JSS Mahavidyapeetha

# JSS SCIENCE AND TECHNOLOGY UNIVERSITY
# MYSURU-570006

FINAL YEAR B.E PROJECT REPORT

2019-2020

# Automated Supermarket Navigation and Guidance System

Submitted by


| Name | USN | e-mail | Phone-no |
|---|---|---|---|
| Srivatsa M N | 01JST16EC091 | mnsrivatsa1998.sv.sm@gmail.com | 9164152740 |
| Suchith B S | 01JST16EC092 | suchithsuchi.sbs@gmail.com | 9972410049 |
| Sumanth S | 01JST16EC096 | sumanthhegde52@gmail.com | 9481518719 |
| V Shaashwath | 01JST16EC108 | shashawath43@gmail.com | 7760732108 |


Submitted in partial fulfilment of the requirement of academic event in BE

Under the Guidance of


**Smt. Anitha S. Prasad**

**Assistant Professor**

**Dept Of Electronics & Communication**

# Acknowledgement

The success of our endeavor depends a lot on the goals set at the onset as well as the constant guidance and motivation received throughout. We take this opportunity to express our deepest gratitude and appreciation towards all those who have helped us directly or indirectly towards the successful completion of this project. We would like to thank Dr.Shankaraiah, Professor & Head of the Department, Electronics Communication Engineering, JSS Science and Technology University, Mysuru, for his insight and valuable time serving as a department head. We would like to express our deep sense of respect and gratitude towards our advisor and guide Smt.Anitha S.Prasad, Assistant Professor, Department of Electronics and Communication Engineering, JSS Science and Technology University, Mysuru, who has been the guiding force behind this work. We are greatly indebted to her for her constant encouragement, valuable advice and for propelling us further in every aspect of our academic life. Her presence and optimism have provided an invaluable influence on our career and outlook for the future. We consider it our good fortune to have got an opportunity to work with such a wonderful person. We thank Dr.Sudarshan Patil Kulkarni, Professor, Department of Electronics Communication Engineering, JSS Science and Technology University, Mysuru, who has helped us in the project with his specialization and in-depth knowledge in the field of robotics.

**Abstract**

Shopping has turned from a necessity to a leisure activity over the past few decades. The terminology of shopping has been revolutionized since the inception of online shopping but the joy of going out for shopping never changes. Whatever the form of shopping, the logistics of a product exists in one or the other stage until the product reaches the customer.

Whenever entering a supermarket, trolley is the first thing that any shopper would require to wander around the mall by placing their item in it and moving ahead to buy their other required items. Technology has improved since then to many other aspects and has made the shopper's job easier for maneuvering around the mall. Amidst such advancements, shoppers are often found in a state of turmoil when it comes to navigating through supermarkets for the desired product. Supermarkets have seen technological advancements in aspects of billing or navigation but never has there been an approach to automate the entire shopping experience of a user towards finding the product or even touching the trolley while moving around the mall.

The solution to this has been a brute force approach than an intelligent approach until now. The project aims to optimally find out the shortest path in order to get all the commodities at one go. This approach saves time whilst fulfilling the needs of the customer. Here we have proposed an approach to automate the shopping experience of a user by finding the shortest path to get all the commodities without even touching the trolley to maneuver it around the mall. The desired products are entered through a mobile application and the trolley acts as a guide to move in shortest possible path to reach the destinations.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# OVERVIEW

Through the course of this report, we shall be looking at various aspects of the project. This chapter serves as a compendium of the project's ecosystem.

## 1.1 Introduction

The world has seen the evolution of technology to an unprecedented extent in the past era of science and this field is still evolving beyond limits. In accordance with this rapid advancement, the expectation of human beings for making their jobs easier has. Everyone is aware of the impact of robotics on the lifestyle of commoners. The field of robotics has spread its roots into all the domains ranging from domestic applications to rocket science. With regards to all these advancements one such realm which has also undergone rapid advancements is shopping. This project is pivoted in achieving a much more interactive shopping experience for the consumer.

Shopping malls created a sensation when they arrived into the market as people were able to obtain most of their goods if not everything, in one single place. And trolley is the first thing that any shopper would require to wander around the mall by placing his/her item in it and move forward to buy their other required items.

Technology has improved since then to many other aspects and has made the shopper's job easier for manoeuvring around the mall. Even though such advancements in the shopping arena exits, customers are often found in a state of turmoil when it comes to navigating the supermarket for the desired product. This is where this project comes into picture. This project aims at creating an interactive robot which helps the user navigate to the required product(s) of his/her choice. The idea is to take the user input of all the products he/she requires and navigate to the required location. Such a system is required in the modern supermarkets with an ever-growing diversity of commodities and increased size of the supermarket. This architecture can be taken to the next level by choosing your own robot to guide one through a vast environment such a warehouse, college campus or any big infrastructures.

## 1.2 Motivation

There was once a time where the human work element was considered the ultimate tool in any field. Then man created machine and now we are almost slaves to our machines. In this industrious world where efficiency and results matter most, we almost completely avoid all human interactions to maximize the efficiency and decrease the time utilized. We are setting milestones maximizing the speed at which tasks are carried out only to realize it isn't fast enough.

Shopping is fairly simple; one enters the mall or a supermarket and chooses the product one needs and pays for it and leaves; That is an ideal world example which never unfolds as expected in practical situations. The entire process can be categorized into three different and distinct phases for understanding, they are:

1. Navigation.
2. Transporting.
3. Billing.

Amidst these three cardinal phases the last phase has been automated up to a viable extent, but the people have not seen much advancement in the second phase or first phase. This is where our project comes into picture.

The work for this project is to develop a self-sufficient system which is completely independent of the human element for its functioning. Our team is converging all ideas to make the navigation phase much easier and more interactive to the consumer. In a vast shopping mall, a commoner would find it highly cumbersome to make his/her way through the various departments to find his or her product; with the aid of our interactive robot one would find it a highly efficient and enjoyable experience to shop.

This enables the user/customer to have a hassle free and effortless shopping experience. The robot intelligently navigates throughout the shopping mall whilst avoiding any collisions in its path. Constructing this elegant system in order to redefine the shopping experience is what motivates us to make this project come alive.

## 1.3 Problem Statement

**Statement**: Through this project we are trying to address one of the many issues we face our day to day life. Shopping is a very subjective task executed by people; some people shop for leisure whilst some do it for buying necessary supplies. In either of the scenarios of shopping consists of three quintessential phases explained earlier.

The problem arises when we need to search for the product in a vast area as one has to venture around the mall/supermarket looking for the product. Once the product is found, navigating it to the next successive product or towards the billing section proves to be cumbersome if the products are heavy or are in a large amount. This particular phase of the shopping experience is what we are trying to address through our project. We are aimed at creating an interactive robot to guide the consumer to the desired locations of the products he/she wants. This system governed by a centralized processing and visual feedback system. The problem statement we have addressed through this project is fairly simple yet an effective advancement in the arena of shopping.

## 1.4 Objective

The objectives of this project can be divided into distinct parts such as the communication establishment, object detection and tracking, actuation, path calculation and navigation and collision prevention. The individual objectives can be explained as follows:

1. **Communication establishment:**
   - Establishing two-way communication between the Raspberry pi 4 and the central processing system.
   - Communication between android application and the central processing system.
2. **Interface Design:**
   - Designing a user friendly android app through which the selection of the desired items as well as the movement of the trolley can be controlled.
   - Designing a user interface for Administrator to maintain the server-side operations.
3. **Path planning and navigation:** calculation of the path to the desired product(s) and redirection towards the billing section(navigation).
4. **Actuation:** Autonomy in the motion of the robot throughout the supermarket. Intelligent following of customer by collision detection and avoidance mechanisms.

These are the currently defined objectives for the project, future upgradations and systemic updates might lead to invariable shifts in project objectives by fractional amount. The overall defined objectives which remain quintessential for the project are as defined above and remain as the main objectives of the project indefinitely.

## 1.5 Project Outcome and Mode of Demonstration

Mode of demonstration will be both hardware and software. Working Robot will be built which mimics the action of the trolley within a miniaturized supermarket environment. Simulation will also be presented as it will be the testimony of the robot actions. Mobile application will also be developed for android version greater than kitkat.

The outcomes of this project can be divided into different stages. The three stages of this project can be segmented in the following order:

1. **User input through Application:** The user input can be successfully accepted through the android application developed. This stage of the project addresses the usability and graphical interface provision for the user. Once the user input is accepted the selected list and provision to edit the list are provided, this increases the flexibility of the interface

2. **Processing and computation:** . This system has a centralized computation technique which allows for further scalability of performance. If higher computational power is required for larger areas and a greater list of products, the server can be upgraded without having to upgrade the hardware of the product. Cloud computing can also be used to suffice the performance shortcomings if any.

3. **Actuation and goal attainment:** From a manual shopping experience to attaining a complete robot-assisted shopping experience, the project has transcended many phases of the current scenarios. The robot/trolley actuates based on the input provided by the user and the processed output provided by the algorithm. Navigating through the least distance path reduces shopping time and distance drastically.

Overall, the problem statements put forth have been addressed with utmost diligence and creativity.

# Chapter 2

# LITERATURE SURVEY

Various technical papers have been published and nearly close to our project has been chosen and details regarding them are provided in the section. Study about existing system has also been discussed.

## 2.1 Previous Research

Controller which will be used for the implementation would be Raspberry Pi 4B. Paper “A Pilot Study: Development of Home Automation System via Raspberry Pi” by M. Mahadi Abdul Jamil and M. Shukri Ahmad [1] explains how automation can be carried out using raspberry pi. Though paper is not much related to our topic, it provides crucial information about the working of raspberry pi. It also provides glimpse about the webcam interface with raspberry pi which is a very important part of our project. Web monitoring of the devices is also discussed in the paper which might be very useful. Overall this paper provides information regarding applications of raspberry pi for automation purposes and nothing more.

In addition to automation, the robotics part of the project i.e. stability of the robot is a crucial part. Papers “A Self Stabilizing Platform" by Vladimir Popelka [2] and “Design of Two-Wheel Balance Car Based on STM3" by Guoping You and Wanghui Zeng [3] deals with usage of 6-axis gyroscope MPU6050. These paper reports on a particular solution build on the sensor MPU6050 and its parameters that can scan 6DOF (3 x 3 accelerometer + gyroscope). Usage of the PID controller for the stability has also been explained. Along with these a little glance over the Kalman filters can also be seen in the paper.

Analyzing the movement of the bot is also important for the smooth functioning of the system. Paper “A New Approach for Line Following Robot Using Radius of Path Curvature and Differential Drive Kinematics "by Jitendra Singh and Prashant Singh Chouhan[4] presents a new idea about the development of the algorithm for a differential drive mobile robot. The paper describes in detail this idea using Geometry to determine the radius of path curvature and based on it the desired difference between angular velocity of two wheels is calculated using Differential drive Kinematics.

Addition to this textbook “Robotics, Vision and Control" by Peter Corke[5] helps in understanding the kinematics of the robot system in detail. It provides us the mathematical equations required for the control of the bots. It also gives details about the implementation of proportional controllers, PD controllers, PID controllers that can be done to monitor the movement of the bots in a more efficient way. It also states about error handling methods to overcome the challenges that would occur and consists of some references to the Simulation tools which has to be adopted for analysing the practical versus the ideal working of the bots. However it is still a textbook thus it will not provide the direct relation of all the concepts to the project. It just provides an abstract.

Kalman filter plays an important role in remote control applications. Paper “Study of Kalman Filtering Algorithm and Fuzzy PID Controller for the Coriolis Acceleration Tester" by ZHANGJian, GUOLei, WEI Shimin, SONGYuan and ZHANGYin[6] provides complete details about the Kalman filter as well as fuzzy PID controller. Even though the paper deviates from our project, theoretical explanation would be a key factor to analyse the movement of the bot.

Since our project is based on monitoring the object in closed surroundings, accurate object tracking becomes major component. Paper named “Moving Object Detection and Tracking in Indoor Environment" by Shucai Wang, Liying Su, Kim Wang and Yueqing Yu[7] discuss about the approach to detect and track moving objects in indoor environment. Certain methods such as background subtraction, frame difference, improved three frame difference has been proposed in the paper. But paper fails to explain which among them is the best method to implement. “Tracking of Moving Objects With Regeneration of Object Feature Points”[8] an IEEE paper by Igor I. Lychko, Alexander N Alfimtsev and Sergey A. Sakulin also speaks about tracking of the object. Major area which this paper concentrates on is the feature extraction of the object. A different method of feature extraction along with mathematical equations has been provided by this paper. Object detection also helps in the process of tracking of the object.

Paper named “Edge based Moving Object Tracking Algorithm for an Embedded System" by Kai Xiang Yang, Ming Hwa Sheu[9] has proposed an image tracking algorithm in for an embedded platform. This also explains about the LBP transforms, edge pixel extraction and histogram.

The paper titled "Dual Model Learning Combined With Multiple Feature Selection for Accurate Visual Tracking"[10] by Jianming Zhang,Xiaokang Jin,Juan Sun,Jin Wang and Keqin Li deals with overcoming the failures that occur during feature extraction features that involve convolution techniques. In this paper, they propose dual model learning combined with multiple feature selection for accurate visual tracking. They have adopted convolution neural networks that work on dual model based learning to fuse the extracted features in multiple layers for faster tracking. But machine learning approaches for feature extraction applications require a big data set has to be trained for accurate results.

In the paper named “ Simulation Of A Human Following Robot With Object Avoidance Function "[11] by Nattawat Pinrath and Nobuto Matsuhira they propose an object avoidance algorithm for use in a human-following robot. The algorithm is tested in a well-known robot simulator called virtual robot experiment platform (V-REP), which includes obstacle avoidance and target re-acquisition. The results show that the model is ready for a physical robot prototype.

In further research, a paper titled " An Adaptive 2D Tracking Approach for Person Following Robot "[12] by Shenlu Jiang, Meng Hang, Ling Li and Toi-yong Kuc present a 2D appearance vision based tracking approach for human following robot. The approach is based on 2D image data which reduce tracking into two dimensions and minimizes the cost of computation. The paper focuses mainly to overcome the high cost of computing in the current existing models.

The development of a human following model has been attempted in the paper " Development of the Human Following Robot Control System Using HD Webcam "[13] by Jutarat Jommuangbut and Kiattisak Sritrakulchai. They have used Raspberry Pi to process visuals taken from the HD webcam and control the movement of the robot. But the processing rate in Rpi is very low and the entire process is time consuming and less accurate.

Majority of computer vision applications assume the pin-hole camera model. However, most optics will introduce some undesirable effects, rendering the assumption of the pin-hole camera model invalid. Fish eye effect is one such distortion that is commonly observed in the cameras. The paper named "Review of Geometric Distortion Compensation in Fish-Eye Cameras"[14] by Ciaran Hughes, Martin Glavin, Edward Jones and Patrick Denny discuss about the above mentioned problem. Both theoretical explanations along with mathematical modelling have been provided.

Images captured using the camera are always perspective in nature and hence has to be converted to a parallel view before being used for any applications to get a photorealistic result. Paper "An Approach for Fully Automating Perspective Images Based On Symmetry and Line Intersection by Robin Tommy and Mohan S"[15] deals with Correction of perspective images are very useful for image based rendering and view metrology. Methods used are perspective transform and plane homographs and has also an algorithm of their own.

## 2.2 Summary of Literature Review

To summarize the literature survey, the previous works done regarding a path following robot system, they track human movements and they adopt a visual aid i.e. camera to do this process. The papers referred for more in-depth information regarding the image processing principles adopted in the previous works, it was noted that majority of the existing works have adopted machine learning and deep learning techniques in implementation. However the proposed system eliminates the machine learning part as the path to be followed by the bot is dynamic in nature and would be implemented using shortest distance algorithm.

Another important notable aspect in the existing systems is that the smart trolley systems present at the moment operate based on a localized approach where each trolley tracks the human/user separately with separate cameras placed on each of them. Whereas a centralised path processing system is proposed in the currently adopted system which has a predefined path for the robot to follow. It does not require a camera to manage the robot as the movement of the robot is handled by the designed algorithm with collision detection and prevention using infrared technology. This makes the system a novel and a cost effective approach for the applications of the project that have been defined.

## 2.3 Existing systems

### 2.3.1 Warehouses

In the world of automation, AGVs (automated guided vehicles) and AMRs (autonomous mobile robots) are robotics innovations that are delivering new efficiencies in warehouse management and operations. AGVs and AMRs come in a wide variety of sizes and capabilities, each designed to support a function of a warehouse or manufacturing system. Some types of AMRs and AGVs include floor cleaners, forklifts, pallet movers and more. Two of the leading developers of the warehouse robots are Balyo Robotics and Fetch Robotics.

**Balyo Robotics:** An international software robotics company with its headquarters in France and U.S. offices in Boston, Balyo manufactures software and systems that can be used to upgrade traditional forklift equipment to offer autonomous robotic functionality.


Figure 2.1: BALYO Automatic Forklift

The figure 2.1 shows the BALYO automatic forklift equipment in action. Algorithm mobile robotics used is based on Simultaneous Localization and Mapping (SLAM), allowing manual material handling products to become real autonomous vehicles. Combining latest LIDAR (Light Detection and Ranging) technologies with its proprietary algorithm, BALYO retrieves information from robot surroundings and computes thousands of positions to self-localize precisely, without adding additional infrastructures to customers environment. it performs a 360° scan of its environment, correlating in real time what it sees (using LIDAR) to what it knows (the reference map), and localizing itself accurately (SLAM).

BALYO technology enables the robot to analyse patterns, forms, detect a human, a location, a pallet, a free space using 2D and 3D sensors… Operating in a mixed environment requires obstacle detection guarantee, whatever the height and form. the robot uses 3D camera information to transform them into a recognizable 3D-model object. All the 3D information available are compared to the 3D digital model of the object previously stored into the robot memory. integration of a barcode reading component, connected into to system, has enabled our robots to read different existing format of barcodes such as 2/5 interleaved, Coda bar, Data Matrix ECC200, QR code. Information is retrieved and transmitted to the robot manager and can be used to trigger specific actions. An ERP/WMS can thus request transport of an identifiable load that the robot will verify before carrying its mission. **Fetch Robotics:** The Fetch Cloud Robotics Platform allows customers to scale and control all operations with the click of a button. With no infrastructure changes required, the Fetch platform enables end users to control material flow, mitigate employee risk for injury and increase the efficiency of their operation.


Figure 2.2: Freight 1500 Automated Transport System

Freight 1500 shown in figure 2.2 is an automated transport system. Which can be considered as simpler version of forklift which balyo manufactures. It has eight 3D cameras and two LiDAR sensors for superior navigation and fast, safe stops when an obstacle is detected. Their ultra-low-profile allows for easy loading and modular integration while a 360-degree LED strip provides high visibility indicator lighting for increased safety. The goods need to be placed on the robot manually unlike balyo's forklift machine which dose things fully automatic.

### 2.3.2 Agriculture

**Weeding and Crop Maintenance:** Weeding and pest control are both critical aspects of plant maintenance and tasks that are perfect for autonomous robots. The Bonirob robot is about the size of a car and can navigate autonomously through a field of crops using camera, LiDAR and satellite GPS. Its developers are using machine learning to teach the Bonirob to identify weeds before removing them. With advanced machine learning, or even artificial intelligence (AI) being integrated in the future, machines such as this could entirely replace the need for humans to manually weed or monitor crops.

Figure 2.3: Bonirob

The UC Davis prototype operates a bit differently. Their cultivator is towed behind a tractor and is equipped with imaging systems that can identify a fluorescent dye that the seeds are coated with when planted, and which transfers to the young plants as they sprout and start to grow. The cultivator then cuts out the non-glowing weeds. **Driverless Tractors:** The tractor is the heart of a farm, used for many different tasks depending on the type of farm and the configuration of its ancillary equipment. As autonomous driving technologies advance, tractors are expected to become some of the earliest machines to be converted. autonomous tractors will become more capable and self-sufficient over time, especially with the inclusion of additional cameras and machine vision systems, GPS for navigation, IoT connectivity to enable remote monitoring and operation and radar and LiDAR for object detection and avoidance.

Figure 2.4: Driverless Tractor

## 2.4 Comparison

An analysed comparison between the existing systems that are operating as full fledged products in the market and the proposed system has been pointed out in the table below.

| Parameter | CartBot | BALYO | Freight |
|---|---|---|---|
| Camera visualization | absent | present | absent |
| Cost | low cost | high cost | high cost |
| Dynamic mapping environment | Present | Present | Present |
| Type of control | Mobile application controlled | System controlled | System controlled |
| Manual mode | Present | Present | Absent |
| Server | Local server | Cloud server | Cloud server |
| Obstacle detection | Present | Present | Present |
| Authentication | Present | Absent | Absent |

Table 2.1: Comparison With Existing Systems.

# Chapter 3

# SYSTEM ARCHITECTURE AND METHODOLOGY

The structural architecture of the system and adapted methodologies are explained in this context. The methodologies help us dive deeper into understanding how the project augments.

## 3.1 Client-Server Architecture

Client-Server Architecture is a distributed system architecture where the workload of client server is separated. Clients are those who request for the services or resources and Server means the resource provider. The server hosts several programs at its end for sharing resources to its clients whenever requested. Client and server can be on the same system or may be in a network.

Client Server architecture is centralized resource system where Server contain all the resources. The server is highly secured and scalable to respond clients. Client/Server Architecture is Service Oriented Architecture that means client service will never be disrupted.

In this type of client server environment user interface is stored at client machine and database are stored on server. Database logic  business logic are stored at either client or server but it must be unchanged. If Business Logic  Data Logic are stored at client side, it is called fat client thin server architecture. If Business Logic Data Logic are stored on server, it is called thin client fat server architecture. This kind of architecture are affordable and comparatively better.

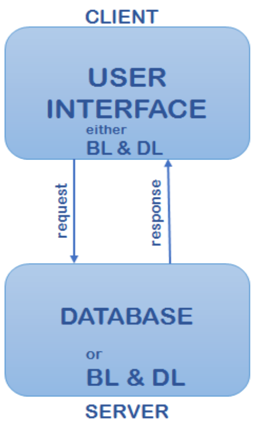



Figure 3.1: Server Client Architecture

Two-tier architecture is useful where a client talks directly to a server. There is no intervening server. It is typically used in small environments. Here, the user interface is placed at user's desktop environment and the DBMS services are usually placed in a server. Information processing is split between the user system interface environment and the database management server environment.

## 3.2 Proposed Block Diagram

The proposed system is to come up with a centralized architecture for the smart trolley actuation as mentioned previously in the objectives. The entire process can be summarized in terms of a single block diagram as shown in figure 3.2. This project can majorly be divided into six main stages.

**Stage 1:** Customer enter into the supermarket. If the user is new to the supermarket, he/she will be given a unique customer id. If he has already visited the supermarket then he/she can use the already given customer id itself.

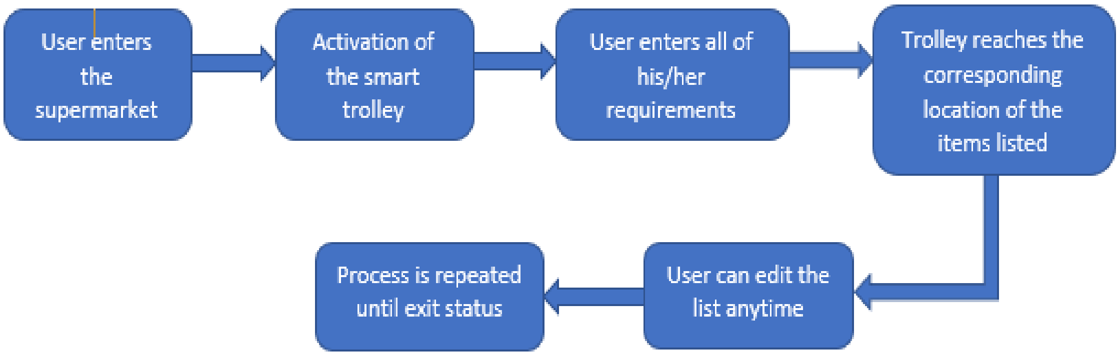



Figure 3.2: Block Diagram

**Stage 2:** Activation of the trolley includes assignment of the trolley to the particular user. Basic level of authentication is done so as to confirm whether he/she is a valid user. Costumer id is used for the authentication of the user.

**Stage 3:** As soon as the trolley is activated the user is allowed to enter all his needs at once. All the items are stored in a list one after the other. Items are checked with the database simultaneously. If there is no match then a notification is displayed to re-enter the item.

**Stage 4:** Once the list is ready, the trolley chooses the shortest path to the corresponding to the items. It reaches the item which is nearer rather than the item that has been listed first. According to this the trolley will be smart enough to avoid the collision with other users/trolleys.

**Stage 5:** This stage is optional provision provided to the user. User can edit the list of items whenever he/she wants to. The is no restrictions to the number of items that can be listed. Algorithm will be developed in such a way to adopt to the updated list and find the new shortest path accordingly.

**Stage 6:** Exit status will stop the movement of the trolley. User can either stop the trolley or can send it to the billing area which will be the final destination. Exit status also has the facility of tracking down the user if in any case he/she has lost the track of the trolley.

## 3.3 Flow Chart

Detailed procedural approach of the proposed system can be depicted into a flow diagram as shown in figure 3.3. As soon as the user enters the supermarket, he activates the trolley with the help of the customer ID. This primary level of authentication helps to validate the customer. Once the trolley is activated user is allowed to enter the list of items required. Items are checked with the database for their presence in the supermarket. If not, available item is cleared from the list and user is allowed to enter new item. If all the items are matched then trolley is actuated to move to the desired location. Effective shortest path determining algorithm is developed so that least amount of distance is covered during the shopping. Once the trolley is reached to the position of the first item it has to wait for the command from the user for its next action. User will be provided with following options.

1. **Next**: Trolley moves to the position of the next item.
2. **Stop**: Trolley stops its movement irrespective of the position.
3. **Resume**: Trolley movement resumes from the stopped position.
4. **Add Item**: User can add new item to the list containing the existing items at any moment.
5. **Billing**: Trolley reaches the billing station automatically.
6. **Abort**: Shuts down the complete system. User have to take the trolley manually.

Apart from these facilities there is also an option for the user to make the trolley to return to the position of the user in case of losing the track of the trolley. The entire process is centralized with single processing unit which acts as the server for the different hosts (trolleys).

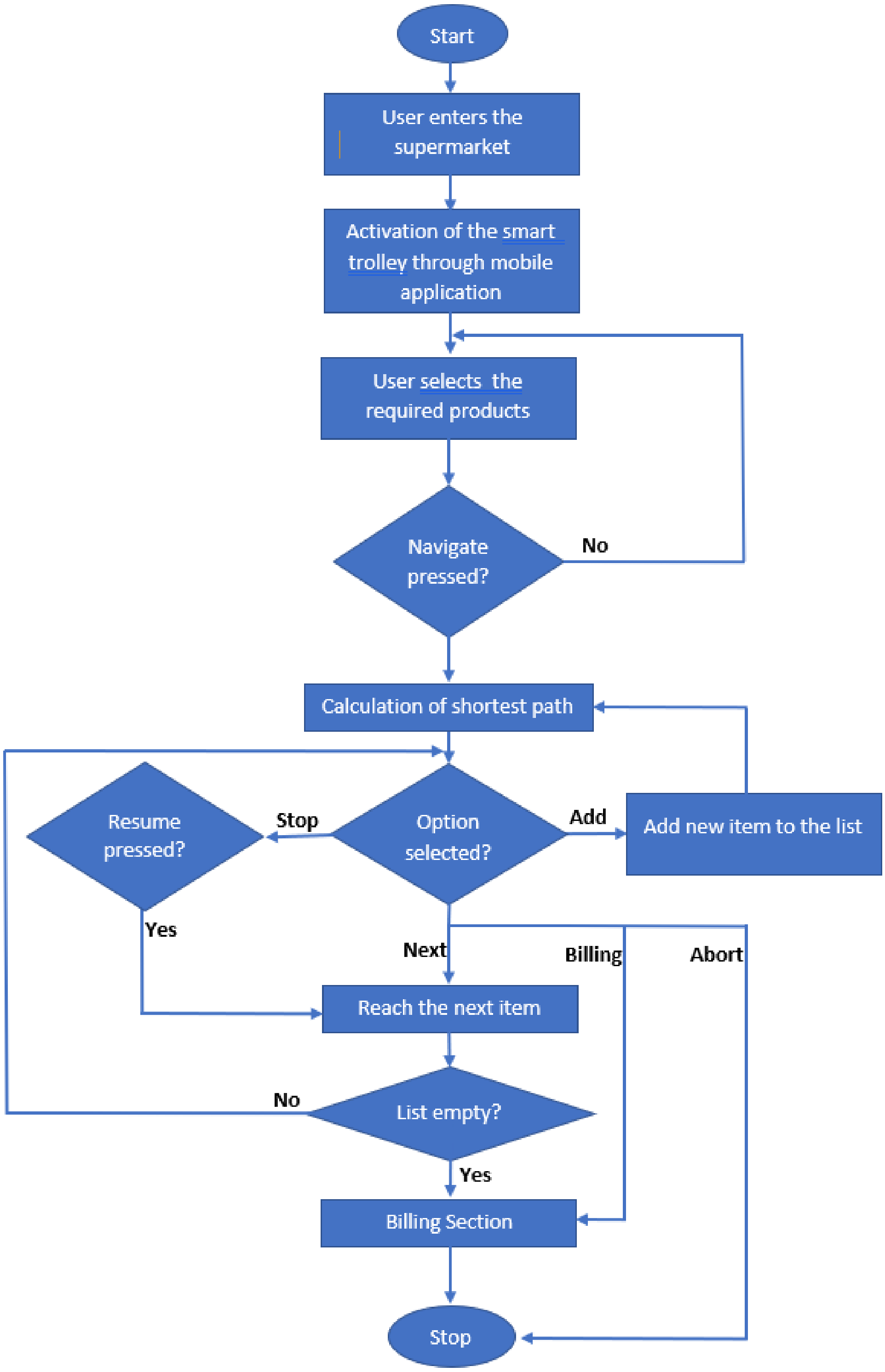



Figure 3.3: Flow Chart

## 3.4 Methodology

### 3.4.1 Bot development

The project is trying to focus on implementation of the entire working of the proposed system. It is a really nice blend of both robotics as well as computer vision. Detailed integration of the hardware and software along with the procedural execution can be interpreted as follows:

1. **Authentication of the user with their respective trolley:**

   In order to link the trolley to its user, authentication needs to be done. As soon as the user enters the supermarket, the device will be authorized to get access to any of the free trolleys. Each user will scan the QR code on trolley through the app which processes a unique key and authorizes user and their trolley dynamically. This architecture makes sure that at any given point of time, there will be unique pairs of trolley and its user.

2. **Taking the input from the user through mobile application:**

   User interface will be an Android application, provided to the customers through a QR code. QR scanner will be placed at the entrance of the supermarket. Interface displays every item to the user section wise. User has to just select the products that are required and proceed with the navigation.

3. **Calculation of the shortest path:**

   A new customized algorithm has been developed on the basis of breadth firs search algorithm of graph. The algorithm calculate path for n! combinations of the total number of goals and selects the shortest among them.

4. **Movement of the trolley:**

   After the calculation of the shortest path, an array of coordinates is generated, which will be the path that trolley should follow. Gyroscope is used for the proper movement and alignment of the robot. PID controller is used for the alignment of the trolley. Detailed explanation of the PID controller is given below.

(a) **PID Controller:**

The PID controller designed for the straight path motion of the bots has a set of equations which are implemented on the values provided by the gyro sensor. Any PID controller that is designed for a kinematic has three PID constants namely proportional constant($K_p$), Integral constant $(K_I)$ and a differential constant($K_D$).

These equations are defined in the control system domain as follows

$$proportional = 1$$

$$differential = error$$

In our project, we are not dealing with sum of errors of the values provided by the gyro sensor. Thus, the integral part of the PID is neglected and hence our controller is defined as a PD controller. We have fixed these values for the respective constants

- Proportional constant, $K_p = 50$
- Integral constant, $K_I = 0$
- Differential constant, $K_D = 1$

The PID equations involving these constants that are used for our application are

For left wheel,

$$PD\ valueL = K_p \times proportional + K_D \times differential$$

For right wheel,

$$PD\ valueR = K_p \times proportional - K_D \times differential$$

(b) **Gyroscope Calibration:**

Gyroscope measures the angle of rotation. We are measuring the z-axis rotation i.e., lateral movement of the robot. The readings of gyroscope vary from 0 to 16384 for one complete 360-degree rotation. If rotation stops at the middle then the value will be reset to zero. Hence the change in values must be counted with a temporary variable. Even after storing the values it will not be in a degree format. For feedback to the PID system we need the parameter to be in degree format. The conversion factor would be

$$Angle\ (in\ degrees) = \frac{readings\ from\ gyro \times 360}{16384}$$

(c) **Track detection using IR sensor:**

LM358 is an Operational Amplifier is used as voltage comparator in the IR sensor. the comparator will compare the threshold voltage set using the present (pin2) and the photodiode's series resistor voltage (pin3).

- Photodiode's series resistor voltage drop > Threshold voltage = Opamp output is High
- Photodiode's series resistor voltage drop < Threshold voltage = Opamp output is Low

When Opamp's output is high the LED at the Opamp output terminal turns ON (Indicating the detection of Track).

(d) **Obstacle detection using Ultrasonic sensor:**

Ultrasonic signal gets bounced back when it hits the object. Distance can be calculated on the basis of time taken by the signal. Time is calculated from the time of the trigger to the time when the echo pin becomes high. This time should be divided by 2 because time recorded would be for the total trip of the signal (Two way). Equation for the calculation of the distance of the object is given below.

$$distance = time \times velocity$$

where,
time = Time taken by the signal to bounce back/2
velocity = Velocity of sound (340 m/s)

5. **Flexibility during navigation:**

Several flexible services are provided to the user during the navigation of the trolley. The services are as follows:

(a) **Next:** Move to the next item.

(b) **Add item:** New items can be added to the existing list. This can be added only when the trolley reaches the previous goal. The list will be updated and new shortest path will be calculated.

(c) **Stop:** User can stop the trolley whenever necessary. This allows the user to pick up the items that he sees during the navigation. Trolley will be in idle state until resume is clicked.

(d) **Resume:** The trolley will resume its motion and automatically and move towards the desired item.

(e) **Abort:** Process can be aborted at any time. If aborted the user has to manually take the trolley to the billing station.

(f) **Billing:** User can send the trolley to the billing station at any time during the navigation.

### 3.4.2 App Design and Development

The proposed application is desired to be a smooth experience for its users with minimal levels but maximum reap of the functionality it offers. This intention begins with the design of a minimalist logo that succeeds in catching the user's attention. As the user will have an understanding of his/her requirements prior to beginning of their shopping experience, it has been decided by the team to not have icons of the items that are present in the shopping market. This helps in making the app efficient in terms of speed of functioning and switching between interface levels.

#### System Architecture

Consider the following flow chart which elucidates the mechanism of the application. The flow chart in Figure 3.4 shows all the possible states of the application which the user can encounter in the shopping cycle.

#### Interface Design Inspiration

The interface design inspiration was drawn from the conventional "Lifestyle" applications. The UI consists of a smooth and elegant flow throughout the applications including the graphics. The UI also consists of a navigation pane . The navigation pane or control panel as called in the application is responsible for controlling the motion dynamics of the robot.

#### Data Integration

The application continuously communicates with the server of the supermarket right from authentication stage till billing of the items purchased. The server keeps receiving data sent by the android application and allows the user to control the trolley based on the action required. The user can control the trolley through the application to make it stop, resume it's movement, proceed to the next item, add more items to the shopping list dynamically, enter the billing section earlier if required and finally to abort the entire system if there is any sort of crash. Thus the application covers all the user scenarios and is efficiently designed to serve the purpose.

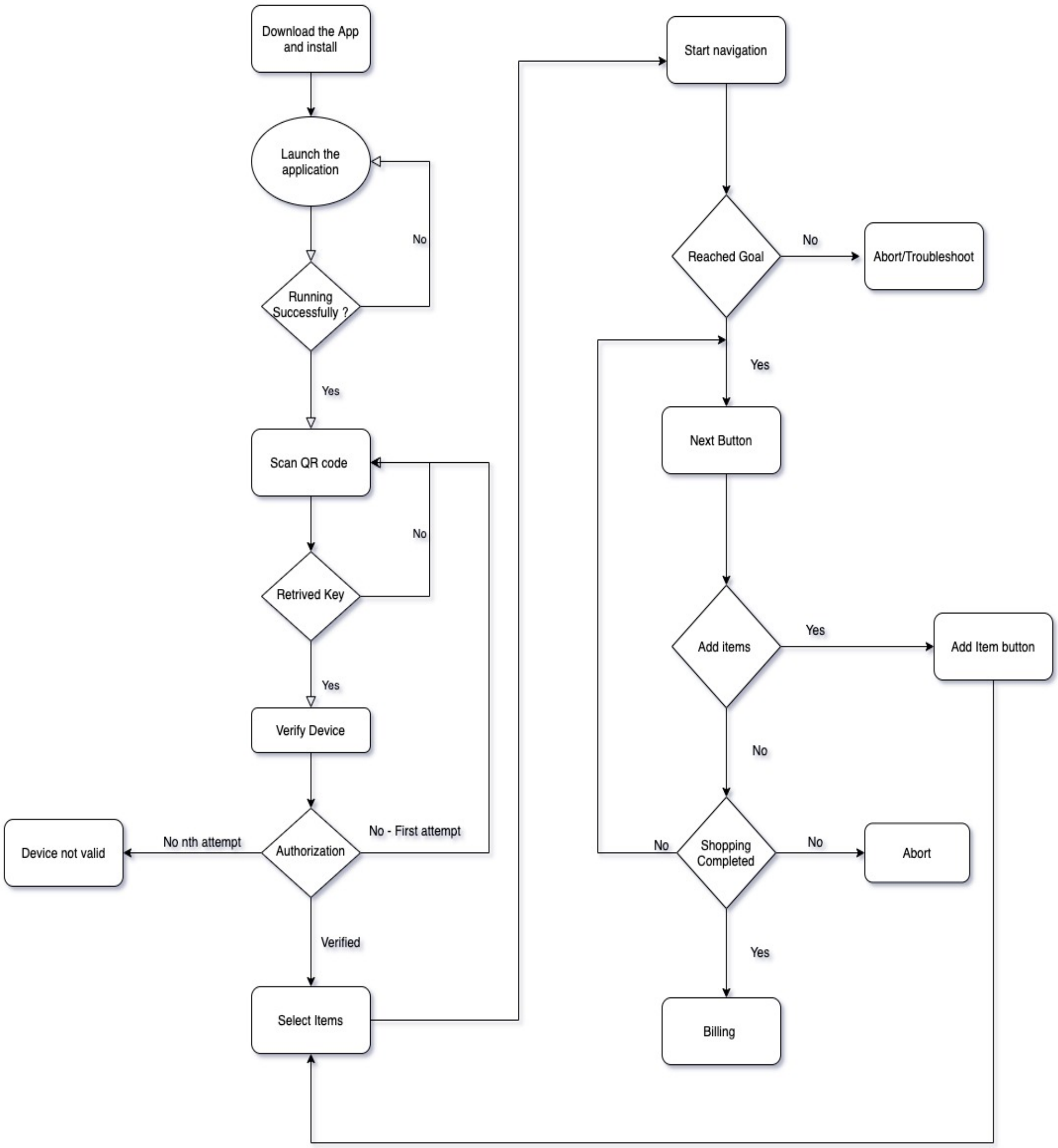



Figure 3.4: Flow Chart of Mobile Application

#### Icon Design

Icon design was a very straightforward approach in the design process. The name of the application and the design of the logo can be viewed in a simple perspective figure 3.5. CartBot is the conjunction of two words cart and robot, since the robot is as a matter of fact a shopping cart here, hence the name. The logo is an out of the book design of a shopping cart which symbolizes the ease of understanding the name and the logo together to give the application a meaning.

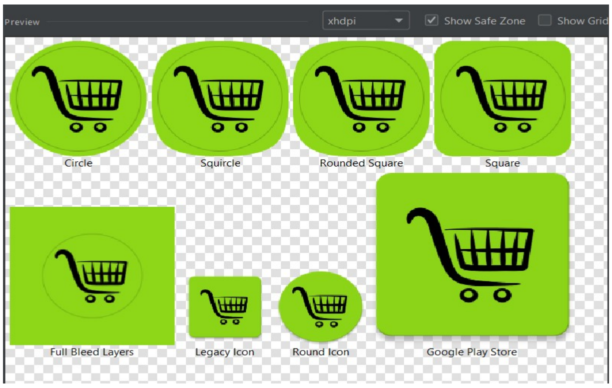



Figure 3.5: Icon Design

### 3.4.3 Application Dependencies and In-app Permissions

The developed application needs certain permissions to be given by the user on the phone for the functioning of the application which contains the dependencies. They are as mentioned below:

1. **Dependencies:**
    - Constraint Layout for XML layouts
    - Minimum Android SDK version - API level 21 (Android 5.0)
    - Target Android SDK version - API level 29 (Android 10)
    - Barcode scanner dependency - Zxing Scanner view

2. **User Permissions:**
    - Camera - For scanning purpose
    - Network - For TCP connection purpose
    - Alert Window - For displaying alert dialog box

# Chapter 4

# SOFTWARE AND HARDWARE COMPONENTS

The most rudimentary apportionment of the project is the hardware and software constituents. Chapter four throws light on how this segregation and diversification are carried out.

## 4.1 Software Requirements

### 4.1.1 Ubuntu 16.04 LTS

Ubuntu is an open source operating system which provides wide range of custom ability. The reason for using Ubuntu is unlike windows it is directly compatible with Raspbian OS which will be installed on to the Raspberry pi. There is no need of graphical interface (putty in windows) if we are working in Ubuntu. Moreover, python can be made inbuilt package of Ubuntu with few command line commands and can easily be implemented without external software (such as PyCharm).

### 4.1.2 Balena Etcher:

It is commonly referred to as just 'Etcher'. It is a free and open-source utility used for writing image files such as .iso and .img files, as well as zipped folders onto storage media to create live SD cards and USB flash drives. Etcher is primarily used through a graphical user interface. Additionally, there is a command line interface available.

### 4.1.3 Raspbian OS

Raspbian is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Buster and Raspbian Stretch. Only Raspbian buster supports Raspberry Pi 4. It is available in the official website of Raspberry pi and is a free image file. This file will be written on to the Micro-SD through Balena Etcher. This operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

### 4.1.4 Python 3.8.2

Python is an interpreted, high-level, general-purpose programming language. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python framework is being used because the Raspbian OS works on python and image processing can be done with more precision with python. It has wide variety of tools which supports for our needs in image processing such as OpenCV, NumPy, Turtle,Tkinter etc.

### 4.1.5 VNC Viewer

Virtual network computing (VNC) is a type of remote-control software that makes it possible to control another computer over a network connection. Keystrokes and mouse clicks are transmitted from one computer to another, allowing technical support staff to manage a desktop, server, or other networked device without being in the same physical location. VNC works on a client/server model. A VNC viewer (or client) is installed on the local computer and connects to the server component, which must be installed on the remote computer. The server transmits a duplicate of the remote computer's display screen to the viewer. It also interprets commands coming from the viewer and carries them out on the remote computer. We are using this for communication between Raspberry Pi and PC by hosting server using this VNC server.

### 4.1.6 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains's Intelligent software and designed specifically for Android development. Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where you can develop for all Android devices.
- A unified environment where you can develop for all Android devices.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- C++ and NDK support.

## 4.2 Hardware Requirements

### 4.2.1 Raspberry Pi 4B

The Raspberry Pi 4 Model B (Pi4B) is the first of a new generation of Raspberry Pi computers supporting more RAM and with significantly enhanced CPU, GPU and I/O performance; all within a similar form factor, power envelope and cost as the previous generation Raspberry Pi 3B+. Along with faster processing speed, it has an inbuilt WiFi module which helps in establishing the wireless communication with central processing unit.

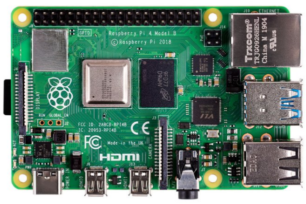


Figure 4.1: Raspberry Pi 4 Model B

1. **Specifications :**
    - Quad core 64-bit ARM-Cortex A72 running at 1.5GHz.
    - 2 Gigabyte LPDDR4 RAM .
    - H.265 (HEVC) hardware decode (up to 4Kp60).
    - H.264 hardware decode (up to 1080p60).
    - VideoCore VI 3D Graphics.
    - Supports dual HDMI display output up to 4Kp60.
2. **Interface:**
    - 802.11 b/g/n/ac Wireless LAN.
    - Bluetooth 5.0 with BLE.
    - 1x SD Card.
    - 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution.
    - 2x USB2 ports and 2x USB3 ports.
    - 1x Gigabit Ethernet port (supports PoE with add-on PoE HAT).
    - 1x Raspberry Pi camera port (2-lane MIPI CSI). item 1x Raspberry Pi display port (2-lane MIPI DSI)
    - 28x user GPIO supporting various interface options

3. **Power Requirements:** The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.

### 4.2.2 MPU 6050 Gyro Sensor

The MPU 6050 consists of a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die together with an on board Digital Motion Processor (DMP) capable of processing complex 9-axis Motion Fusion algorithms. The parts integrated 9-axis Motion Fusion algorithms access external magnetometers or other sensors through an auxiliary master I2C bus that supports up to 400KHz, allowing the devices to gather a full set of sensor data without intervention from the system processor.

This is used to implement PID controller for the straight movement of the trolley. Due to uneven mechanical stress/strain on the chassis and uneven surface bot fails to achieve a straight line motion and also perfect 90°turn. With the help of this sensor the deviation of the bot can be measured and thus based upon the error PID can be implemented.

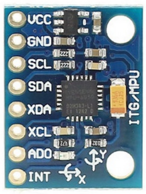



Figure 4.2: MPU 6050 Gyro Sensor

**Specifications:**

- Operating voltage 2.5V to 3.6V.
- Sensitivity 16384 LSB/g.
- Operating temperature -40 to 85C.
- Operating current 3.9mA.

- Gyroscopes range: +/- 250 500 1000 2000 degree/sec.
- Acceleration range: +/- 2g, +/- 4g, +/- 8g, +/- 16g.
- Supply current-20mA.

### 4.2.3 IR Array

Analog Digital Line Array has 8 IR LED/Phototransistor pairs. Each sensor has its own digital and analog output also a Potentiometer to adjust the sensitivity of the individual sensors. Each Phototransistor equips a pull-up resistor to form a voltage divider that produces an analog voltage output between 0 V and VIN (which is typically 5 V) as a function of the reflected IR. The lower output voltage is an indication of greater reflection. The outputs are all independent. This IR array is used in measurement of number of blocks that the bots has covered in its movement.

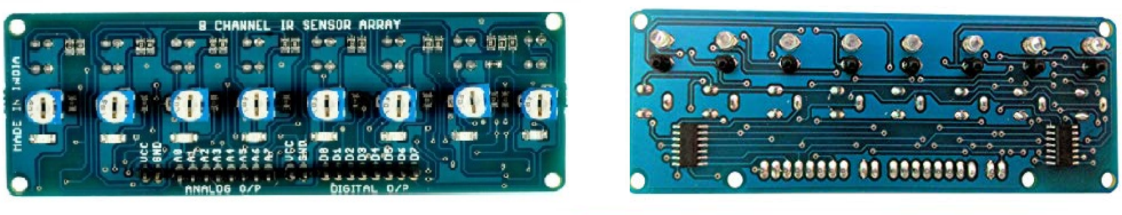



Figure 4.3: IR Array

**Specifications:**

- Operating voltage 4.5V to 5V.
- Optimal Sensing Distance (mm) 3.
- Operating temperature - -40 to 85C.
- Max. Recommended Sensing Distance (mm) 6.
- Distance between two IR sensor (mm) 15.
- Supply current 150mA.

### 4.2.4 HC-SR04 Ultrasonic Sensor

HC-SR04 Ultrasonic sensor is a 4 pin module which has VCC, trigger, echo and GND. It is used for distance measuring and obstacle detection. The module has an ultrasonic transmitter and receiver at the front. The Module automatically sends eight 40 kHz pulses and detects whether there is a pulse signal back.

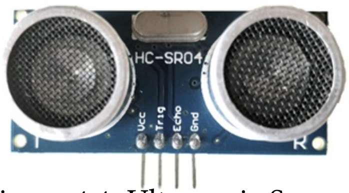



Figure 4.4: Ultrasonic Sensor

**Specifications:**

- Operating Voltage: DC 5 V.
- Operating Current: 15mA.
- Operating Frequency: 40Hz.
- Max Range: 4m.
- Min Range: 2cm.
- Measuring AngleL -15.
- Trigger Input Signal: -10uS TTL pulse.

### 4.2.5 Micro SD Card (32GB)

A 32GB class 10 Samsung Micro SD card along with the card reader. Card reader is necessary to insert the Micro SD into the laptop and burn it with the Raspbian OS. Class 10 or above is suggested for faster experience and data transfer.


Figure 4.5: Micro SD Card

### 4.2.6 Motor Driver L298N

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. This depends on the voltage used at the motors VCC. The module has an on-chip 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the on-chip 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

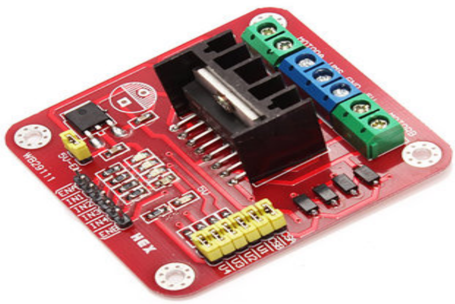


Figure 4.6: L298N Motor Driver

### 4.2.7 Robot Chassis (Fiber Body)

The chassis that has been used is made of light weighted fiber with double Decker architecture. It supports 4 wheels arrangement as well as attachment of many sensors. The chassis can be extended to one more layer if necessary.

Figure 4.7: Fiber Robot Chassis

### 4.2.8 DC Gear Motors with Plastic Wheels and Wheel Encoder

These DC gear motors are specifically compatible with the above-mentioned chassis. These can also carry the wheel encoder above the axle. Plastic wheels are very light-weighted and can move smoothly at an optimum pace.

Figure 4.8: Motor Set

### 4.2.9 Battery

The battery used for the project is a rechargeable lead acid battery with a power capability of 12V and a 1.3Ah current rating. However only about 30% of the battery power is utilized for the smooth functioning of the system.


Figure 4.9: Battery

### 4.2.10 Power bank

The Raspberry-Pi 4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA. Therefore, a power bank with 5V, 2.5A supply is used as power supply for Raspberry-Pi 4B.

Figure 4.10: Power Bank

# Chapter 5

# IMPLEMENTATION AND TESTING

From chapters one to four the report unravels as to how the project is constructed. In chapter five we shall see how the project is implemented and the various tests it's subjected to.

## 5.1 Software Implementation

### 5.1.1 User Interface of Client:

The interface provided for the client to interact with the trolley system is an Android Mobile application. It is named as 'CartBot'. The proposed application is a novel idea that integrates both an android application and hardware robotic prototype installed on the trolley that guides the user to walk around through the supermarket to shop all the items in their shopping list. Thus the application has been termed as 'CartBot' serve justice to its functioning.

CartBot is the culmination of a precisely and meticulously drawn plan for an application to satisfy its purpose. The design of the application began from the inception of the idea that a user interface must be created for a user who is shopping with the robot for ease of access and maneuverability. This enables the user to have a hassle-free shopping experience. The robot intelligently navigates throughout the shopping mall while avoiding any collisions in its path. The way the application is made available to the user is through the use of a QR code which consists of the URL which allows the user to download the application upon entering the store. The entire overview of the application can be divided into three primary segments.

1. **Networking**

   CartBot is connected to the server which is run on a central processing system. The entire connection happens over the same network which means, the central system, the application (user's phone) and the robot will be connected to the same physical wireless network. Hence making the communication and exchange of data much easier. The WiFi network will filter out the unauthorized devices not pertaining to the MAC address and hence the application on the user's device is connected securely to the shopping mart's network.

2. **Authentication**

   In Order to verify the user and to pair him/her with a unique trolley the user has to be first verified and that happens through the "Verify" CTA button. Once the application has been downloaded and installed onto the device of the user the initial steps would be to verify scan another QR code which provides a unique key to the device. Once the device is verified then the user now has "Unlocked" the trolley and hence can proceed to input the items he/she wishes to buy.

3. **User Interface Levels**

   The interface levels for the application is planned to cover the domains of security in the form of authentication of user before accessing the trolley which further opens up different intents that each provide a functionality of their own.

   The application is designed to have a unique way of authentication with an encoded QR code that needs to be scanned that shall be verified the supermarket server running in the background. Once the authentication is successful, the user gets to use the application to choose his required items and navigate to their respective positions in the supermarket guided by the smart trolley that communicates with the android application at each step. The novelty in this integrated working system is that the user gets to spend least amount of energy in procuring all the items under their list with the support of a shortest path algorithm running on the server that navigates the user to each item in the shopping list with the least number of steps possible.

The following interface levels are planned to be built for the application

(a) Opening Screen with the application logo and title .

(b) Authentication screen to validate the device.

(c) Shopping items list consisting of items available at the supermarket.

(d) Confirmation screen consisting of items to be shopped by the user before start of navigation.

(e) Control screen to interact with the trolley to move around the items and control the trolley as per requirement dynamically.

As the user is subjected to these steps from the launch of the application to the end of the shopping journey, the procedural cycle is now complete.

**Application Interface Implementation**

The application interface focuses on keeping the interaction between the shopper and the system to be smooth and dynamic with as less complexity involved throughout the process. Thus, minimal screens with accurate requirements are designed in the application

- **Screen 1: Authentication**
  TCP/IP connection will be established between server and host as soon as the user scans the QR code on the trolley. This connection will be established on a definite port (ex:7989) and it is also made sure that connection is maintained till the termination of the process. Authentication process is used to detect whether the user is valid and also to make sure that particular trolley is associated with the same user till the end. Figure 5.1 shows the authentication screen.


Figure 5.1: Authentication Screen

- **Screen 2: Selection of Products**

  Products that are only available at that time will be displayed to the user as shown in figure 5.2 . Selected Items will be stored in a list format.

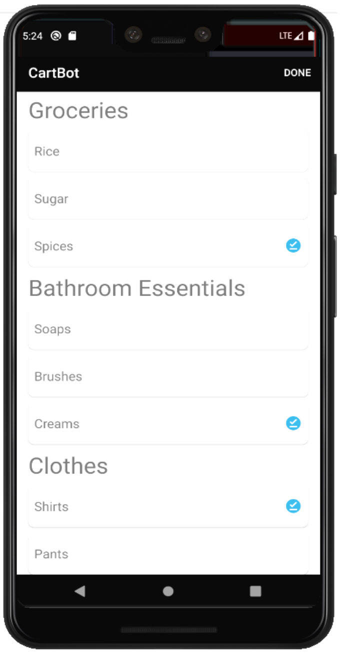



Figure 5.2: All Products Screen

- **Screen 3: Confirmation**
  Application will display the list of products that are selected to the user as shown in figure 5.3. User can either wish to proceed or can go back and select or deselect products. User clicks on navigate button once user confirms the list of items. Once the navigate button is pressed, server will receive the data representing the list of items.

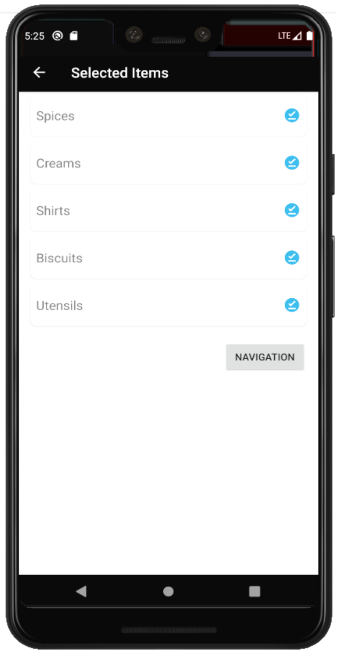



Figure 5.3: Selected Products Screen

- **Screen 4: Controls**
  There are many controls provided to the user convenience. Figure 5.4 shows the controlsthat are displayed on the user screen.

  1. **Stop button:** This will stop the movement of the trolley. This helps if user finds any new product on the way which is not in his bucket list.
  2. **Resume button:** Trolley will resume its movement from rest.
  3. **Next button:** This will take the trolley to the next nearest item in the list.

4. **Add item button:** Users can add new items whenever they find the necessity
5. **Billing button:** This will take the trolley to the billing section irrespective of the number items left in the list.
6. **Abort button:** This is during an emergency failure of the system. It shuts down all the actions of the trolley and terminates the connection.
7. **Return button:** This helps if user loses track of the trolley, as return button will make the trolley to go back to the previous product.

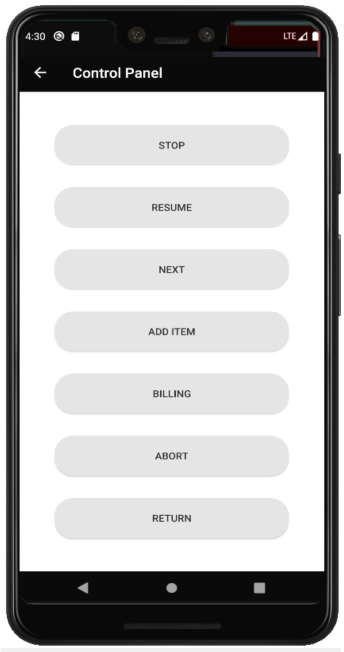



Figure 5.4: Control Screen

### 5.1.2 User Interface of Server

The administration side will be provided an interface to handle the designing of the mall layout based on the requirement. Python provides various options for developing graphical user interfaces (GUIs). The most commonly used is the Tkinter. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are a number of widgets which you can put in your Tkinter application. Importing Tkinter is same as importing any other module into the python module.
Steps to Create a GUI application using Tkinter:

1. Import the Tkinter module.
2. Create the GUI application main window.
3. Add one or more of the above-mentioned widgets to the GUI application.
4. Enter the main event loop to take action against each event triggered by the use.

We have a tier-2 client-server architecture. In two-tier architecture client has to send requests only and has no intelligence. Required database, logical and mathematical calculations are stored within the server. Server control is given only to the admin and requires credentials to access the system.

Interface has been created at the admin side so that they can handle the structure of the entire system. Layout of the supermarket can be edited at any time according to the new products that has been placed or moved to some other place.

- **Login page:** If anyone wants to access the server then the user needs to enter the admin credentials in order to acess the system. Credentials are stored within the database of the system with MD5 encryption. Thus, providing utmost security.

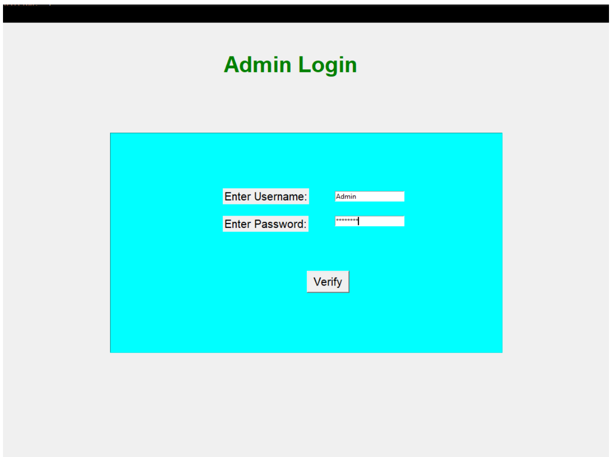



Figure 5.5: Admin Login Page

- **Admin panel:** Once admin gets into the system, user is provided with two options. First one being hosting the server so that client can send requests and second thing being editing the layout for maintenance purpose.

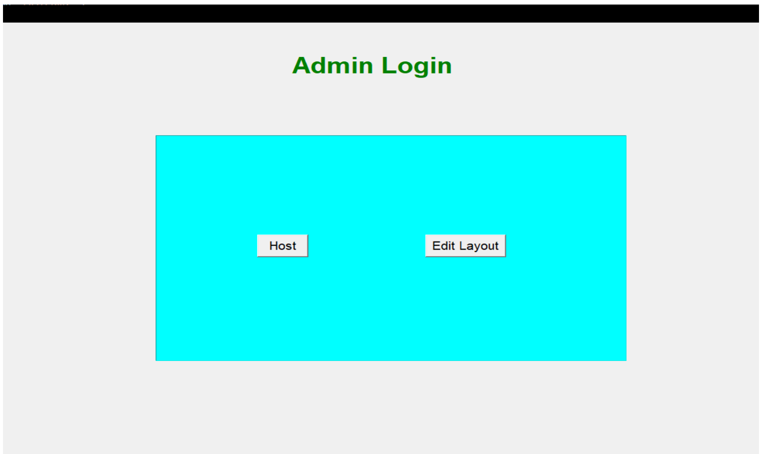



Figure 5.6: Admin Control Panel:

- **Edit Layout:** Admin can edit the layout of the supermarket as per his/her needs. As we can see at left side of the figure, number of rows and columns has to be mentioned so that the grids can be generated. Generate button will generate the layout with the particular rows and columns. Higher the number of cells, better will be the accuracy of the trolley

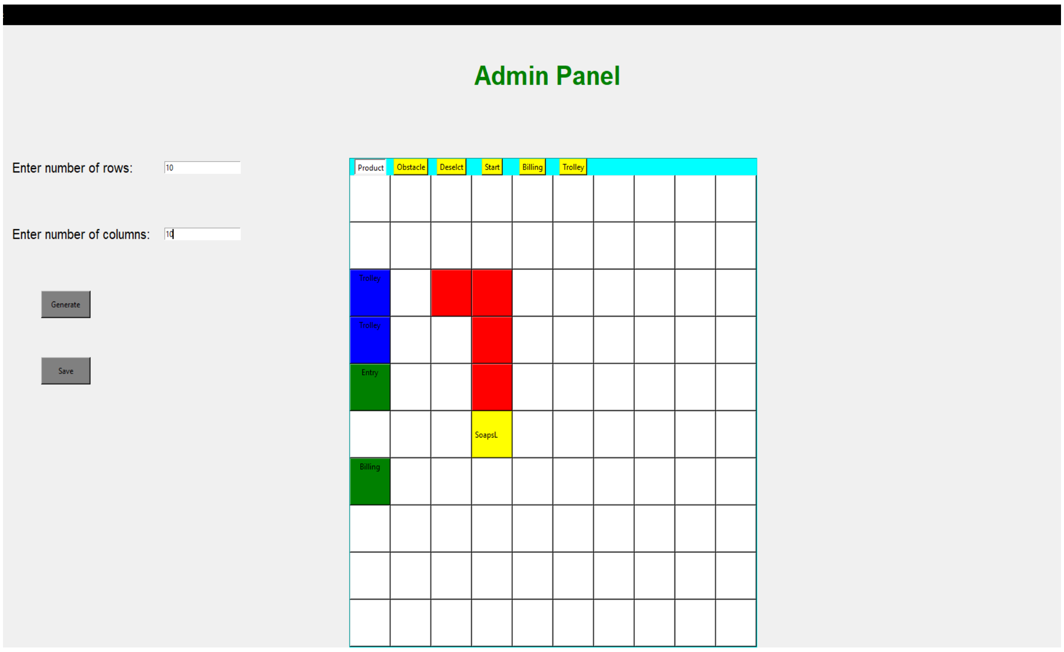



Figure 5.7: Edit Layout

Each option on top of the layout has its own functionality. Admin has to select any one of them and click on particular cell. The Cell which has been clicked acquires the characteristics that has been imparted by the selected option.

1. **Products:** By selecting this option admin is mentioning the position of the product within the supermarket. As soon as user selects any of the cell a pop-up appears asking for name of the product and its position. Position represents the side towards which the product is facing. This is necessary because the trolley must know which side of the rack that it has to stop to get the product. Limitations: Each cell can contain only two products i.e., one to the left and the other to the right.

2. **Obstacles:** This option will highlight the selected cell with red background as shown in the figure. There are the cells where bot cannot entry. It might be an empty rack or unloaded items that may present in the supermarket.

3. **Deselect:** If the admins want to remove the product or any obstacle, he/she can select this option

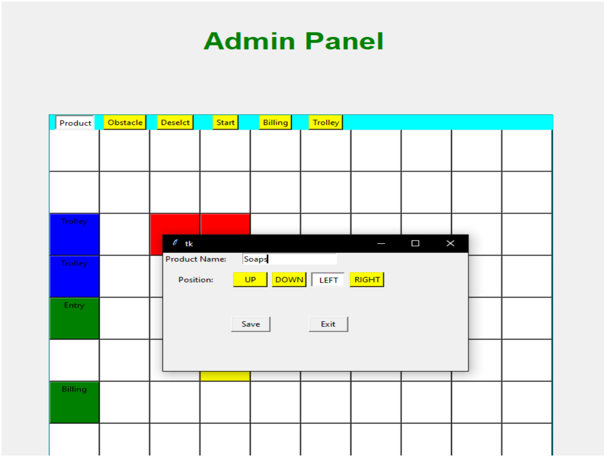



Figure 5.8: Adding Products

4. **Start:** This option is used to determine the entry point. This will be the position from which the movement of the trolley initiates.
   Limitation: Only one cell can be selected as a start point.
5. **Billing:** This option is used to determine the billing section. This position will be the last cell where the entire process terminates.
   Limitation: Only one cell can be selected.
6. **Trolley:** This option is used to represent the trolley section in the supermarket. We can have multiple cells representing this section.

- **Hosting server:** This will display the entire layout to the user and simulation of the movement of trolley can be viewed here. It hosts a TCP/IP connection at a certain port number. User can connect to this port after performing the authorization process. It is to be noted that the server has to be hosted before customer scans the QR code placed on the trolley.

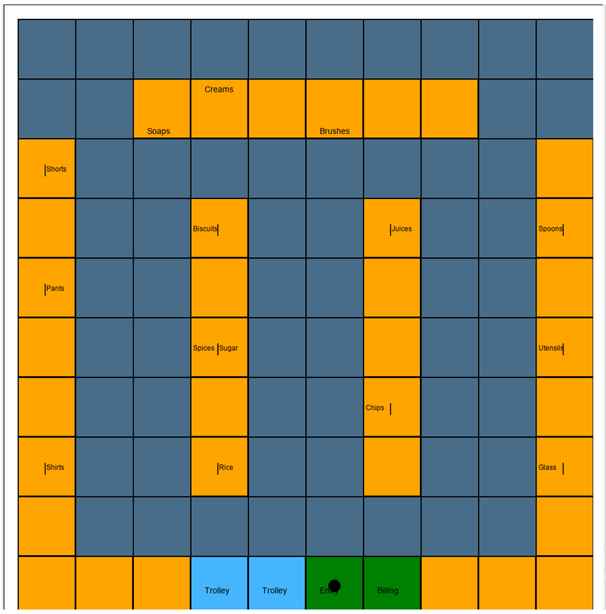



Figure 5.9: Final Layout

### 5.1.3 Algorithm Design

The Algorithm that has been developed is mainly focused on finding the shortest distance between two points in a matrix. Before we go through the entire algorithm there are certain assumptions that are made taking into consideration of the real-time implementation of the project.

**Assumption 1 :** Entire layout of the supermarket is assumed to be fit within a square area. It is divided into certain rows and columns forming a matrix. In other words, imaginary grids will be drawn over the area of the supermarket.

**Assumption 2 :** Movement is restricted only to four directions. We will discuss this part in finding out the valid neighbours of a particular cell in further sections.

**Assumption 3 :** Product and obstacle placements are mentioned perfectly as they are in the supermarket in accordance with the virtual grid system that has been created. This work will be done by the admin team of the supermarket. Every portion of the supermarket can be located with certain co-ordinates.

Terms that are frequently used:

1. **Cell:** Each block in the grid is called as a cell. Each cell consists of unique co-ordinates.
2. **Neighbours:** Each cell has at the most four neighbours. Not every neighbour is considered as valid.
3. **Goal:** Each product that has been selected by the user is called a goal.
4. **Start:** Position where the actuation of the bot starts.
5. **Billing:** Position where the entire process terminates. This position would be a billing counter in the supermarket.
6. **Obstacles:** Place where the bot cannot move to. It will not contain any products.

- **Basic Structure:**
  Basic structure of the algorithm is shown in the figure 5.11 . It accepts the list of goals from the user in the form of list. Each and every possible combination of the goals is taken into consideration and is stored in a two-dimensional matrix. Billing section's position will be added at the end of every combination as it will be the last goal of the user. For example, if there are 5 products to visit, there will be 5 factorial i.e 120 lists that are created with each list containing different possible combinations of 5 products. This approach makes sure that all the goals are covered in a minimal span of the area.

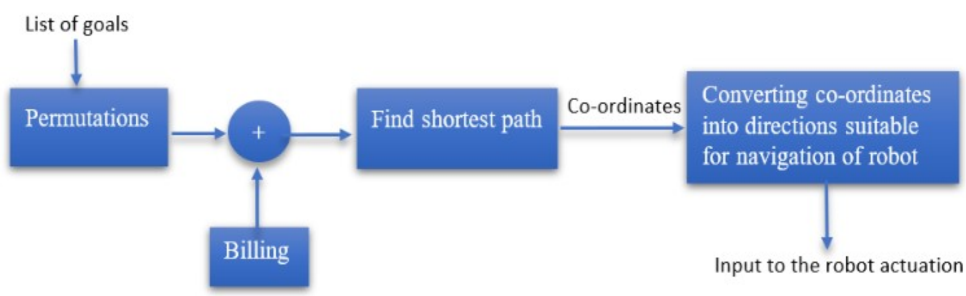



Figure 5.10: Algorithm Block Diagram

Shortest path to all the combinations is calculated and stored accordingly. Among the 120 shortest paths the best one will be chosen and the tracking of the goals goes in that order itself. This ensures that if goal 3 is nearer to goal 1 than goal2, then goal 3 will be covered before goal 2. The order of the input from the user does not matter with the usage of this approach.

After going through finding the shortest path part of the algorithm, we get the path to follow in form of co-ordinates. That needs to be converted to directions such as left, right, up, down to actuate the robot. Once it is taken care of the output is fed into the RPi.GPIO to handle the actions of the robot.

**Finding shortest path:**
This part of the algorithm returns the co-ordinates to all the different goals in the form of a list. This is the core part of the program which runs as many times as that of the number of combinations that exist. It consists of :

1. **Queue:**It holds the list of paths that has been traced or visited.
2. **Visited List:** This will hold the track of the cell that has been already visited. This helps us to prevent the program from going into the deadlock situation.
3. **Obstacle list:** List containing the position of obstacles blocking the movement of the trolley.
4. **Product to position dictionary:** This helps to fetch the position of the product based on its name.

Algorithm that has been proposed is a modified version of Breadth First Search in graph. We keep track of the movement of the robot in available directions with the help of path list and queue. Available directions is decided by the valid neighbour detected.

**How can a valid neighbour be detected ?**

Each cell will have four neighbours, but all of them are not suitable for the movement of the robot. There are certain criteria which determine the validity of the neighbour. The neighbour satisfying following conditions are considered to be valid :

1. Co-ordinates of the neighbour must lie within the perimeter of the entire lay-out.
2. It should not be present in the product list.
3. It should not be present in the obstacles list.
4. It should not be present in the visited list.

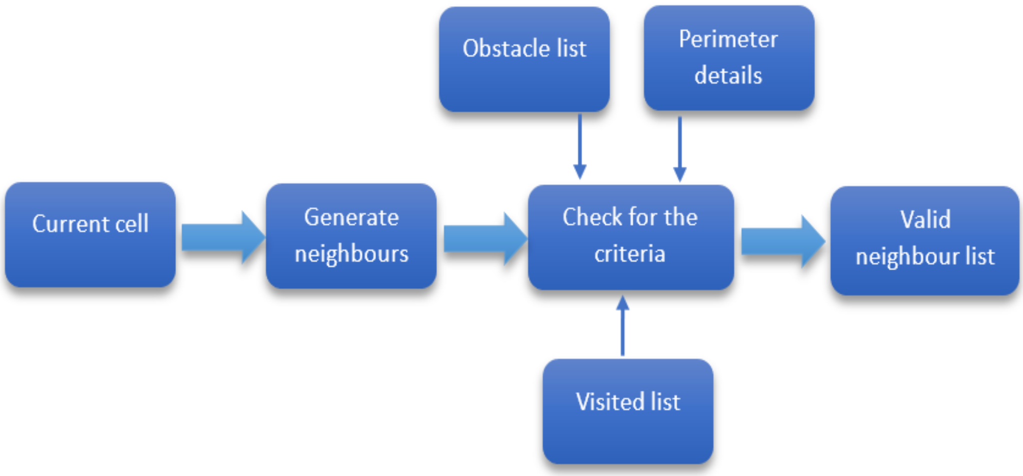



Figure 5.11: Block Diagram Representing The Process of Determining Valid Neighbours

- **Algorithm Explanation:**
  Consider an example shown in the figure 5.12. It shows there are three obstacles. Object has to move from start to end position. Each cell is given a name from A to N in order to track the movement.

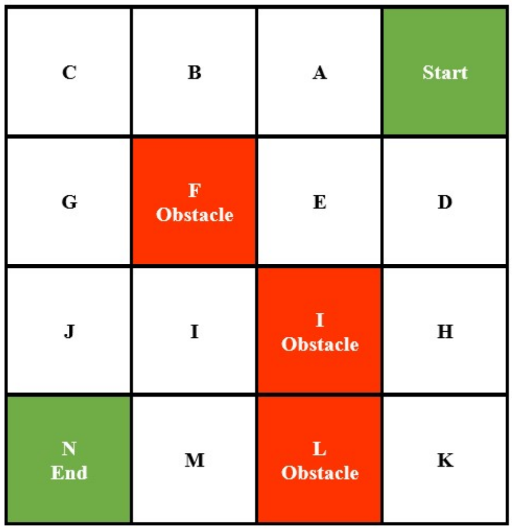
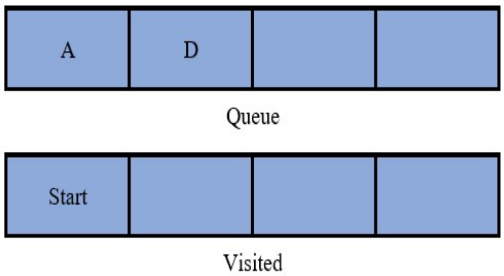



Figure 5.12: Layout

Valid neighbours of start are A and D. Current path would be empty right now. It will be added to the path list and inserted to the queue. Visited list would be updated with the initial position.

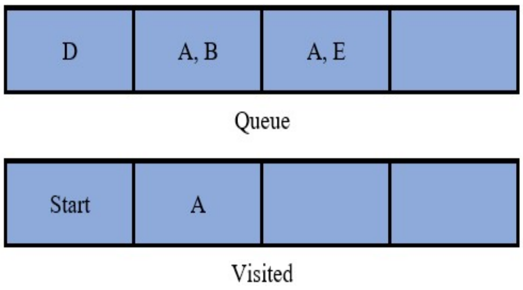
**Current path list: Null**

Pop the list from the queue and make it the current path. Popped list would be A.




**Current Path List: A**

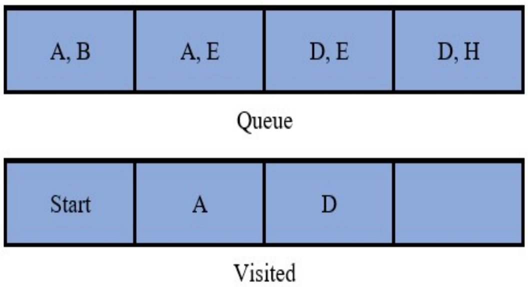
At this point the object is assumed to be at the end of the current path list. A is added to the visited list. Valid neighbours of last added element to the current path list (that would be A) are B and E. They are added to the current path list (A,E and A,E) and pushed into the queue.




Pop the list from the queue and make it the current path. Popped list would be D.

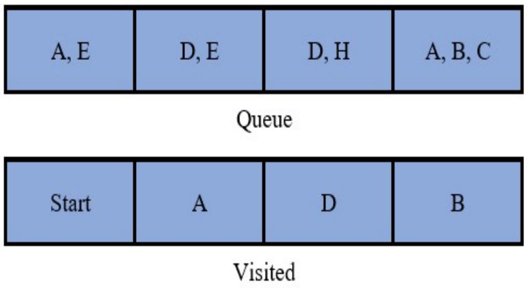
**Current path: D**

At this point the object is assumed to be at D. D is added to the visited list. Valid neighbours of last added element to the current path list (that would be D) are H and E. They are added to the current path list (D,E and D,H) and pushed into the queue. Pop the list from the queue and make it the current path.Popped list would be A,B.




**Current path List: A,B**

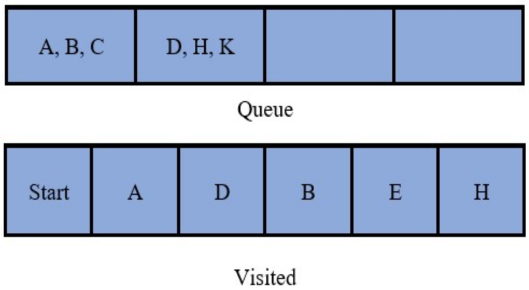
At this point the object is assumed to be at B. B is added to the visited list. Valid neighbour of last added element to the current path list (that would be B) is only C because F is an obstacle and A is already visited, thus not considered as valid neighbours. C is added to the current path list (A,B,C) and pushed to the queue.




Pop the list from the queue and make it the current path.
Popped list would be A,E. Since there are no valid neighbours for E there is no path to continue. Hence continue the process by ignoring it. Next Path list would be D,E. This also faces the same fate as there are no neighbours to E.
Popped list from the queue would be D,H.

**Current path: D,H**

At this point the object is assumed to be at H. H is added to the visited list. Valid neighbour of last added element to the current path list (that would be H) is only K. K is added to the current path list(D,H,K) and pushed to the queue.
Pop the list from the queue and make it the current path.

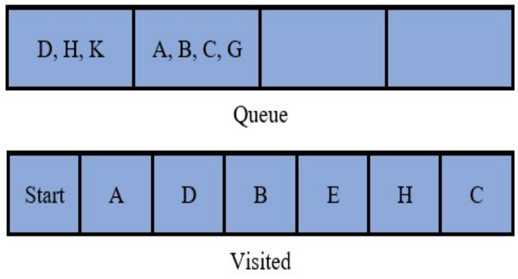



Popped list would be A,B,C

**Current path: A,B,C**

At this point the object is assumed to be at C. C is added to the visited list. Valid neighbour of last added element to the current path list (that would be C) is only G. G is added to the current path list (A,B,C,G) and pushed to the queue.

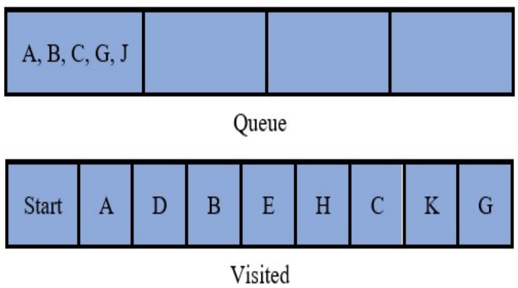
Pop the list from the queue and make it the current path.




Popped list would be D,H,K. Since K has no neighbours next element will be popped and made as current list. Popped list would be A,B,C,G

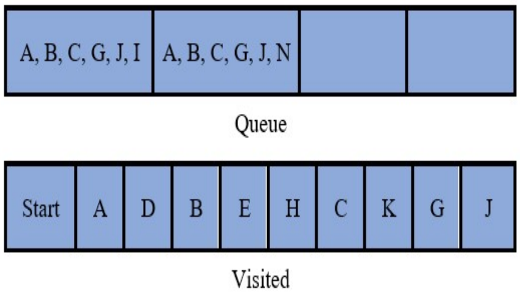
**Current path: A,B,C,G**

At this point the object is assumed to be at G. G is added to the visited list. Valid neighbour of last added element to the current path list (that would be G) is only J. J is added to the current path list (A,B,C,G,J) and pushed to the queue.




**Current path: Start-ABCGJ**

At this point the object is assumed to be at J. J is added to the visited list. Valid neighbours of last added element to the current path list (that would be J) are I and N. Both of them are added to the current path list (A,B,C,G,J,N and A,B,C,G,J,I) and pushed to the queue.




Since the one of the neighbours is equal to the destination that needs to be reached, program terminates and list with the particular neighbour (A,B,C,G,J,N in this example) is taken out and will be the shortest path between start and end.

To sum up, the entire process is depicted in the flow chart figure 5.13. path list will be added with the currently visiting cell and is pushed to the queue. Pop the element from the queue which will be a path list previously pushed. Consider the last element of the list. Find its valid neighbours. Add each of them separately to the path list as those are the ones which becomes current cell in the next run and push them to the queue. Again, pop from queue and repeat the process until the neighbour becomes the end point i.e., billing section specifically.

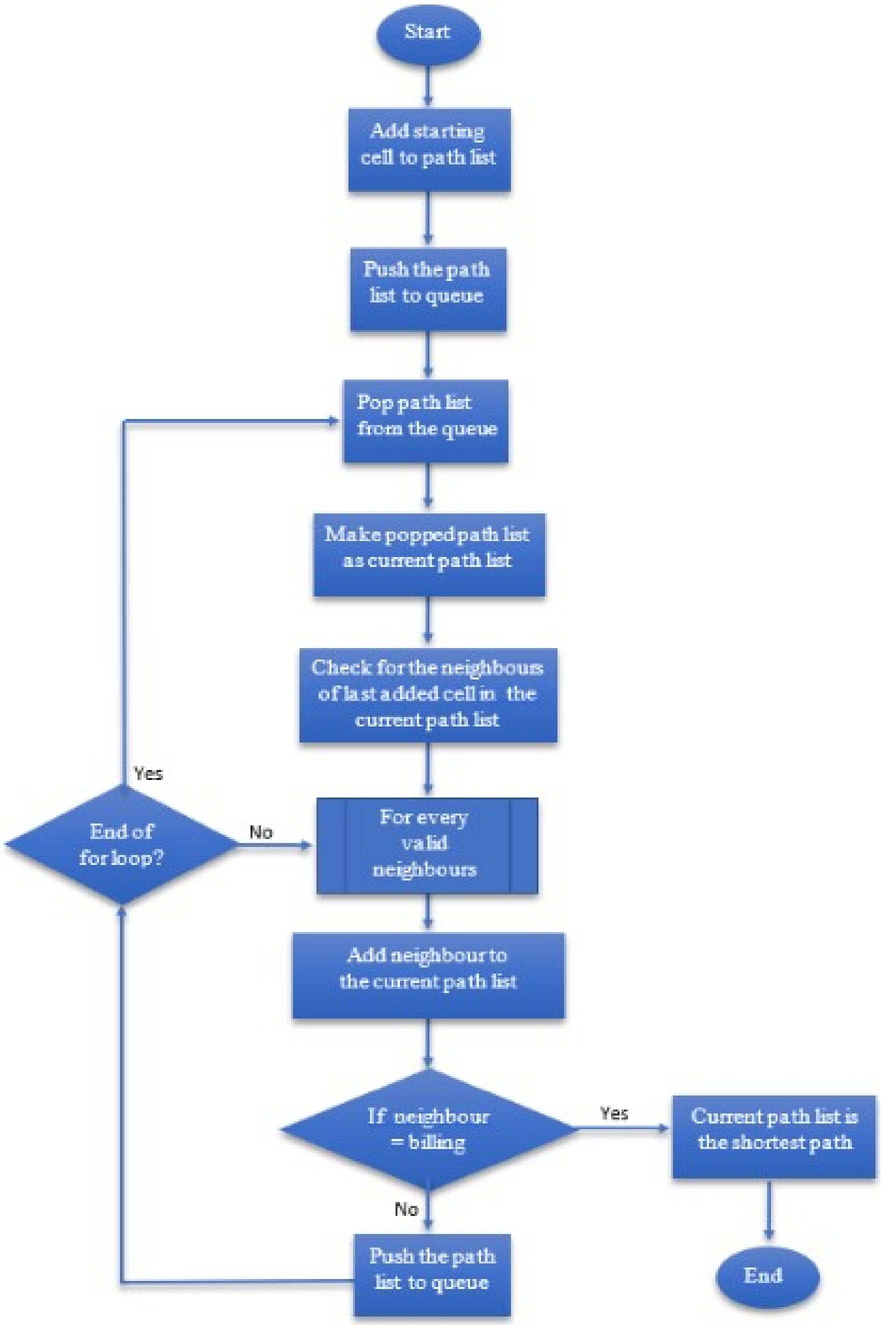



Figure 5.13: Algorithm Flow Chart

## 5.2 Hardware Implementation

Miniaturized structure of a simple supermarket structure has been created for the demo purpose. Each block of tile represents each cell in the matrix. Figure 5.14 gives a proper representation of the supermarket. Red block represents the obstacles through which the movement of trolley is restricted. Products are mentioned in green cones. Yellow tile indicates the start and the blue tile represents the billing counter.

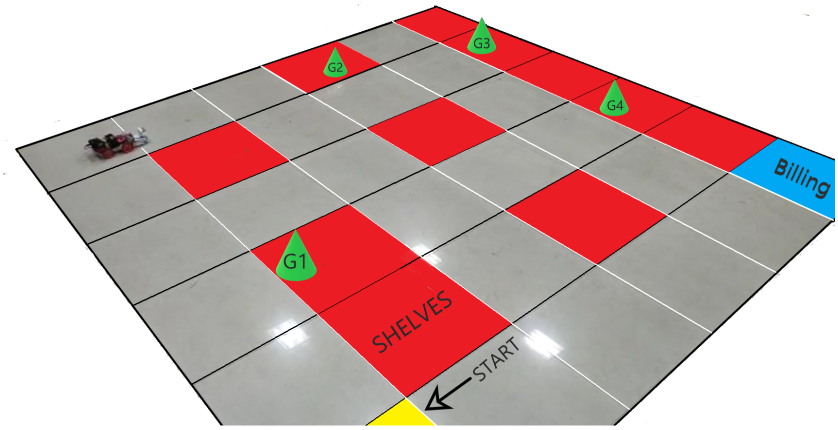



Figure 5.14: Miniaturized Supermarket Layout

All the hardware components mentioned in hardware requirements sections are assembled to form a robot that mimics the action of the trolley. Top view of the assembly is as shown in the figure 5.15. Lead acid battery Is placed in the middle to balance the weight. It weighs around 700g. Power bank is placed at the back end. Raspberry Pi will be facing front. Type C cable runs from the power bank to the Raspberry Pi input port. Motor driver is placed in between battery and raspberry pi. Wheels are attached to the axis with permanent glue to have high precision in the movement. Gyroscope is placed in between the chassis so that it is not disturbed by any other component.

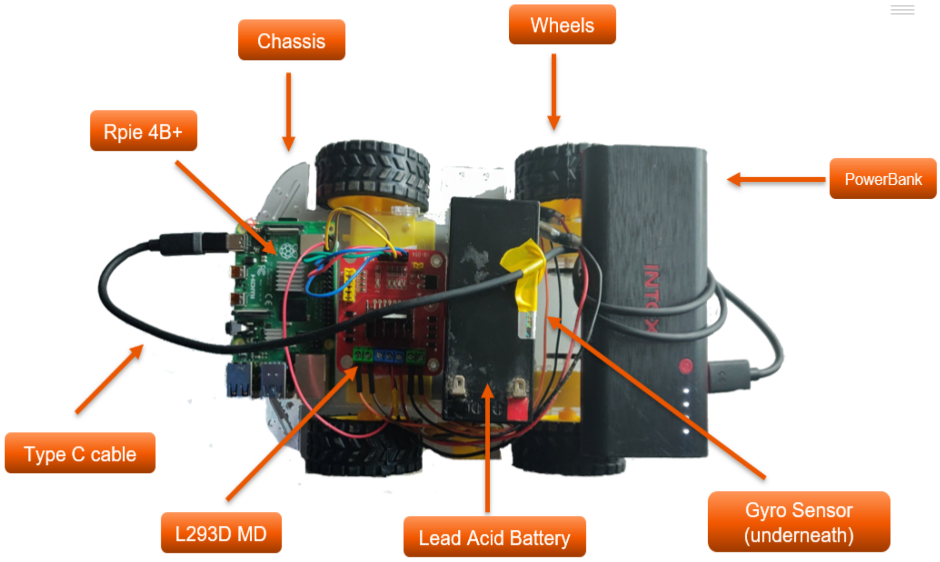



Figure 5.15: Top View of the Assembly

Figure 5.16 depicts the front view of the robot. IR Track is used for precision in movement of the robot. IR array is used to detect the track. This track will be imprinted on the floor of the entire supermarket. As we can see there are 8 IR pairs with calibration ports for each one of them.

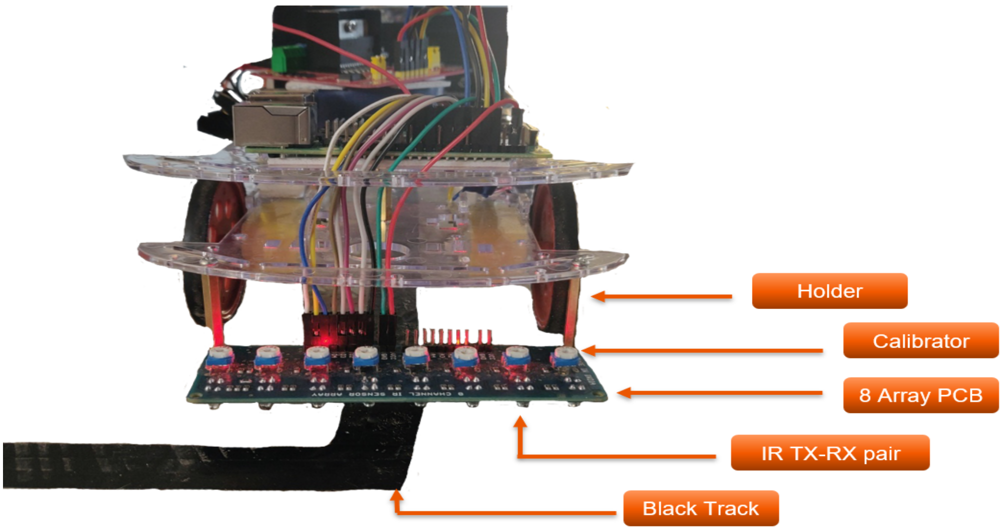



Figure 5.16: Front View of the Assembly

Lighting affects performance of the IR array. Therefore, we have designed an IR shield as shown in figure 5.17 which prevents the influence of external lighting. A strip of LED is placed inside the shield. This ensures the constant lighting for the IR array. Proper readings can be taken from this arrangement. The LED strip is driven by a regular 9V battery.

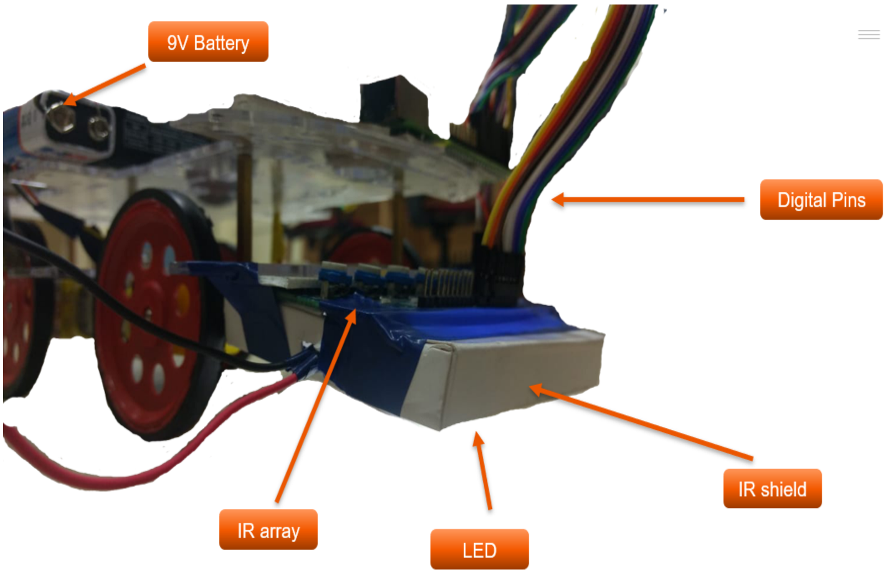



Figure 5.17: IR Shield

## 5.3 System Integration

We have discussed how individual components are implemented in the previous sections. Now let us see how each and every component are integrated to have a complete working system. Figure 5.18 shows the schematic representation of the entire system.

Mobile application has been developed on java platform and has TCP/IP connection to communicate with the server. Mobile application only communicates with server and has nothing to do with the actuation of the trolley. Mobile application can communicate with server only if the device is authorized properly. Mobile application request to the server for the establishment of TCP/IP connection and server responds to the request. Hence there will be two-way communication between the application and server. After the establishment of the connection, mobile application sends control messages as entered by the user.

Once the server gets the required commands, it will be transferred to the core algorithm. Algorithm with the help of database processes the data and calculates the required actions and this fed back to the server. The server transfers these results to Raspberry Pi which is the main driving unit of the trolley. Note that, most of the processing part is done within the server itself. Hence it is a two-tier client-server architecture.

Raspberry Pi requires a power source with minimum specifications of 5V, 2A. With the help of PID-controller Raspberry Pi controls the speed of the gear motors in turn controlling the movement of the robot. IR array gives feedback about the position and gyro sensor gives feedback about the angle. Ultrasonic provides feedback of any obstacle in the direction of movement of the trolley. With these three feedbacks, PID controller corrects the speed of the motors thus corrects the motion of the robot.

Gear motors require different power source with minimum specification of 12V and 1A. Therefore, there is a need of two power sourced on the system as Raspberry Pi and gear motors require a different power inputs. We can also add power converters with only single power source but that will increase the cost of the implementation.

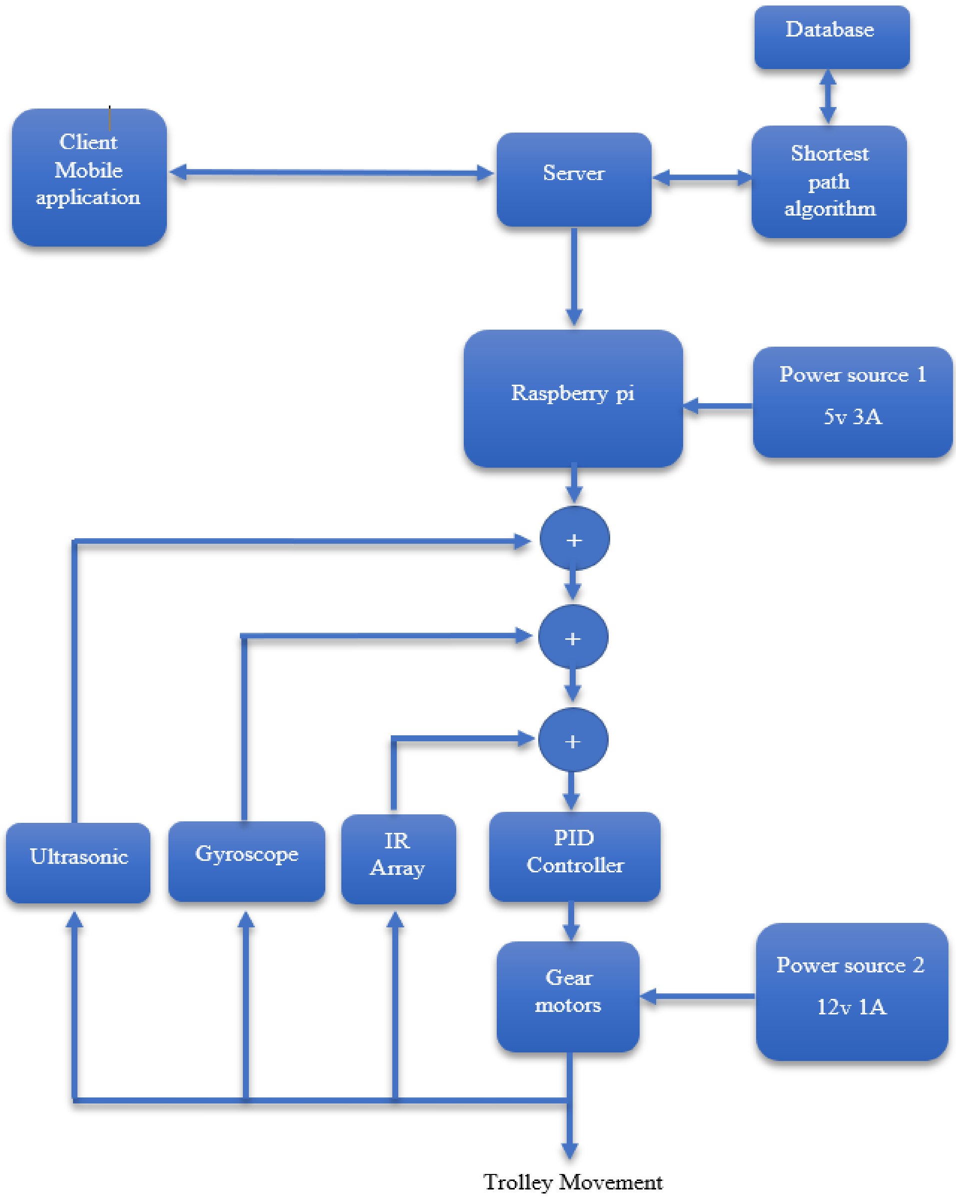



Figure 5.18: Schematic Diagram of System Integration

## 5.4 Simulation Results

Simulation is performed on the python platform using turtle graphics. Turtle has three attributes: location, orientation (or direction), and pen. The pen, too, has attributes: colour, width, and on/off state. The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90°" . The pen carried by the turtle can also be controlled, by enabling it, setting its colour, or setting its width.

A full turtle graphics system requires control flow, procedures, and recursion: many turtle drawing programs fall short. From these building blocks one can build more complex shapes like squares, triangles, circles and other composite figures. The idea of turtle graphics, for example is useful in a Lindenmayer system for generating fractals. Turtle geometry is also sometimes used in graphics environments as an alternative to a strictly coordinate-addressed graphics system

We have mimicked the action of the trolley in the supermarket with the help of turtle graphics. Circle represents the trolley and the path traced by the trolley is tracked with a line. Impression of the circle is left on the place where the trolley stops (in front of the desired products). Each and every option that has been provided is depicted in the simulation.

There are different scenarios that a user may encounter while using the system. Major scenarios are tested and simulation results are obtained accordingly. Assume that the 5 products shown in figure 5.19 are selected and tested for different scenarios.

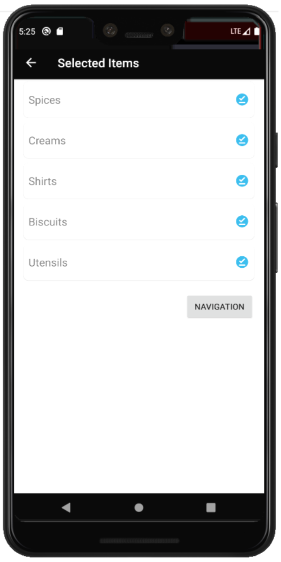



Figure 5.19: Products Selected for Simulation

1. **Scenario 1:** When user fails to scan the proper QR code.
   When user scans the QR code present on a particular trolley, it gets authenticated by the server. If he/she uses this on any other Q R code which is not relevant or QR code that has been disabled due to improper functionality of the trolley, then user will be notified with a not authorized pop-up as shown in figure 5.20.

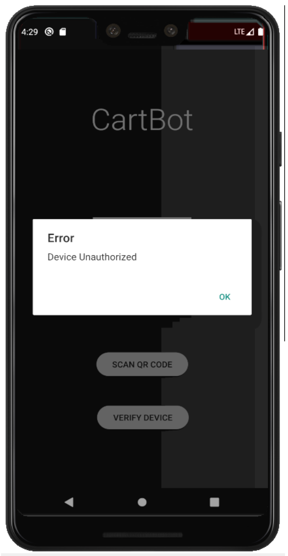



Figure 5.20: Identifying an Un-authorized Device

2. **Scenario 2:** Ideal Working Conditions

   This is the scenario where everything goes right and user collects all the products required by the user. At the end of the process he/she will be automatically taken to the billing section if no products are left to purchase. Note that he/she can also add items in the middle of the process if required.

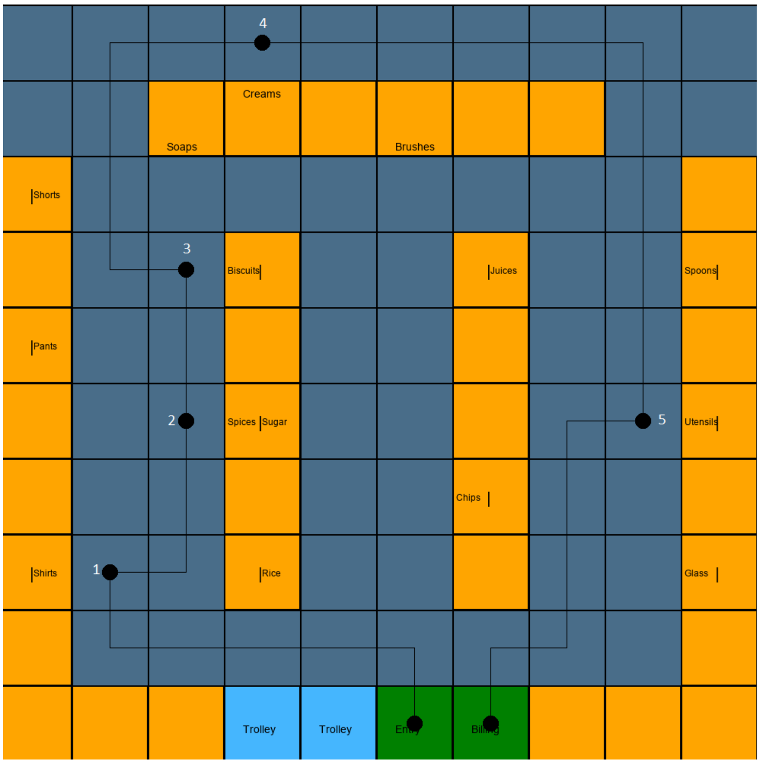



Figure 5.21: Simulation Output for Scenario 2

3. **Scenario 3:** Emergency situation

   This scenario explains the functionality of the billing option. For example, assume that a user selects 5 products and starts shopping. After collecting 4 out of 5 products, user might want to finish the shopping or in the middle of shopping if any emergency situation appears for the user, they can press the billing option provided in the application. This will take the user directly to the billing section from their current position, ignoring the rest of the products.

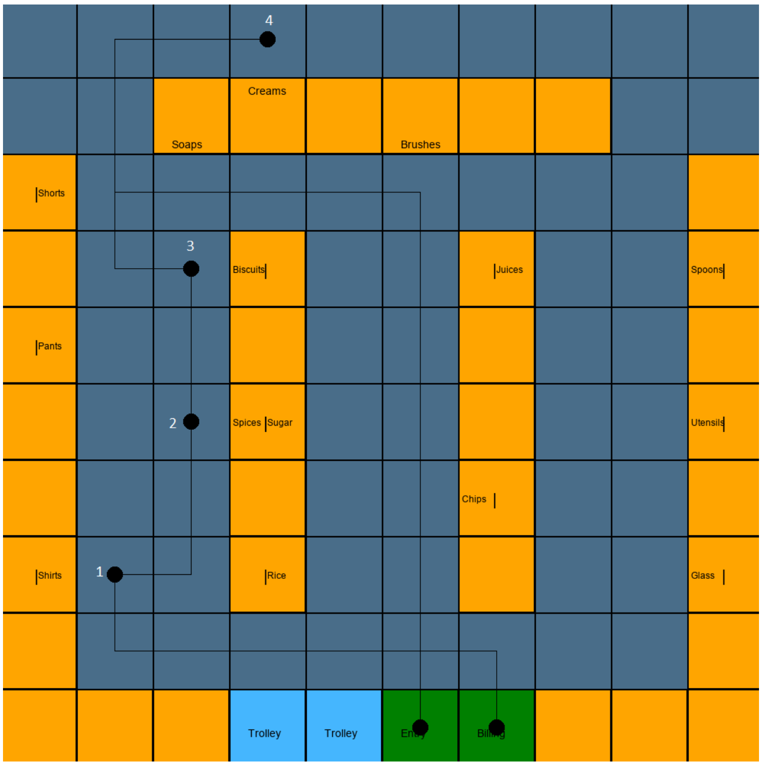



Figure 5.22: Simulation Output for Scenario 3

4. **Scenario 4:** Aborting the system
   In case of hardware malfunction or if user wishes to take the trolley on their own, then he/she can press the Abort button provided. The entire system will be shut down. TCP/IP connection will be terminated and the user will be shown a pop-up message as shown in figure 5.23 and taken to the home screen. Trolley acts as a regular trolley which we see in the conventional supermarkets.

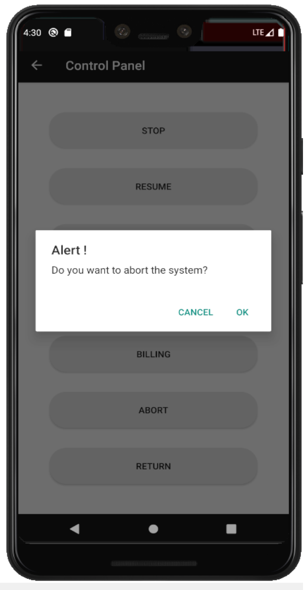



Figure 5.23: Alert Screen for Abort

## 5.5 Component Testing:

**Gyroscope:**

Gyroscope readings vary from 0 to 16384 for complete 360°rotation. This value can be obtained only during the constant motion of the robot in the Z-axis (lateral movement). This value is converted into degrees as discussed in the design part. The resolution of the Gyroscope would be 360/16384 degrees.

**Infrared sensor:**

Infrared sensor is sensitive to lighting effects. The output of the device will be high when placed on black surface and low on plain white surface. It gives effective readings within 3 cm range from the surface. It can be calibrated to give readings at different height (3cm being ideal). Hence black track is used against white tiles in the project.

**Raspberry Pi power source:**

The power source used for the Raspberry Pi 4 B is a 5V-2A power bank. The optimal power source for the Raspberry Pi 4B is 5V at 3A from a Type C cable. The primary power source used was not suitable enough to provide the required amperage for the microprocessor.

**Ultrasonic distance:**

The distancing for the ultrasonic device was tested out at various minimum distancing lengths in order to achieve the best possible "Safe distance" to avoid collision. The robot is of a length of 20 cm and the minimum safe distance considering the speed of 500RPM was measured to be 15 cm.

**Power output RPM:**
The power source used for the drivetrain of the robot is a 12V 1.33A Lead Acid Battery. The DC motors are driven by the L298N motor driver which provides the optimum voltage regulation for the DC motor with great RPM control. The set RPM for the motors are at 20-25% PWM capacity for optimum speed and vehicular control of the robot.

**Shortest path calculation time:** This factor completely depends on the type of server that has been used. Higher the server clock speed, faster will be the path calculation time. This factor plays a major role as a customer cannot wait for a long time to initiate the shopping. Since we are calculating all possible combinations of the selected items, increase in number of items increases the calculation time. Table 5.1 gives the details about the calculation time with respect to the number of products selected.

The reading recorded in table 5.1 was the result of testing performed on the server with 2.7 GHz (3.1 GHz maximum) base clock speed and 8 GB of RAM. With increase in the clock speed, calculation time decreases.

Table 5.1: Number of products v/s Execution time

| **Number of Products** | **Number of Permutations** | **Execution time (in sec)** |
|---|---|---|
| 1 | 1 | 0.0049860477447509766 |
| 2 | 2 | 0.032912492752075195 |
| 3 | 6 | 0.1466076374053955 |
| 4 | 24 | 0.24334955215454102 |
| 5 | 120 | 1.7852246761322021 |
| 6 | 720 | 11.096779823303223 |
| 7 | 5040 | 77.61944890022278 |

From testing of individual components to integrating all of them into a single system, many issues have been faced. Major issues and solutions that has been adopted for them are listed in table 5.2

Table 5.2: Difficulties and Solutions Table

| Sl.No | Difficulties faced | Solution used to overcome |
|---|---|---|
| 1 | Gyroscope readings were not measured in degrees and needed to be converted into degrees. | Relationship between gyroscope readings and degree values was found by constant testing. Resolution of Gyroscope was found to be 360/16384 |
| 2 | IR calibration was also an issue because of the variable lighting condition from all the directions. Even shadows were affecting the readings | An IR shield was created to protect it from external light source. LED strip was provided which shed light only below the IR sensors. |
| 3 | Raspberry-Pi and motor driver required different power input and cannot be drawn from a single power source | We tried with implementing power converter circuits which did not go well. Therefore we used two different power sources with common ground. |
| 4 | Even with the new setup, achieving precise actuation of the robot is difficult. Many factors such as friction, obstacles, overshooting affects the motion of the robot. | Both IR and Gyroscope readings were used as the feedback to the PID controller. IR is used to get the position of the robot with respect to the black tract and gyro helped to correct the angle of deviation |

| Sl.No | Difficulties faced | Solution used to overcome |
|---|---|---|
| 5 | Gyroscope values changed at a rapid rate hence readings should be taken all the time. But the actuation of the robot needs delay because of the lower rpm that we are using. Thus the readings of gyroscope were not accurate | Multi-threading was used to take the readings of the gyroscope. |
| 6 | Skidding of the wheel from the axle | Permanent glue was used to stick the wheels to the axle. |
| 7 | Increasing the number of products, increases the shortest path calculation time | Usage of system with a higher clock speed |

# Chapter 6

# CONCLUSION

"All good things must come to an end" Chapter 6 marks the cessation of this intricate yet aesthetic documentation. Read on to see how the project attains resolution at the curtain's call.

## 6.1 Project Summary

After successfully procuring all the products needed and the customer clicks the “Billing” button for the trolley to reach the billing section. This action can be considered as the cessation of the product cycle. Since the inception of this novel idea to the implementation of it into a product, the entire process has been highly involving and educational. Different pillars of this project such as the GUI, Server, and the hardware have evolved and been improvised at every step of the way. The simulation serves as a testimony to all the possible shopping scenarios. The hardware demonstration vindicates how once can shop without any monitoring of the trolley. The application hence provides control and interface to the user in the most efficient way. All these factors put together culminate to demonstrate the optimum working of the project in its ecosystem.

The product has been subjected to various scenarios and test cases to check for its robustness and versatility. The algorithm has been improved and made effective to deal with a countable infinite number of goals. The robot which depicts the trolley has been scrutinized to various physical parameters and has been improvised for stellar performance. All in all the project has achieved its objectives and fulfilled the mission statement.

## 6.2 Applications

When one develops a project such as ours, it would be in-efficacious if it were not allowed to spread its wings. The immaculate structure behind this elegant algorithm, crisp and light user interface of the application developed, state of the art robotics hardware, and the harmony in which these elements work together to produce an undeniably clever result is simply amazing. The project in itself is an application of the skeletal structure it's married to. The framework hence built can be applied to any suitable environment which requires autonomy in tasks and around dynamic user input. Shopping malls serve as a good example of an environment this can be applied to. Consider the following environments to which this framework can be inculcated:

1. **Supermarket:** Increase in the area of supermarket will result in roaming around different sections searching for the components. This automation in transport facility helps the user to carry the goods all over the place without putting extra energy. Also, user can carry any amount of product irrespective of the total weight.

2. **Factories and Warehouse:** Factories are usually a huge real estate. They consist of various wings and departments which is a classic example of segmentation. Segmentation means different products would be located in various physical locations, now a guidance and navigation system as such would be if immense help to identify and procure the desired products. This framework can be successfully applied to such areas as the algorithm is can be scaled to a countable infinite level.

3. **Industries:** Automobile industries such as vehicle manufacturing industries requires transportation of parts from manufacturing unit to the assembling unit.

4. **Agriculture:** It has a very wide range of application in agriculture field. Ploughing, seed implantation, irrigation etc. can be atomized.

5. **Libraries:** Libraries in a way aren't that different when it comes to roaming around as compared to a shopping mall. You go around looking for products in aisles you want and depending they're stacked according to the category and content. For a robot leading you to the goal, it doesn't matter whether you're going to get yogurt or "The Theory of Everything", a goal is a goal regardless of what it is. Hence this system can be scaled and applied to a library in which it can exercise its duties invariably.

6. **Guiding robot:** When we pan out and consider this project in a much broader sense, it can be viewed as an assistant to a human in need of directions. In places like college campuses or huge tech parks, people are often in turmoil when it comes to directions or locations. Personal assistants such as these can mitigate the dilemma and guide the user to the required destination. Of course, one would argue that they could use Google Maps to find the direction. But how often do you come across a robot which would not only reach you to your destination accurately to a few centimeters but would also be your tour guide? Now that's the kind of applications we're hoping to achieve in the future works of this project.

## 6.3 Advantages

The advantages of this project over the other systems is one hundred percent, because there is no other system like this. The advantages of this ecosystem created can be condensed into the following points:

1. High efficiency in the shopping path using the in-house algorithm developed results in optimum shopping time.

2. Easy GUI for user understanding and highly helpful for elderly customers in order to find the products.

3. Flexible and scalable mapping environment which can be applied to many locations of various sizes and shapes.

4. Ease of access and secure through private network hosted and QR authentication mechanisms.

5. State of the art actuation unit ensures optimum maneuverability and computation speed for faster response time.

6. Centralized processing takes the payload off the actuation module and ensures modularity in the ecosystem of the product.

7. Application created as a GUI enables the user to interact with the robot rather than just following.

8. Billing feature reduces the entire 3rd phase of a shopping customary. It can be further worked upon to automate the billing as the product is dropped into the cart. Once all the products are dropped into the cart, the cart tally's the bill and you'll be automatically charged over the internet. The trolley can meet you downstairs in the parking at your car for you to unload the items.

## 6.4 Limitations

When the project is implemented in a real-time environment certain difficulties may appear. Considering all the scenarios, limitations of the system developed are listed below:

1. **Overloaded servers:** When there are frequent simultaneous client requests, servers get severely overloaded, forming traffic congestion. In a P2P network adding more nodes will increase its bandwidth since it's calculated as the sum of bandwidths of each node in the network.

2. **Impact of centralized architecture:** Since the architecture is centralized, if a critical server fails, client requests are not accomplished. Therefore client/server lacks robustness of a good P2P network (resources are distributed among many nodes).

3. **Overcrowding:** There are certain parameters that we can control and others we just can't. Overcrowding is one drawback that's encountered when the environment is congested with people and the robot cannot get a definitive obstacle-free path to maneuver. If the number of people in a block increase, the robot faces one too many obstacles to move around freely.

4. **Static Map:** The entire product works the best for static environments(static in terms of product placements). Conventional supermarkets always have a systemic and by-the-book way of arranging items. If there is a displacement of any product from it's the original position, the robot wouldn't know it's new location unless updated by the administrator. The provision for updating the map is provided as an administrator-level privilege to counter these challenges.

## 6.5 Future Work

The architecture of the project hence developed has a multitude of applications. The basic idea behind the project is to provide a guidance system for the customer in an enclosed environment. Since this kind of architecture acknowledges the various necessities and parameters involved in the environment, the deployment can account to almost all situations thrown at it.

The future scopes of this project can be broken down into various compartments involving a broader spectrum and a narrow spectrum of elements included in its development. The plethora of technologies available at our disposal in the 21$^{st}$ century is overwhelming but yet subtly elusive during integration. Factors such as cost, developmental procedures and viability come into the picture when planning on upgrading the product.

The quintessential factor for a product/project to achieve sustainability is the ability to be robust and versatile in nature. In this particular venture we can categorize the upgradation into two unequivocal and distinct segments. Hardware and software. The hardware and software constructs of the product are relationships which need to be in perfect harmony and tranquillity to perform optimally. Surely, software updates are much easier to be methodized into the product's life cycle but the lifespan of the product's hardware is questionable with the current resources put into use. Hence one must focus on a golden ratio of upgrades possible in-order to maintain the sanity of the administrator/customer.

Coming to the physical and tangible attributes of the hardware compartment of the project. The following domains can be improved in the near future for enhanced performance.

1. **Battery:**
   Battery that has been used by motor driver is lead acid battery which are bulkier and heavier in nature. Lithium Polymer batteries can be used in place of lead acid batteries.

2. **Power train:**
   An efficient power converter can be designed so that single power source can be used to fuel both Raspberry Pi and motor driver.

3. **Sensors:**
   All the required sensors can be embedded on to a single PCB module and placed along with Raspberry Pi. This decreases the area consumed by the sensors. High performance sensors can be used that are built only for specific applications.

4. **On-Board processor:**
   As we mentioned earlier clock speed of any processor matters the most when it comes to speed of execution. Better processors can be used for the project purpose so that the efficiency can be increased.

5. **Connectivity:**
   High level networking architecture can be built within the supermarket arena ensuring better and reliable connectivity. Also, better firewalls can be implemented to increase the security factor of the system.

With regards to the software architecture of the project the following amends can be made to increase the efficiency and speed of execution. Also, a refinement of the software inevitably leads to better user experience as well.

1. **Refined Algorithm:**
   Algorithm that has been developed works on all the test cases and scenarios of the supermarket. The only flaw is the time of execution. Further this section can be addressed and time of execution can be reduced.
2. **Better UI:**
   Design of mobile application as well as interface created to handle the server can be made better with addition of graphics to the layout.
3. **Cloud support and User database for mobile application:**
   The database can be made online for ease of access and online processing systems can also be used for customized specifications.Online platforms like AWS,Azure or GCP are such platforms which can come in handy.
4. **Troubleshooting Abilities for mobile application:**
   Error reporting and error logs can be sent to the developer team for addressing crash reports.
5. **Updates and bug fixes need to be provided periodically:**
   Regular updates and bug fixes can be pushed to the application and server, in order to maintain better consumer life cycle management.

# References

[1] "A Pilot Study: Development of Home Automation System via Raspberry Pi" by M. Mahadi Abdul Jamil and M. Shukri Ahmad, 2nd International Conference on Biomedical Engineering (ICoBE), 2015.

[2] “A Self Stabilizing Platform" by Vladimir Popelka, Proceedings of the 2014 15th International Carpathian Control Conference (ICCC), 2014.

[3] “Design of Two-Wheel Balance Car Based on STM3" by Guoping You and Wanghui Zeng, 9th International Conference on Information Technology.in Medicine and Education (ITME), 2018.

[4] “A New Approach for Line Following Robot Using Radius of Path Curvature and Differential Drive Kinematics" by Jitendra Singh and Prashant Singh Chouhan, 6th International Conference on Computer Applications In Electrical Engineering-Recent Advances (CERA), 2017.

[5] “Robotics, Vision and Control" by Peter Corke, textbook, 2nd edition, Springer publications.

[6] "Study of Kalman Filtering Algorithm and Fuzzy PID Controller for the Coriolis Acceleration Tester” by ZHANGJian, GUOLei, WEI Shimin, SONGYuan and ZHANGYin, 35th Chinese Control Conference (CCC), 2016.

[7] “Moving Object Detection and Tracking in Indoor Environment" by Shucai Wang, Liying Su, Kim Wang and Yueqing Yu, Proceedings of 2011 International Conference on Electronic  Mechanical Engineering and Information Technology, 2011.

[8] “Tracking of Moving Objects With Regeneration of Object Feature Points” by Igor I. Lychko, Alexander N Alfimtsev and Sergey A, Global Smart Industry Conference (GloSIC), 2018.

[9] “Edge based Moving Object Tracking Algorithm for an Embedded System" by Kai Xiang Yang, Ming Hwa Sheu A, IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2016.

[10] "Dual Model Learning Combined With Multiple Feature Selection for Accurate Visual Tracking" by Jianming Zhang,Xiaokang Jin,Juan Sun,Jin Wang and Keqin Li, IEEE access journal article, 2019.

[11] "Simulation Of A Human Following Robot With Object Avoidance Function" by Nattawat Pinrath and Nobuto Matsuhira, 12th South East Asian Technical University Consortium (SEATUC) 2018.

[12] "An Adaptive 2D Tracking Approach for Person Following Robot" by Shenlu Jiang, Meng Hang, Ling Li and Toiyong Kuc

[13] "Development of the Human Following Robot Control System Using HD Webcam" by Jutarat Jommuangbut and Kiattisak Sritrakulchai, International Symposium on Computer Science and Intelligent Controls (ISCSIC), 2017.

[14] “Review of Geometric Distortion Compensation in Fish-Eye Cameras” by Ciaran Hughes, Martin Glavin, Edward Jones and Patrick Denny, IET Irish Signals and Systems Conference (ISSC), 2008.

[15] “An Approach for Fully Automating Perspective Images Based On Symmetry and Line Intersection" by Robin Tommy and Mohan S, International Conference on Image Information Processing, 2011.

[16] Tkinter documentation: https://docs.python.org/3/library/tk.html

[17] Turtle documentation: https://docs.python.org/3.3/library/turtle.html?highlight=turtle

[18] BALYO automatic forklift : https://www.balyo.com/products/counterbalanced-stackers yale-mc15

[19] Freight automated transport system : https://fetchrobotics.com/products-technology/virtualconveyor/freight-robots/

[20] Bonirob: https://www.popsci.com/meet-bonirob-plant-breeding-weed-smashing -robot/