respective customers' table. The customer will be able to order a drink from their table which will be relayed to Alfred, where he will arrive at their table and dispense the requested drinks. The staff will be able to request Alfred to come back for charging and refilling when desired.

2.2 Context Diagram

Figure 1 is a context diagram of the drink serving robot, Alfred.

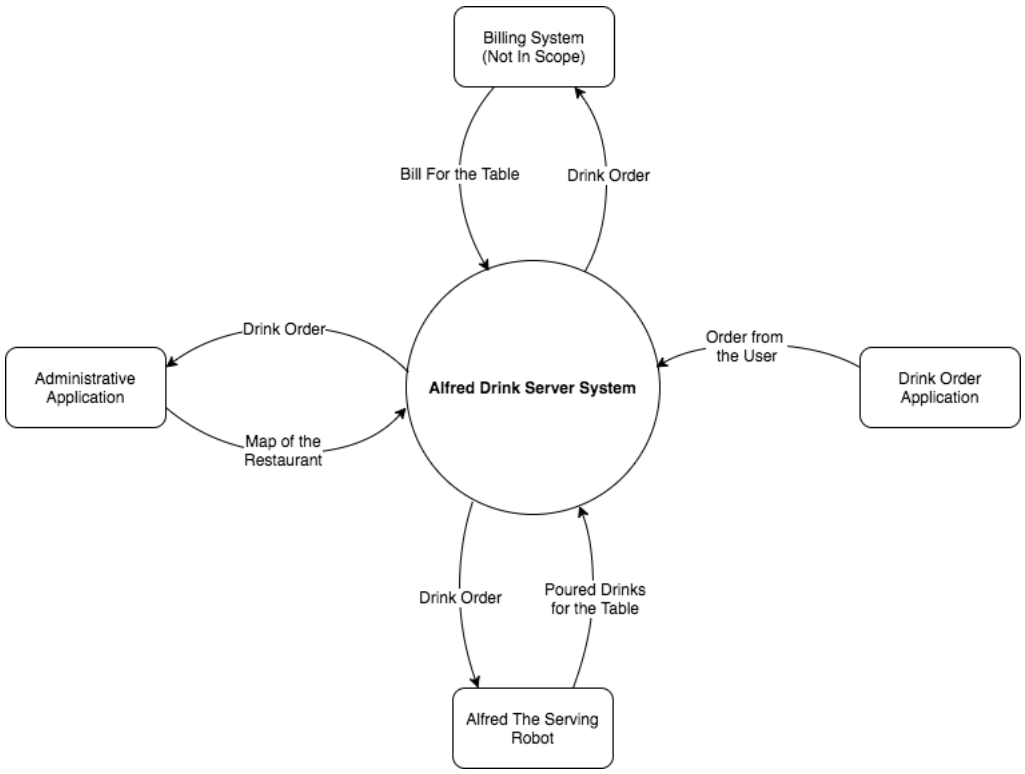

Figure 1: Drink Serving Robot Context Diagram

2.3 Diagram of Components

Figure 2 is a diagram that shows the interaction of components of the drink serving robot system.

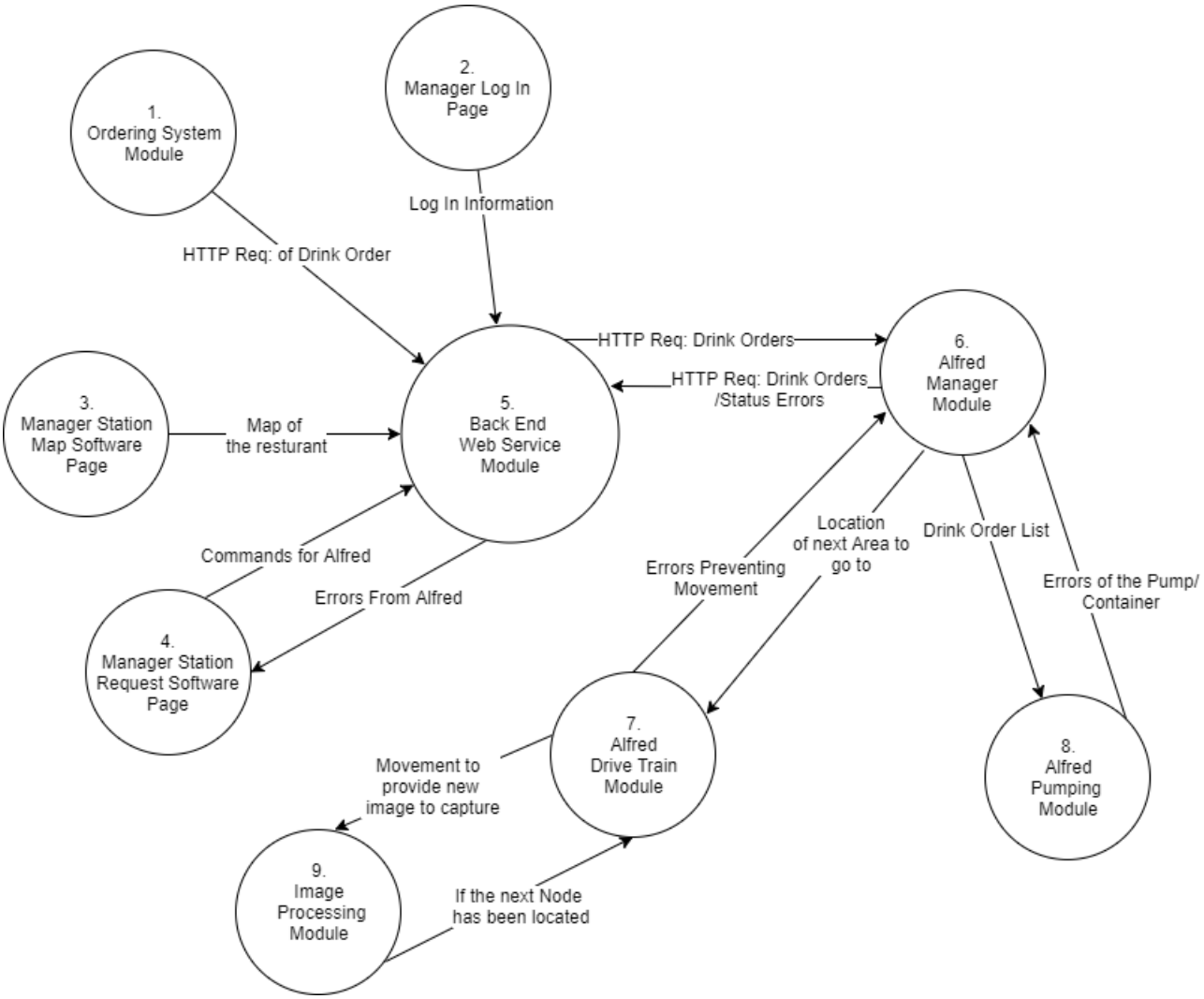

Figure 2: Drink Serving Robot System Component Diagram

2.4 Assumptions

Alfred's assumption are represented in the tables below.

A1	The environment will only be comprised of a one story building with no steps.
Rationale	Different environment elevations are beyond the scope of the project. Alfred will not be able to navigate around steps.

A2	The ground within the establishment will be smooth without defect and with moderate friction.

Rationale	Imperfect floors with too much or too little friction are beyond the scope of the project. Alfred will not be able to navigate properly with imperfect floors and the wheels will slip with too little friction.

A3	The establishment will be well light with a drop ceiling of a maximum height of 9 feet.
Rationale	Poor lighting and different ceiling heights are beyond the scope of the project. A level ceiling is needed for node placement as well as proper lighting will be needed so that Alfred will be able to read the nodes.

A4	The establishment will be able to support a strong and stable Wi-Fi connection (internet connection).
Rationale	Poor internet connections are beyond the scope of the project. To support communication between all components a strong and stable internet connection is needed.

A5	The width of the walkways will be wide enough to accommodate all people.
Rationale	If a table is not accessible to a human, it will not be accessible for Alfred.

A6	Orders will be placed via an Android or iOS application.
Rationale	Eliminates the need for human interactions, making Alfred completely autonomous.

A7	The height of a table will not exceed 30".
Rationale	This will help simplify the scope of the project and reduce the customers' discomfort when reaching for a drink.

A8	The serving size of a medium sized cup will not vary largely in terms of ounces.
Rationale	The standard ounces in a cup will be restricted to 12oz. to accommodate as many users as possible and to limit the scope.

2.5 Components to Requirements

3 System Variables

3.1 Monitored and Controlled Variables

The following is a list of variables that will be monitored.

Monitor Name	Monitor Type	Range	Units	Comment(s)
$w_{wheel_{left}}$	Speed	[0, 100]	rad/s	Left Wheel Speed
$w_{wheel_{right}}$	Speed	[0, 100]	rad/s	Right Wheel Speed
$m_{container}$	Mass	[0, 1.0]	Kg	Weight of the storage device
m_{drink}	Mass	[0, 1.0]	Kg	Weight of the drink
$b_{cuptaken}$	Boolean	[0,1]	N/A	If the cup has been taken
$d_{objects}$	Distance[]	[0, 10.0]	m	Set of distances to closest obstacle
V_{batt}	Voltage	[0, 20.0]	m	Voltage levels of batteries

The following is a list of variables that will be controlled.

Controlled Name	Controlled Type	Range	Units	Comment(s)
w_{motor}	Speed	[0, 100]	rad/s	Motor Speed
$percent_{dutycycle_{left}}$	Percent	[0,1.0]	%	The duty cycle of the left side of the drive-train
$percent_{dutycycle_{right}}$	Percent	[0, 1.0]	%	The duty cycle of the right side of the drive-train
V_{pump}	Voltage	[0, 5.0]	m	Voltage going to the liquid pump
$errors_{drivetrain}$	Unsigned Byte	[0, 2^8]	N/A	Errors from drive-train
$errors_{pump}$	Unsigned Byte	[0, 2^8]	N/A	Errors from pumping system module
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred
$LED_{drinksignal}$	Boolean	[0, 1]	N/A	Signal of drink that it is ready to be picked up
Q_{pump}	Flow Rate	[0, 100]	(m^3/s)	Flow rate of the pump

3.2 Constants

The following is a list of system constants.

Constant Name	Constant Type	Value	Units	Comment(s)
$V_{battMin}$	Voltage	9.0	m	The minimum voltage necessary for drive-train movement
$m_{DrinkMin}$	Mass	0.35	Kg	The minimum weight of the drink to be considered as ready for the customer.
$t_{timeout}$	Time	30	s	The maximum time Alfred or the ordering application will wait for a message from the server before timing out.
$t_{pumptimeout}$	Time	5	s	The maximum time Alfred will try to pump without noticing a change in the weight of the tank.
$t_{maxpump}$	Time	10	s	The maximum time Alfred will try to dispense a drink within.
Q_{pump}	Flow Rate	TBD	(m^3/s)	Flow rate of the pump
$freq_{bauderate}$	Frequency	9600	(Herts)	The rate at which UART communication will be performed at.
$obstruction_{timeout}$	Time	10	s	The amount of time that Alfred will stop movement due to an object in its path before ruling that there is an obstruction and throwing an error.

4 Behaviour Overview

1. **Ordering System Module**: Provided the order information, this module will communicate with the server to "order a drink".
2. **Manager Login Page**: Given the login credentials, will authenticate administrator of the system with the server.
3. **Manager Station Map Software Page**: Will allow administrator to create or modify the map of the area where Alfred will deliver drinks. This map will then be sent and stored on the server.
4. **Manager Station Request Software Page**: Will allow administrator to execute commands for Alfred, as well as view incoming error codes from Alfred. Said commands, will be sent to the server, which will then communicate with Alfred.

5. **Back End Web Service Module**: Will route communication to different components of the system. Will also be responsible for authentication and management of queue.
6. **Alfred Manager Module**: Endpoint for communication with Alfred. Will manage communication with server, as well as send any errors that Alfred is experiencing.
7. **Alfred Drive Train Module**: Responsible for driving and managing the motors based on desired route. Will also be sending errors preventing movement to Alfred Manager Module.
8. **Alfred Pumping Module**: Will control pumping system in regards of when to pour, how long, rate of dispensing, etc. Will also be sending errors pertaining to the pump or container to Alfred Manager Module.
9. **Image Processing Module**: Will detect any obstacles in the way as well as locate incoming nodes. Will communicate with Alfred Drive Train Module, to determine wether any required action based on results.

5 Component Traceability

Component Module:	**Functional and Non-Functional Requirement:**
Ordering System Module	Table Ordering Functional Requirement 1
	Table Ordering Functional Requirement 2
	Non Functional Requirement 4
	Non Functional Requirement 5
	Non Functional Requirement 8
	Non Functional Requirement 9
	Non Functional Requirement 15
	Non Functional Requirement 30
	Non Functional Requirement 32

Table 13: Ordering System Traceability

Component Module:	**Functional and Non-Functional Requirement:**
Manager Login Page Module	Non Functional Requirement 4
	Non Functional Requirement 5
	Non Functional Requirement 8
	Non Functional Requirement 9
	Non Functional Requirement 30

Table 14: Manager Login Page Traceability

Component Module:	Functional and Non-Functional Requirement:
Manager Station Map Software Module	Administrator Functional Requirement 1
	Administrator Functional Requirement 2
	Non Functional Requirement 4
	Non Functional Requirement 5
	Non Functional Requirement 8
	Non Functional Requirement 9
	Non Functional Requirement 29
	Non Functional Requirement 30
	Non Functional Requirement 33

Table 15: Manager Station Map Traceability

Component Module:	Functional and Non-Functional Requirement:
Manager Station Request Software Module	Administrator Functional Requirement 3
	Administrator Functional Requirement 4
	Administrator Functional Requirement 5
	Non Functional Requirement 4
	Non Functional Requirement 5
	Non Functional Requirement 8
	Non Functional Requirement 9
	Non Functional Requirement 29
	Non Functional Requirement 30
	Non Functional Requirement 31
	Non Functional Requirement 33

Table 16: Manager Station Request Traceability

Component Module:	Functional and Non-Functional Requirement:
Back End Web Service Module	Alfred Functional Requirement 1
	Alfred Functional Requirement 2
	Alfred Functional Requirement 4
	Alfred Functional Requirement 6
	Alfred Functional Requirement 7
	Alfred Functional Requirement 8
	Alfred Functional Requirement 9
	Alfred Functional Requirement 10
	Table Ordering Functional Requirement 1
	Table Ordering Functional Requirement 2
	Non Functional Requirement 26
	Non Functional Requirement 27
	Non Functional Requirement 31
	Non Functional Requirement 32
	Non Functional Requirement 33

Table 17: Back End Web Service Traceability

Component Module:	Functional and Non-Functional Requirement:
Alfred Manager Module	Alfred Functional Requirement 3
	Alfred Functional Requirement 7
	Alfred Functional Requirement 8
	Alfred Functional Requirement 9
	Alfred Functional Requirement 10
	Alfred Functional Requirement 11
	Alfred Functional Requirement 12
	Alfred Functional Requirement 13
	Alfred Functional Requirement 14
	Non Functional Requirement 11
	Non Functional Requirement 16
	Non Functional Requirement 17
	Non Functional Requirement 19
	Non Functional Requirement 20
	Non Functional Requirement 21
	Non Functional Requirement 26
	Non Functional Requirement 27
	Non Functional Requirement 31

Table 18: Alfred Manager Traceability

Component Module:	Functional and Non-Functional Requirement:
Alfred Drive-Train Module	Alfred Functional Requirement 3
	Alfred Functional Requirement 12
	Non Functional Requirement 1
	Non Functional Requirement 2
	Non Functional Requirement 11
	Non Functional Requirement 13
	Non Functional Requirement 16
	Non Functional Requirement 18
	Non Functional Requirement 20
	Non Functional Requirement 38

Table 19: Alfred Drive-Train Traceability

Component Module:	Functional and Non-Functional Requirement:
Alfred Pumping System Module	Alfred Functional Requirement 1
	Alfred Functional Requirement 4
	Alfred Functional Requirement 5
	Alfred Functional Requirement 6
	Alfred Functional Requirement 7
	Alfred Functional Requirement 8
	Alfred Functional Requirement 9
	Alfred Functional Requirement 11
	Alfred Functional Requirement 13
	Non Functional Requirement 1
	Non Functional Requirement 2
	Non Functional Requirement 10
	Non Functional Requirement 12
	Non Functional Requirement 17
	Non Functional Requirement 19

Table 20: Alfred Pumping System Traceability

Component Module:	Functional and Non-Functional Requirement:
Alfred Image Processing Module	Alfred Functional Requirement 10
	Non Functional Requirement 16
	Non Functional Requirement 20

Table 21: Alfred Image Processing Traceability

6 Component Overview

6.1 Ordering System Module

6.1.1 Description

This module will be used for user input by taking the orders of the client. This user input will be taken in by the mobile application based on different button inputs/ radio button selections. These orders are then

packaged by the module to be sent to the server based on an HTTP request.

6.1.2 Inputs and Outputs

Inputs: User input defining:

Input Name	Input Type	Range	Units	Comment(s)
$Order_{Num}$	Unsigned Integer User Input	[0,5]	count	Number of Orders
$Order_{Drinks}$	User Input	N/A	N/A	List of Ordered Drinks

Outputs: Packaged Information within HTTP Request for:

Output Name	Output Type	Range	Units	Comment(s)
$Order_{Num}$	Unsigned Integer User Input	[0,5]	count	Number of Orders
$Order_{Drinks}$	User Input	N/A	N/A	List of Ordered Drinks
$Order_{TableNum}$	Unsigned Integer User Input	$[0,2^{16}]$	N/A	Table Number
$Order_{Rid}$	Unsigned Integer User Input	$[0,2^{16}]$	N/A	Identification of the restaurant

6.1.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
$Order_{Num}$	Unsigned Integer User Input	$Order_{Num}$ of type string	Input Regulation
$Order_{Drinks}$	User Input	$Order_{Drinks}$ not of type string	Input Regulation

6.1.4 Timing Constraints

Timing Constraints are based on the server sending a success signal within $t_{timeout}$ seconds.

6.1.5 Initialization

At startup/new order of the application, it will start a blank order page where the user will be able to add drink orders and use radio buttons to select the desired drink.

6.2 Manager Login Page Module

6.2.1 Description

This module is a web based application for the managers to be able to log into the management systems.

6.2.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$UserName$	String User Input	10 chars	N/A	N/A
$Password$	String User Input	20 chars	N/A	N/A
$GotoMapPage$	Button User Input	[0,1]	N/A	N/A
$GotoAlfredInfo$	Button User Input	[0,1]	N/A	N/A

Outputs: To be displayed to the user.

Output Name	Output Type	Range	Units	Comment(s)
FailureMessage	String	10 chars	N/A	Failure message if there is an incorrect information
GotoPage	Action	N/A	N/A	navigating to the correct page if it was a success.

6.2.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
UserName	String User Input	N/A	N/A
Password	String User Input	N/A	N/A
GotoMapPage	Button User Input	N/A	N/A
GotoAlfredInfo	Button User Input	N/A	N/A

6.2.4 Timing Constraints

Given optimal networking conditions, the server must respond with $t_{timeout}$ seconds.

6.2.5 Initialization

This will default to a HTML page with no information.

6.3 Manager Station Map Software Module

6.3.1 Description

This module will be used for the user input to create a map where the manager of the restaurant will be able to define the details about the restaurant that will help Alfred with it's navigation to different tables.

6.3.2 Inputs and Outputs

Inputs: User input defining:

Input Name	Input Type	Range	Units	Comment(s)
$Resturant_{Width}$	Integer User input	[0,20]	m	N/A
$Resturant_{Length}$	Integer User input	[0,20]	m	N/A
$Resturant_{CanTravel}[][]$	Button[][] User input	N/A	N/A	A set of areas in which Alfred can travel to
$Resturant_{Tables}[][]$	Button[][] User input	N/A	N/A	A set of areas in which contains tables
$Resturant_{CannotTravel}[][]$	Button[][] User input	N/A	N/A	A set of areas in which Alfred cannot travel to
$Resturant_{Home}$	Button User input	N/A	N/A	An area that defines Alfred's home location

Outputs: A text file that contains the following information.

Input Name	Input Type	Range	Units	Comment(s)
$Resturant_{Width}$	Integer	[0,20]	m	N/A

$Resturant_{Length}$	Integer	[0,20]	m	N/A
$Resturant_{D}ata$	Char	[H,X,0,T]	N/A	Table Information

6.3.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
$Resturant_{Width}$	Integer User input	$Resturant_{Width}$ of type string	Input Regulation
$Resturant_{Length}$	Integer User input	$Resturant_{Length}$ of type string	Input Regulation
$Resturant_{CanTravel}[][]$	Button[][] User input	N/A	N/A
$Resturant_{Tables}[][]$	Button[][] User input	N/A	N/A
$Resturant_{CannotTravel}[][]$	Button[][] User input	N/A	N/A
$Resturant_{Home}$	Button User input	N/A	N/A

6.3.4 Timing Constraints

Timing Constraints are based on the server sending a success signal within $t_{timeout}$ seconds.

6.3.5 Initialization

At startup/new order of the application, it will load in the users map that is associated with their profile. If this is the first time using the application then it will default to a 1x1 map.

6.4 Manager Station Request Software Module

6.4.1 Description

This module is used for the manager to determine Alfred's state from an office as well as being able to override Alfred's system to come back to the home base.

6.4.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
req_{Home}	Boolean User input	[0,1]	N/A	Calling back Alfred
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred

Outputs: To be displayed to the user.

Output Name	Output Type	Range	Units	Comment(s)
$warn_{LowLiquid}$	Boolean	[0, 1]	N/A	Low liquid levels
$error_{LiquidLeak}$	Boolean	[0, 1]	N/A	Leaking of liquid error
$error_{NotPumping}$	Boolean	[0, 1]	N/A	Not pumping error
$error_{NoMovement}$	Boolean	[0, 1]	N/A	Not able to move error
$error_{LowBatt}$	Boolean	[0, 1]	N/A	Low battery Error

6.4.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
req_{Home}	Boolean User input	N/A	N/A
$errors_{Alfred}$	Unsigned Byte	N/A	N/A

6.4.4 Timing Constraints

One timing constraints are based on the server sending a success signal within $t_{timeout}$ seconds. Another constraint is that Alfred will return within the time of w_{wheel}*distance.

6.4.5 Initialization

At startup this interface should pull the last status of the robot and display it to the user. If no previous data is found then it will return that there are currently no errors.

6.5 Back End Web Service Module

6.5.1 Description

This module is a server component that will hold order information for different tables. This will authorize different users to be able to accept different drink requests from the ordering system. It will return the next drink order for the restaurants Alfred robot.

6.5.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$HttpReq_{Orders}$	HTTP Request	N/A	N/A	HTTP request package with the information for drink orders.
$Map_{textfile}$	Text File	N/A	N/A	A text file with a map of the restaurant
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred
$UserName$	String	10 chars	N/A	N/A
$Password$	HashCode	N/A	N/A	Success result for authenticated results.

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$HttpReq_{Orders}$	HTTP Request	N/A	N/A	HTTP request package with the information for drink orders.
$Map_{textfile}$	Text File	N/A	N/A	A text file with a map of the restaurant
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred
$Login_{Status}$	String	10 chars	N/A	Login Success or Fail
$Http_{Result}$	String	10 chars	N/A	N/A

6.5.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
$HttpReq_{Orders}$	HTTP Request	$HttpReq_{Orders}$ formatting error	Server Validation

$Map_{textfile}$	Text File	N/A	N/A
$errors_{Alfred}$	Unsigned Byte	N/A	N/A
$UserName$	String	N/A	N/A
$Password$	HashCode	N/A	N/A

6.5.4 Timing Constraints

Given optimal networking conditions, the server must respond with $t_{timeout}$ seconds.

6.5.5 Initialization

This server will be initialized by a database with one restaurant which will be used for the purposes of testing the system.

6.6 Alfred Manager Module

6.6.1 Description

This module acts as a manager to the different components of Alfred. It will run on a raspberry pi to command the drive-train to move to different nodes of the map. The manager will also communicate through serial communication using a USB when ordering the next drink to the pumping system.

6.6.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
Map	Graph	N/A	N/A	Graph of the text file with a map of the restaurant
$Orders_{drinks}$	Order[]	N/A	N/A	A list of drink orders ordered by table.
$b_{requestHome}$	Boolean	[0, 1]	N/A	A request to come back to the base
$errors_{drivetrain}$	Unsigned Byte	[0, 2^8]	N/A	Errors from drive-train
$errors_{pump}$	Unsigned Byte	[0, 2^8]	N/A	Errors from pumping system module

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$NextOrder$	Order	N/A	N/A	Next Drink Order
$NextNode$	Node	[(0,0), (length(Map), width(Map)]	N/A	The next node to travel to
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred

6.6.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
Map	Graph	N/A	N/A
$Orders_{drinks}$	Order[]	N/A	N/A
$b_{requestHome}$	Boolean	N/A	N/A
$errors_{drivetrain}$	Unsigned Byte	N/A	N/A
$errors_{pump}$	Unsigned Byte	N/A	N/A

6.6.4 Timing Constraints

Given optimal networking conditions, the system must receive a response within $t_{timeout}$ seconds. The communication with the pumping system must be done at the specified $freq_{bauderate}$.

6.6.5 Initialization

This will assume that there is no requests at startup. It will request from the server asking for the next drink order. If there is no map previously defined within a text file then it will request this as well.

6.6.6 Raspberry PI Specifications

Manufacturer: Raspberry PI
Processor: Broadcom BCM2387 chipset.
Memory: 1GB
Power:Micro USB socket 5V1, 2.5A
GPIO: 17 pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector: 15-pin MIPI Camera Serial Interface (CSI-2)
Memory Card Slot: Push/pull Micro SDIO

6.7 Alfred Drive-Train Module

6.7.1 Description

This module will provide power to Alfred's drive-train. It will use the feedback of the left and right encoders and take the error to perform PI control on them. This PI control output will then be translated into a duty cycle for each side to be able to power the DC motors with pulse width modulation. This module will communicate to the image recognition software to receive the position of the next marker and use this information of where it currently is on its path. This module will use ultrasonic sensors to get a set of distances (d_{object}) to the nearest object to determine if it is safe to continue moving.

6.7.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
Map	Graph	N/A	N/A	Graph of the text file with a map of the restaurant
$NextNode$	Node	[(0,0), (length(Map), width(Map)]	N/A	The next node to travel to
$w_{wheel_{left}}$	Float	[0, 100]	rad/s	Left Encoder Input
$w_{wheel_{right}}$	Float	[0, 100]	rad/s	Right Encoder Input
$d_{objects}$	Float[]	[0, 10.0]	m	N/A
$Marker_{PosX}$	Unsigned Integer	[0,2^{16}]	Pixels	N/A
$Marker_{PosY}$	Unsigned Integer	[0,2^{16}]	Pixels	N/A
$b_{NextMarkerFound}$	Boolean	[0,1]	N/A	N/A
V_{batt}	Float	[0, 20.0]	m	N/A

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$percent_{dutycycle_{left}}$	Float	[0,1.0]	%	N/A
$percent_{dutycycle_{right}}$	Float	[0, 1.0]	%	N/A
$errors_{drivetrain}$	Unsigned Byte	[0, 2^8]	N/A	N/A

6.7.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
Map	Graph	Map formatting error	Map set from User Interface which constricts format of file.
$NextNode$	Node	N/A	N/A
$w_{wheel_{left}}$	Float	N/A	N/A
$w_{wheel_{right}}$	Float	N/A	N/A
$d_{objects}$	Float[]	N/A	N/A
$Marker_{PosX}$	Unsigned Integer	N/A	N/A
$Marker_{PosY}$	Unsigned Integer	N/A	N/A
$b_{NextMarkerFound}$	Boolean	N/A	N/A
V_{batt}	Float	N/A	N/A

6.7.4 Timing Constraints

This module will have to deliver speed requirements that of walking speed so that it will be able to serve tables at a timely manner.

6.7.5 Initialization

This module will not be started until manager starts the process and initializes the proper information.

6.7.6 Diagram of Simulink Control System

Figure 3 is a diagram that shows the top level of the dc motor control system.

6.7.7 DC Motors Circuit Diagram

Figure 4 is a circuit diagram for showing the DC motor drive-train.

6.7.8 DC Motor Specifications

Manufactured Part Number: 393111-01
Rated Voltage: 18V
Current: 3A-5A

6.7.9 H-bridge Specifications

Manufactured Part Number: IRF3205
Rated Voltage: 3 36V

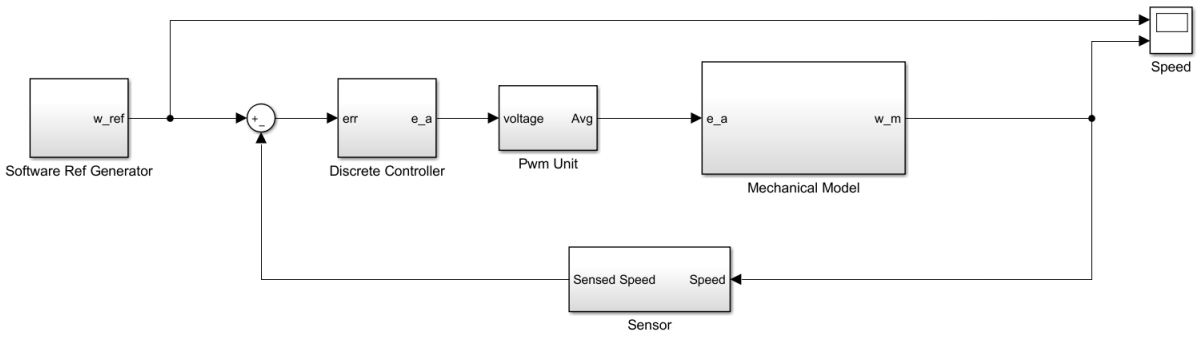

Figure 3: Top level of the dc motor control system

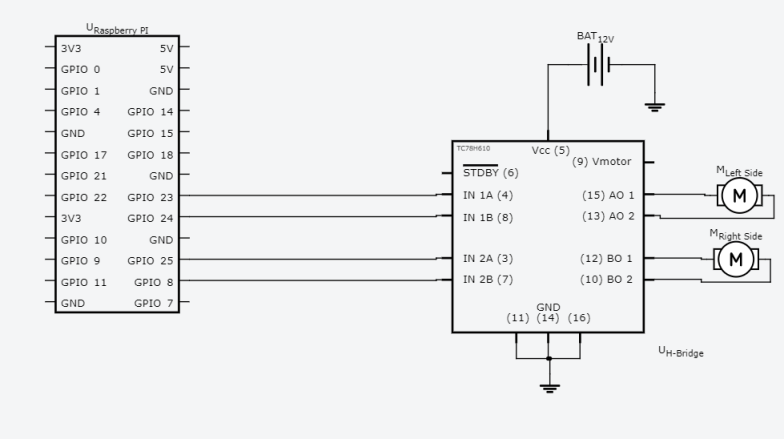

Figure 4: H-Bridge Circuit Diagram

Rated Current: 10A continuous, Peak 30A

6.7.10 Encoder Specifications

Shaft Diameter 6mm
Working Range: 3-5V DC

6.8 Alfred Pumping System Module

6.8.1 Description

This module consists of an Arduino Mega which will receive information from the Alfred manager module through UART to receive the next drink order. This will then begin to dispense the specific drink until it has reached $m_{containerMin}$. It will then show the customer that the drink has been completed by turning on: $LED_{drinksignal}$. It will then read $b_{cuptaken}$ from a light sensor to determine if the cup has been taken at which point it will then wait for the next drink cup to get in place and begin pouring. If it is not able

to pump liquid, or it is losing fluid when not dispensing, then it will send the appropriate errors through UART back to the manager.

6.8.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$b_{cuptaken}$	Boolean	[0,1]	N/A	N/A
$m_{container}$	float	[0, 1.0]	Kg	N/A
m_{drink}	float	[0, 1.0]	Kg	N/A
$Order_{drinks}$	Order[]	N/A	N/A	List of Drinks

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$LED_{drinksignal}$	Boolean	[0,1]	N/A	N/A
V_{pump}	float	[0, 5.0]	V	N/A
$errors_{pump}$	Unsigned Byte	[0, 2^8]	N/A	N/A

6.8.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
$b_{cuptaken}$	Boolean	N/A	N/A
$m_{container}$	float	N/A	N/A
m_{drink}	float	N/A	N/A
$Order_{drinks}$	Order[]	$Order_{drinks}$ formatting error	Server Validation

6.8.4 Timing Constraints

This module will have to dispense the drink within $t_{maxpump}$

6.8.5 Initialization

This module start by waiting for the manager to send the drink information for the system to pour. All errors within the system will start as false until they have been triggered.

6.8.6 Diagram of DC Pump Control System

Figure 5 is a diagram that shows the top level of the dc pump control system.

6.8.7 DC Pump Circuit Diagram

Figure 6 is a diagram that shows the DC Pump Circuit.

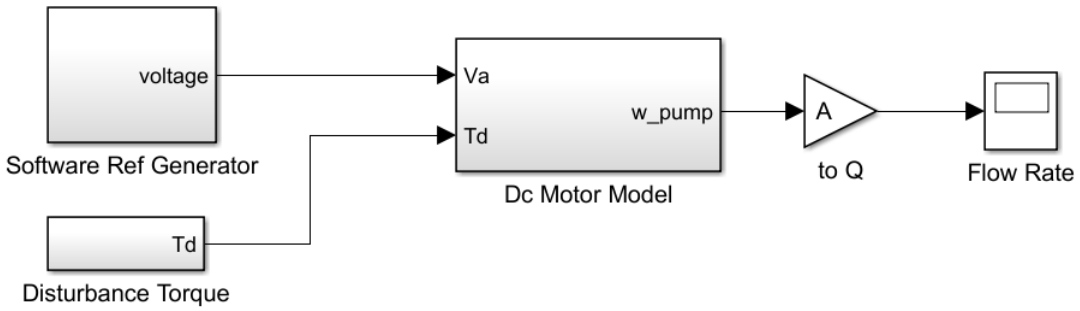

Figure 5: Top level of the dc motor control system

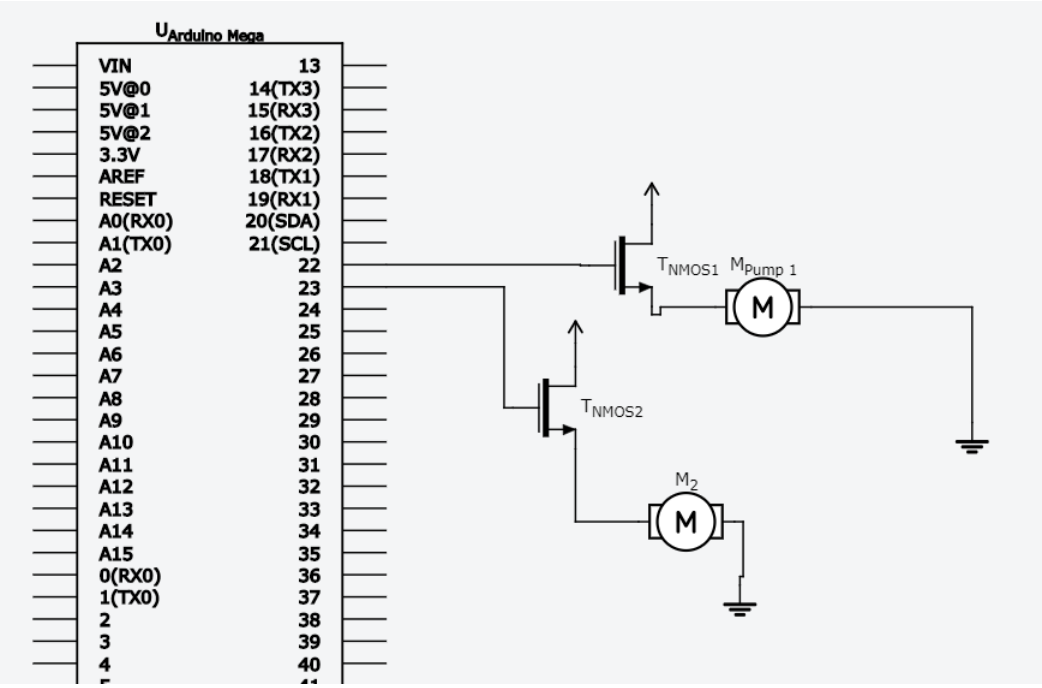

Figure 6: DC Pump Circuit Diagram

6.8.8 Liquid Temperature Circuit Diagram

Figure 7 is a circuit diagram for sensing the temperature of the liquid.

6.8.9 Weight Detection Circuit Diagram

Figure 8 is a circuit diagram for sensing the weight of the storage containers.

6.8.10 DC Pump Specifications

Manufacture : Yosoo
DC Voltage: 3-6V
Flow rate: 80-120L/H
Material: engineering plastic

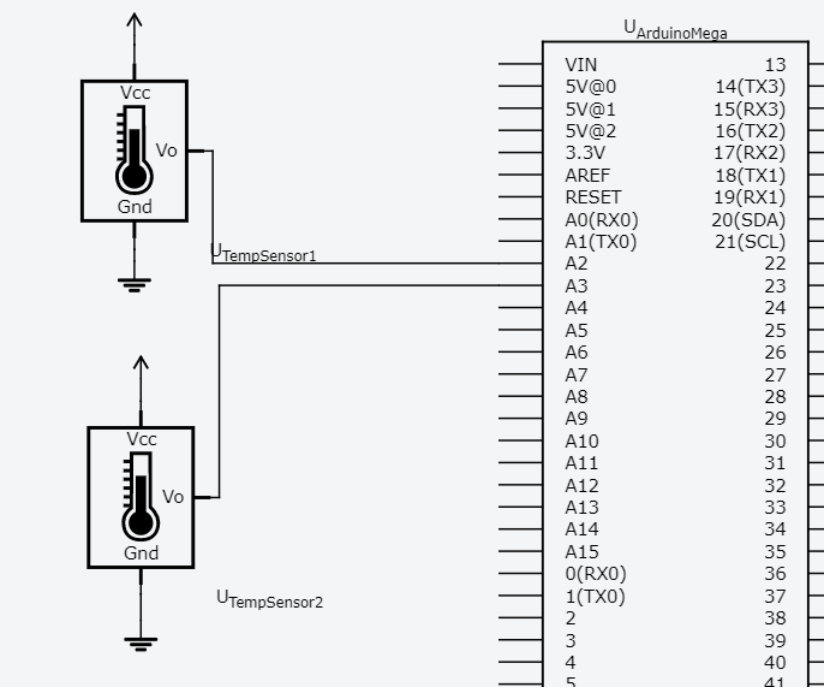

Figure 7: Top level of the dc motor control system

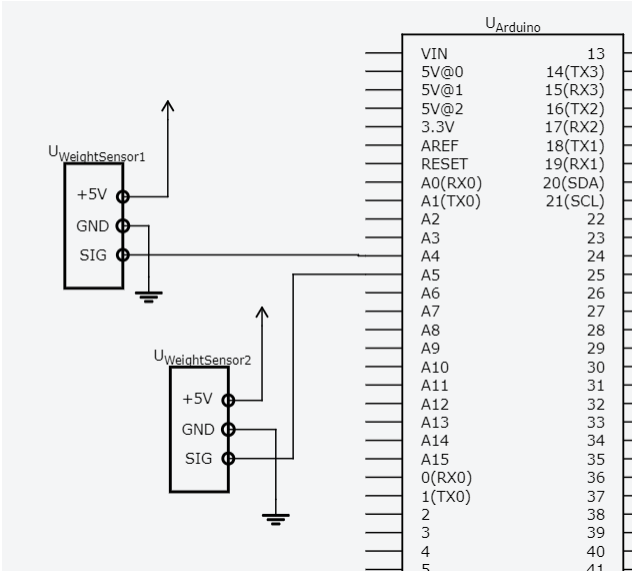

Figure 8: Weight Detection Circuit Diagram

Diameter: 24.5mm by 46mm
Outside diameter: 0.3"

6.8.11 Arduino Mega Specifications

Microcontroller: ATmega1280
Operating Voltage: 5V
Digital I/O Pins: 54 (of which 15 provide PWM output)

Analog Input Pins: 16
DC Current per I/O Pin: 40 mA
DC Current for 3.3V Pin: 50 mA
Flash Memory: 128 KB
Clock Speed: 16MHz

6.8.12 Container Specifications

Manufacture : Rubbermaid
Dimensions: 10 1/2" x 9" x 4" Capacity: 18 Cups/4.2 L

6.8.13 Tubing Specifications

Manufacture : Plumb Craft
Diameter: 1/4"ID Food Grade Tubing will be used.

6.9 Alfred Image Processing Module

6.9.1 Description

This module will receive information from a camera while the drive-train is in motion. This camera will be looking at the ceiling looking for positions of circles that will denote one meter distances set up within a grid around the ceiling of the restaurant. This module will determine if there is a new marker within the image and if so where is the X and Y position of that marker to provide to the drive-train.

6.9.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$Image_{input}$	Bitmap	N/A	N/A	Image of the ceiling

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$Marker_{PosX}$	Unsigned Integer	[0,2^{16}]	Pixels	N/A
$Marker_{PosY}$	Unsigned Integer	[0,2^{16}]	Pixels	N/A
$b_{NextMarkerFound}$	Boolean	[0,1]	N/A	N/A

6.9.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
$Image_{input}$	Bitmap	N/A	N/A

6.9.4 Timing Constraints

This information must process in time for the drive train to be able to navigate based off of it.

6.9.5 Initialization

This module start by the drive-train application.

6.9.6 Raspberry PI Camera Specifications

Manufacturer: Raspberry PI
Resolution: 1080p30, 720p60 and 640 ÃŮ 480p60/90
Field of View (FOV): 62.2 degrees by 48.8 degrees

7 Likelihood of Change

Module	Likelihood of Change	Rationale
Ordering System Module	Very Unlikely	Key implementation aspect
Manager Login Page Module	Very Unlikely	Key implementation aspect
Manager Station Map Software Module	Very Unlikely	Key implementation aspect
Manager Station Request Software Module	Very Unlikely	Key implementation aspect
Back End Web Service Module	Very Unlikely	Key implementation aspect
Alfred Manager Module	Very Unlikely	Key implementation aspect
Alfred Pumping System Module	Very Unlikely	Key implementation aspect
Alfred Image Processing Module	Very Unlikely	Key implementation aspect

8 Normal Operation

Alfred is a mostly autonomous robot, only requiring human intervention in the event of an error or warning. Alfred will be able to navigate the restaurant by itself, and will serve drinks to tables. Customers will be able to place orders via a mobile application, which will be sent to a server. Orders to serve will be sent to Alfred using a FIFO protocol. Once Alfred has finished with an order, it will be able to request for a new order to serve. Management will be able to recall Alfred to the kitchen at any point using the admin console. In the event of a recall, Alfred will finish the current job and will return to the kitchen afterwards. Management will also be able to create a map of the restaurant and upload it to Alfred, which will give Alfred the means to navigate the restaurant.

9 Undesired Event Handling

Alfred will be able to detect undesired behaviours and conditions such as low liquid levels below threshold $m_{containerMin}$, any issues with pumping liquid, low battery level, leaking liquids, and any blockages in the current path once Alfred has been blocked for a time greater than $obstruction_{timeout}$. In the event of any error condition, Alfred will send an error code to the kitchen, to alert the staff of the issue. Wherever possible, Alfred will return to the kitchen in an error condition to request a fix. Otherwise, if movement is not possible, the kitchen staff will have to pick Alfred up from the dining room. Along with alerting management of any current errors, Alfred will also be able to indicate whether it is returning to the kitchen or requires pickup.

10 References

Raspberry pi data sheet

[1]"Exploring Raspberry Pi", 2017. [Online].
Available: http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf.

[Accessed: 23- Dec- 2017].

Arduino Mega Information

[2] "Arduino Mega". [Online]. Available:
https://www.arduino.cc/en/Main/ArduinoBoardMega. [Accessed: 23- Dec- 2017].

DC Pump Information

[3] "3-6V Mini Micro Submersible Pumps DC Motor Pump Water Pump - 80-120L/H". [Online].
Available:
https://www.amazon.ca/gp/product/B01LWQCXEL/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1.
[Accessed: 23- Dec- 2017].

Pump Container Information

[4] "Rubbermaid 1856059 Modular Cereal Keeper". [Online].
Available:
https://www.amazon.ca/gp/product/B00BEUDXRW/ref=oh_aui_detailpage_o09_s00?ie=UTF8&psc=1.
[Accessed: 23- Dec- 2017].

Raspberry camera pi data sheet

[4] "CAMERA MODULE". [Online]. Available:
https://www.raspberrypi.org/documentation/hardware/camera/. [Accessed: 23- Dec- 2017].