



# LazyBots

MCMaster UNIVERSITY

Development Process and Implementation

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

<b>1</b>	<b>Revisions</b>	<b>3</b>
<b>2</b>	<b>Purpose</b>	<b>4</b>
2.1	Scope . . . . .	4
2.2	Context Diagram . . . . .	4
2.3	Diagram of Components . . . . .	4
<b>3</b>	<b>System Variables</b>	<b>4</b>
3.1	Monitored and Controlled Variables . . . . .	4
3.2	Constants . . . . .	5
<b>4</b>	<b>Behaviour Overview</b>	<b>6</b>
<b>5</b>	<b>Component Overview</b>	<b>7</b>
5.1	Ordering System Module . . . . .	7
5.2	Manager Login Page Module . . . . .	7
5.3	Manager Station Map Software Module . . . . .	8
5.4	Manager Station Request Software Module . . . . .	9
5.5	Back End Web Service Module . . . . .	9
5.6	Alfred Manager Module . . . . .	10
5.7	Alfred Drive-train Module . . . . .	10
5.8	Alfred Pumping System Module . . . . .	11
5.9	Alfred Image Processing Module Module . . . . .	12
<b>6</b>	<b>Normal Operation</b>	<b>12</b>
<b>7</b>	<b>Undesired Event Handling</b>	<b>13</b>

## List of Tables

1	VIC Table of Revisions . . . . .	3
---	----------------------------------	---

## List of Figures

1	Drink Serving Robot Context Diagram . . . . .	5
2	Drink Serving Robot System Component Diagram . . . . .	6

## 1 Revisions

Table 1: VIC Table of Revisions

Date	Revision Number	Authors	Comments
November 24 <sup>th</sup> , 2017	Revision 0	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	-

## 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 requests a drink. Currently in an office setting, workers must leave their offices to get their own drinks. Also, 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 respected customers' table. The customer will be able to order a drink from their table which will be followed by Alfred arriving at their table and dispensing the requested drinks. The staff will be able to request Alfred to come back for charging and refilling when desired.

### 2.2 Context Diagram

The following is a context diagram of the drink serving robot, Alfred.

### 2.3 Diagram of Components

The following is a diagram that shows the interaction of components of the drink serving robot system.

## 3 System Variables

### 3.1 Monitored and Controlled Variables

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

- $w_{wheel_{left}}$  = Speed of the left sides wheels (rad/s)
- $w_{wheel_{right}}$  = Speed of the right sides wheels (rad/s)
- $m_{container}$  = Weight of the storage device (kg)
- $b_{cupTaken}$  = If the cup has been taken (boolean)
- $d_{objects}$  = Set of distances to closest obstacle (m)
- $V_{batt}$  = Voltage levels of batteries (V)

The following is a list of variables that will be controlled

- $w_{motor}$  = Speed of the motor (rad/s)
- $percent_{duty_{cycle_{left}}}$  = The duty cycle of the left side of the drivetrain (%)
- $percent_{duty_{cycle_{right}}}$  = The duty cycle of the right side of the drivetrain(%)
- $V_{pump}$  = Voltage going to the liquid pumps. (V)

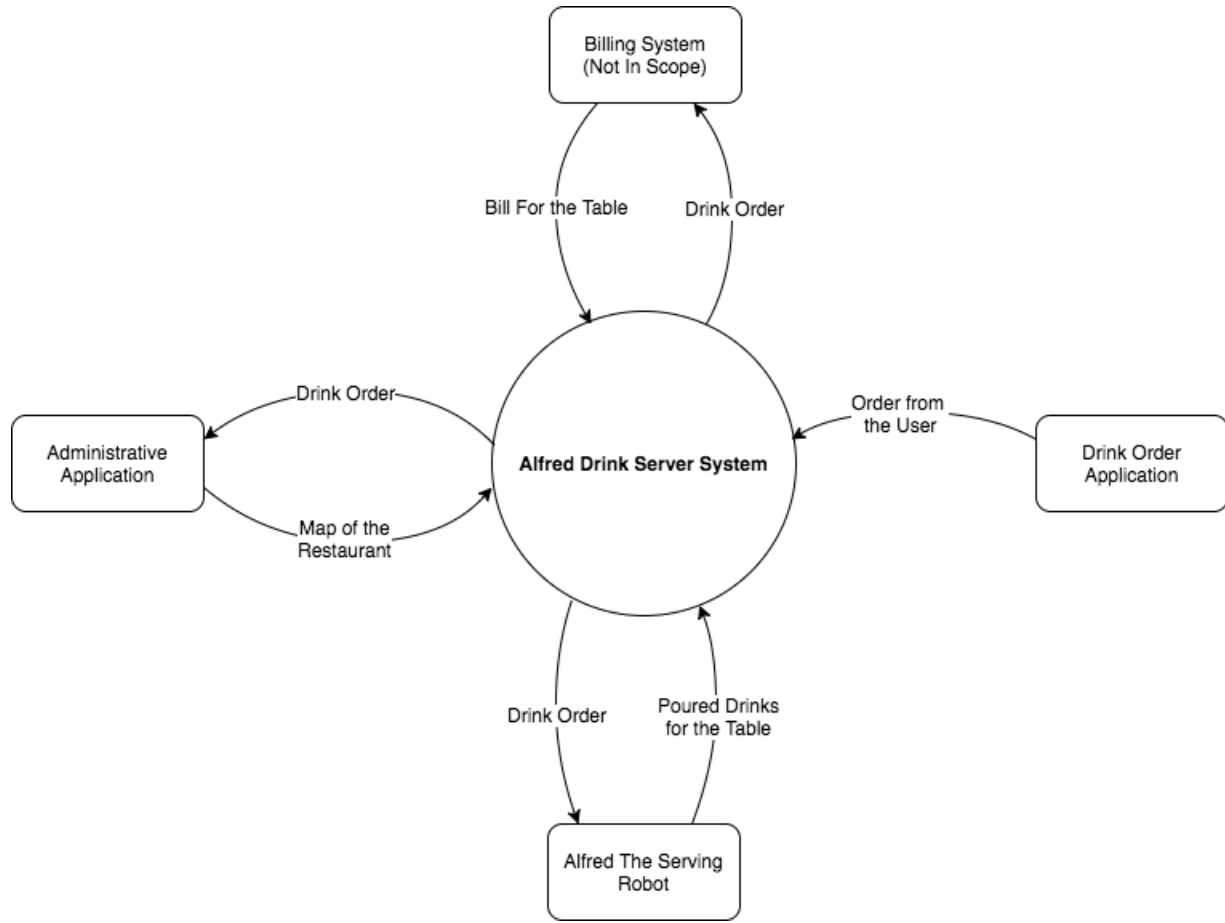


Figure 1: Drink Serving Robot Context Diagram

- $LED_{drinksignal}$  = Signal of drink that it is ready to be picked up (boolean)
- $error_{alfred}$  = Error codes sent from Alfred (unsigned byte)
- $error_{drivetrain}$  = Error codes sent from Alfred's drivetrain module (unsigned byte)
- $error_{pump}$  = Error codes sent from Alfred's pumping system module (unsigned byte)
- $Q_{pump}$  = Flow rate of the pump ( $m^3/s$ )

### 3.2 Constants

The following is a list of system constants

1.  $V_{battMin}$  - The minimum voltage necessary for drivetrain movement (V).
2.  $m_{containermin}$  - The minimum weight of liquid necessary for a pump to remain functional. (g)
3.  $m_{drink}$  - The minimum weight of the drink to be considered as ready for the customer. (g)
4.  $t_{timeout}$  - The maximum time Alfred or the ordering application will wait for a message from the server before timing out. (s)
5.  $t_{pumptimeout}$  - The maximum time Alfred will try to pump without noticing a change in the weight of the tank. (s)
6.  $t_{maxpump}$  - The maximum time Alfred will try to dispense a drink within. (s)

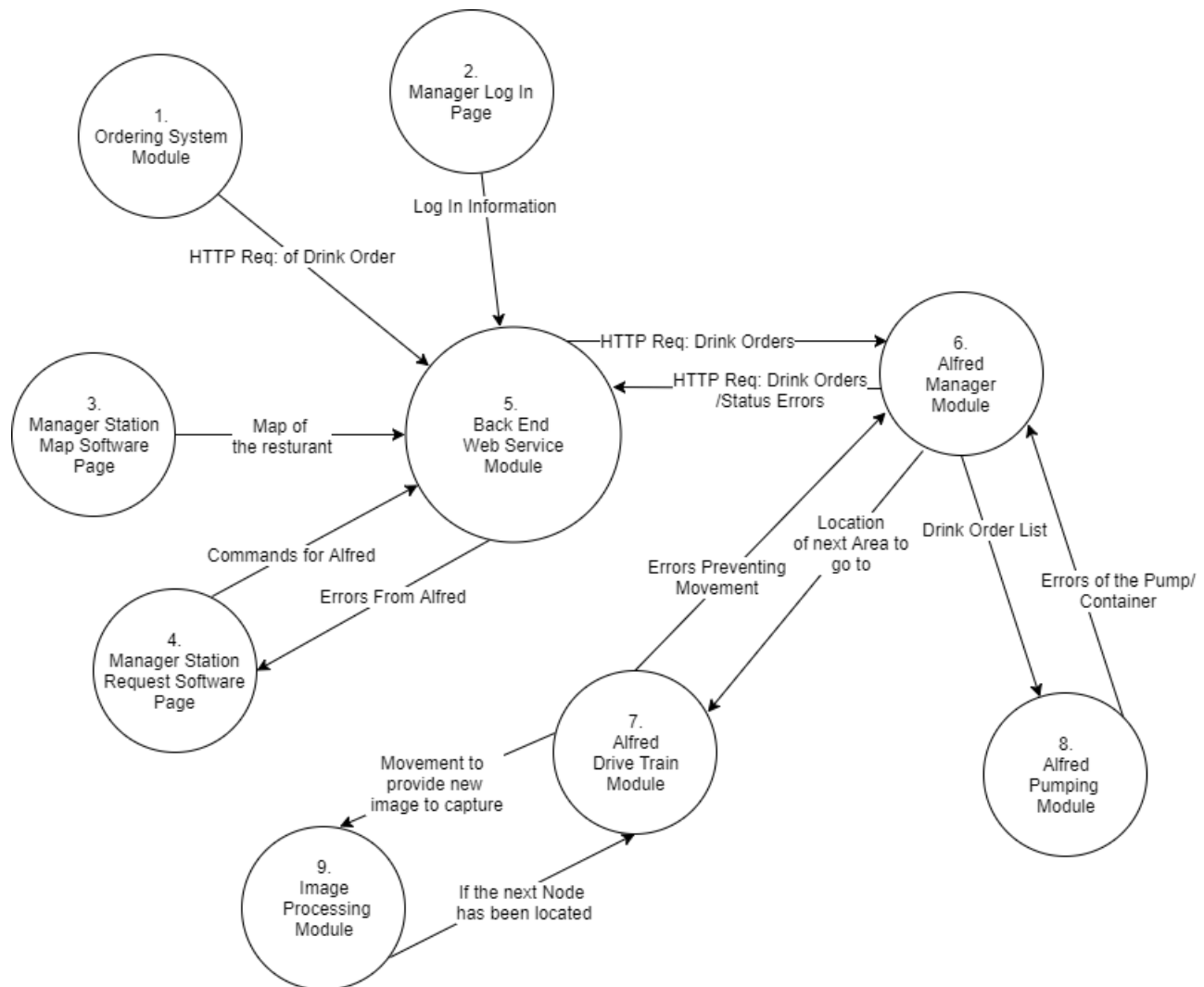


Figure 2: Drink Serving Robot System Component Diagram

7. **Steps/Revolution** - The number of steps within a revolution of the stepper motor (count/rev)
8.  $flow_{pump}$  - The amount of liquid that will be pumped at a specific voltage ( $m^3/s$ )
9.  $freq_{baudrate}$  - The rate at which UART communication will be performed at.
10.  $obstruction_{timeout}$  - 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

The system is composed of four major parts; the robot, the server, the client application as well as the administrator application. The system will behave in a manner such that the administrator will initially have the capacity to map out the area where the robot will be delivering the drinks. This is done through a graphical interface. Once the administrator creates the map, it will be sent and stored on the server. The administrator will also have the ability to specify the drinks available “” which also will be stored on the server. Once the robot is started, it will pull the map from the server to be able to navigate around the area. The client will be able to "order" a drink through the application. This will send the order to the server and will be put in a queue. The robot will be pulling the orders from that queue following a FIFO protocol.

As the robot pulls each "order", it will navigate to the table associated with it, and will pour the drinks for that order. The administrator will have the capacity to override the robot at any point, to force its return to "home". Once the drink levels are low in the robot, it will send an "error code" to the server, which will alert the administrator as well.

## 5 Component Overview

### 5.1 Ordering System Module

#### 5.1.1 Inputs and Outputs

**Inputs:** User input defining:

1. Number of Orders
2. List of Ordered Drinks

**Outputs:** Packaged Information within HTTP Request for:

1. Number of Orders
2. List of Ordered Drinks
3. Table Number
4. Identification of the restaurant

#### 5.1.2 Description

This module will be used as the user input to take 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.

#### 5.1.3 Timing Constraints

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

#### 5.1.4 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.

### 5.2 Manager Login Page Module

#### 5.2.1 Inputs and Outputs

**Inputs:**

1. String representing the restaurants Username
2. String representing the restaurant's Password
3. Indicating which manager component they would like to use

**Outputs:**

1. Failure message if there is an incorrect information
2. navigating to the correct page if it was a success.

### 5.2.2 Description

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

### 5.2.3 Timing Constraints

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

### 5.2.4 Initialization

This will default to a HTML page with no information.

## 5.3 Manager Station Map Software Module

### 5.3.1 Inputs and Outputs

**Inputs:** User input defining:

1. The width of the restaurant
2. The length of the restaurant
3. A set of areas in which Alfred can travel to
4. A set of areas in which contains tables
5. A set of areas in which Alfred cannot travel to
6. An area that defines Alfred's home location

**Outputs:** A text file that contains the following information

1. The width of the restaurant
2. The length of the restaurant
3. A set of nodes in which Alfred can travel to
4. A set of nodes in which contains tables
5. A set of nodes in which Alfred cannot travel to
6. A node that defines Alfred's home location

### 5.3.2 Description

This module will be used as 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.

### 5.3.3 Timing Constraints

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

### 5.3.4 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.



## 5.4 Manager Station Request Software Module

### 5.4.1 Inputs and Outputs

**Inputs:** User input defining:

1. calling the robot back to the home base

**Outputs:** Displaying the following information to the user

1. Low liquid levels
2. Leaking of liquid error
3. Not pumping error
4. Not able to move error
5. Low battery Error

### 5.4.2 Description

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

### 5.4.3 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  $w_{wheel} * \text{distance}$ .

### 5.4.4 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.

## 5.5 Back End Web Service Module

### 5.5.1 Inputs and Outputs

**Inputs:**

1. HTTP request package with the packaged information for drink orders.
2. A text file that holds the information about the restaurant map.
3. HTTP request package with the current list of errors from Alfred
4. Log in information

**Outputs:**

1. HTTP request package with the packaged information for drink orders.
2. A text file that holds the information about the restaurant map.
3. HTTP request package with the current list of errors from Alfred
4. Login Success or Fail
5. Success result for authenticated results.

### 5.5.2 Description

This module is a server component that will hold account information for different restaurants. This will perform 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.

### 5.5.3 Timing Constraints

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

### 5.5.4 Initialization

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

## 5.6 Alfred Manager Module

### 5.6.1 Inputs and Outputs

#### Inputs:

1. Text file with a map of the restaurant
2. A list of drink orders ordered by table
3. If there is a request to come back to the base
4. Errors from drive-train system module
5. Errors from pumping system module

#### Outputs:

1. Next Drink Order
2. Next Node to Travel to
3. Errors from Alfred

### 5.6.2 Description

This module acts as the manager to the different components of Alfred. It will command the drive-train to move to different nodes of the map. The manager will also communicate through UART to the different on board micro-controller for commanding the next drink to the pumping system.

### 5.6.3 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  $freq_{baudrate}$ .

### 5.6.4 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.

## 5.7 Alfred Drive-train Module

### 5.7.1 Inputs and Outputs

#### Inputs:

1. Text file with a map of the restaurant
2. The next node it has to travel to.
3.  $w_{wheel_{left}}$
4.  $w_{wheel_{right}}$

5.  $d_{objects}$
6.  $Marker_{PosX}$
7.  $Marker_{PosY}$
8.  $b_{NextMarkerFound}$

**Outputs:**

1.  $percent_{dutyCycle_{left}}$
2.  $percent_{dutyCycle_{right}}$

**5.7.2 Description**

This module will provide power to the drive-train of Alfred. 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 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.

**5.7.3 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.

**5.7.4 Initialization**

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

**5.8 Alfred Pumping System Module****5.8.1 Inputs and Outputs****Inputs:**

1. Drink Order List
2.  $b_{cuptaken}$
3.  $m_{containerMin}$
4.  $m_{drink}$

**Outputs:**

1.  $LED_{drinksignal}$
2.  $V_{pump}$

**5.8.2 Description**

This module 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.

### 5.8.3 Timing Constraints

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

### 5.8.4 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.

## 5.9 Alfred Image Processing Module Module

### 5.9.1 Inputs and Outputs

**Inputs:**

1.  $pic_{ceiling}$

**Outputs:**

1.  $Marker_{PosX}$
2.  $Marker_{PosY}$
3.  $b_{NextMarkerFound}$

### 5.9.2 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.

### 5.9.3 Timing Constraints

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

### 5.9.4 Initialization

This module start by the drive-train application.

## 6 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.

## 7 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.