# Mini Project 2B

# Report

# On

**Smart Medicine Dispenser**

|  |  |  |
| --- | --- | --- |
|  | By |  |

**Shalom Pakhare / 8822**

**Sadiya Shaikh / 9021**

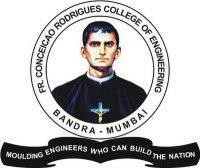
**Abhishek Athani / 9067**

**Arpita Kar / 9073**

**Semester 6/ TE ECS**

Supervisor

**Prof. Narayanan Kallingal**



**Department of Electronics and Computer Science**

**Fr Conceicao Rodrigues College of Engineering**

### Bandra, Mumbai - 400 050

**University of Mumbai**

**(AY 2021-22)**

# CERTIFICATE

This is to certify that the Mini Project entitled **“Smart Medicine Dispenser”** is a bonafide work of **Shalom Pakhare (8822), Sadiya Shaikh (9021), Abhishek Athani (9067) and Arpita Kar (9073)** submitted to the University of Mumbai in partial fulfillment of the requirements for Mini Project 2B, Semester VI, Third Year **Electronics and Computer Science.**

#### (Prof. Narayanan Kallingal)

Supervisor

#### (Prof. ) (Prof. )

Head of Department Principa

# Mini Project Approval

This Mini Project entitled **“Smart Medicine Dispenser”** by Shalom Pakhare (8822), Sadiya Shaikh (9021), Abhishek Athani (9067) and Arpita Kar (9073) is approved for the requirements for Mini Project 2B Third Year **Electronics and Computer Science.**

**Examiners**

**1………………………………………**

(Internal Examiner Name & Sign)

#### 2…………………………………………

(External Examiner name & Sign)

Date: Place:

# Contents

[Abstract ii](#_TOC_250000)

Acknowledgments iii

1. Introduction 1
   1. Introduction
   2. Motivation
   3. Problem Statement & Objectives

#### Literature Review 4

1. **Proposed System**
   1. Introduction
   2. System Block diagram
   3. Theory of operations and Process Design
   4. Details of Hardware & Software

**3.4** Experimental setup

#### Results and Discussion 10

* 1. Description of Results
  2. Application area
  3. Conclusion and Future Work

#### References 11

**Program Code**

# Abstract

Sometimes it is tough to buy medications from a medical store during emergency. For example, if a person experiences a sudden headache in middle of nowhere and he does not have medicines then it might be tedious for him to find and buy medications from a medical store. To solve this problem, we will create a vending machine which will provide general medicines, ointments, sprays and balms. It can be installed at train stations, subways, metro, malls, bus stops, tram stops, and airports.

This machine will be very easy to use as it will have touch screen lcd for user to give the required input easily without any confusion and also RFID payment system for easy payment method without any problem.

# Acknowledgement

In completing this project report on project titled Smart Medicine Dispenser, we had to take the help and guideline of a few respected people, who we deserve to thank. The completion of this project report gives us much Pleasure.

We would like to show my gratitude to Prof. Narayanan Kallingal for giving us a good guideline for project throughout numerous consultations. I would also like to expand my deepest gratitude to all those who have directly and indirectly guided us in writing this project report.

Here we thank all the people for their help directly and indirectly towards the completion of the project successfully.

## Chapter 1 Introduction

**Introduction**

We have observed that most of the time we don’t have medicines when we need it the most. Also, all chemist shops are not day and night, so if a person needs urgent medicine at midnight then they have to wait till morning.

To solve this problem, we have come up with a solution called as ‘Smart Medicine Dispenser” which is an automatic vending machine which will have all urgent required medicines and supplies which a person can get just by scanning their RFID card and giving their required input on the screen touch LCD display.

**Motivation**

The main motive to create this project is to help people to get required medicines or supplies such a pain sprays, balm, etc from this vending machine which will be located near their areas or on airports, railways stations, bus stops, etc.

**Objectives & Problem Statement**

**Objectives:**

* To create a system were general medicines and first aids are available at public places for 24x7.
* Easy payment using RFID card.
* Reduces the possibility of human errors.

**Problem Statement:**

Problem like database and backend integration, mechanical error for motor and

Smart Medicine dispenser is a vending machine which will provide general medicines and first aid.

## Chapter 2 Literature Review

* After referring to “Vending machine operating mechanism.” by Hoffman Arlan J. it can be installed at public and transit places like train stations, subways, metro, malls, bus stops airports, restaurants and lodges at highways.
* After referring to “A comparison of Arduino, Raspberry Pi and ESP8266 Boards” by Samson Otieno Ooko it is decided to use raspberry pi over Arduino as it is compatible and have more pins as compared to Arduino. As it also gives a use of ethernet availability.
* It is decided to use stepper motor over servo motor as it is more accurate and cost effective.
* It is decided to use RFID card for online payment as it is more convenient. Because if a person does not have an online payment option, then cannot buy the medicines referring to “RFID Based Shop Billing Machine using Raspberry Pi.” by Ms. Disha Dilip Ved1, Prof. P. B. Ghewari.

## Chapter 3

### Proposed System

**Introduction**

Smart Medicine dispenser is a vending machine which will provide general medicines, sanitary pads, balms, sprays, ointments and first aid.

The main motive of this project is to help people to get required medicines or medical aids such as prays, balm, ointments, etc. from this vending machine. It can be installed at public places like airports, railway stations, malls, bus stands, hotels, restaurants, highways, schools and colleges.

**System Block Diagram**

Given below is the block diagram which explains how our software and hardware will interact with each other for the vending machine to give the required output.

A picture containing chart

Description automatically generated

Given below is the flow chart to show the proper sequence on how the whole system will work.

Diagram

Description automatically generated

**Theory of operations and Process Design**

This system is a made in a way to give a user friendly environment for all the users and dispense the correct medicine and proper quantity provide by the user.

The basic flow of operation is as follow:

* The inventory will be updated after every transaction and refilling the inventory.
* An RFID card will be inserted, then the system will check the RFID card details.
* All available medicines will be displayed with cost and its quantity on LCD screen.
* Customer has to select the medicine name which is available and its quantity in the system.
* The system will automatically calculate the cost of medicine depending on the quantity and the cost will be displayed on the screen.
* As the customer will give a confirmation, the system will deducted money/amount from account and it will be updated.
* Then system will drop the required medicine and then controller will again update the inventory.

**Details of Hardware & Software**

**Hardware:**

**Raspberry Pi 3B+:**



The Raspberry Pi is a remarkable device: a fully functional computer in a tiny and low-cost package. Whether you’re looking for a device you can use to browse the web or play games, are interested in learning how to write your own programs, or are looking to create your own circuits and physical devices, the Raspberry Pi – and its amazing community – will support you every step of the way.

The Raspberry Pi is known as a single-board computer, which means exactly what it sounds like: it’s a computer, just like a desktop, laptop, or smartphone, but built on a single printed circuit board. Like most single-board computers, the Raspberry Pi is small – roughly the same footprint as a credit card – but that doesn’t mean it’s not powerful: a Raspberry Pi can do anything a bigger and more power-hungry computer can do, though not necessarily as quickly.

**7 Inch Touch Screen LCD Display:**



The 7 Inch LCD Touch Display with Driver Board Kit for Raspberry Pi gives users the ability to create an all-in-one, integrated projects such as tablets, infotainment systems, and embedded projects. The 800×480 physical resolutions (resolution range can be adjusted between 640 x 480 to 1600 x 1200) display connects via an adapter board which handles power and signal conversion. Only two connections to the Pi are required; power from the Pi’s GPIO port and a ribbon cable that connects to the DSI port present on all Raspberry Pi’s. Touchscreen drivers with support for 10-finger touch and an on-screen keyboard will be integrated into the latest Raspbian OS for full functionality without a physical keyboard or mouse.

### **Package Included:**

* **1 x 7-inch LCD display**
* **1 x USB cable for the Touch Screen**
* **1 x HDMI to HDMI cable**

**RFID 13.56MHZ Reader Writer Module:**

****

RC522 - RFID Reader / Writer 13.56MHz with Cards Kit includes a 13.56MHz RF reader cum writer module that uses an RC522 IC and two S50 RFID cards. The MF RC522 is a highly integrated transmission module for contact-less communication at 13.56 MHz. RC522 supports ISO 14443A/MIFARE mode.

RC522 - RFID Reader features an outstanding modulation and demodulation algorithm to serve effortless RF communication at 13.56 MHz. The S50 RFID Cards will ease up the process helping you to learn and add the 13.56 MHz RF transition to your project.

The module uses SPI to communicate with microcontrollers. The open-hardware community already has a lot of projects exploiting the RC522 – RFID Communication, using Arduino.

Package Content: -

* 1x  A RFID-RC522 Module
* 1x  RFID Blank Card
* 1x  special-shaped card (as shown by the key ring shape)
* 1x Straight Pin
* 1x Curved Pin

**Stepper Motor:**

DC motor drives are defined as amplifiers or power modules that interface between a controller and a DC motor. They convert step and direction input from the controller to currents and voltages compatible with the motor. These units are sometimes called variable speed drives, referring to a majority of DC motor drives which adjust shaft speed. In industry, a 'drive controller' is a motor drive which incorporates functions of a programmable logic controller (PLC) and drive interface to regulate the speed, torque, horsepower, and direction of a DC motor.

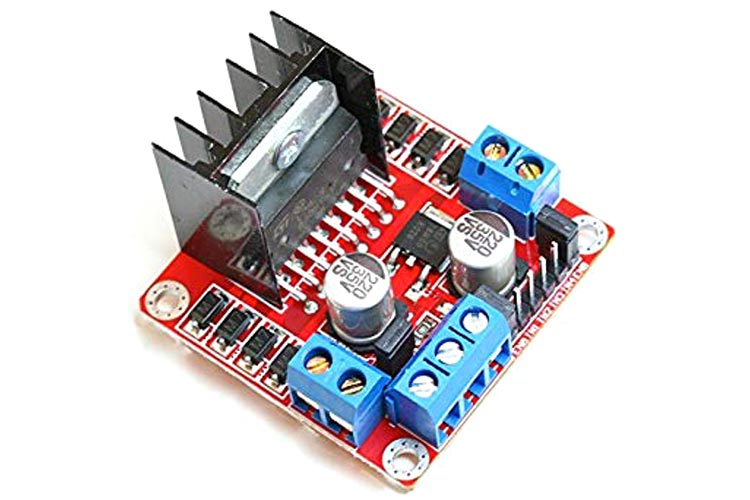
DC motors tend to be less complex than AC motors and are normally less expensive for most horsepower ratings. They are capable of providing large startup torques exceeding 400% of the rated continuous torque. They have a long history of use in variable speed applications with a wide range of options available for this purpose.

**L298N Driver:**

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

Features & Specifications:

1. Driver Model: L298N 2A
2. Driver Chip: Double H Bridge L298N
3. Motor Supply Voltage (Maximum): 46V
4. Motor Supply Current (Maximum): 2A
5. Logic Voltage: 5V
6. Driver Voltage: 5-35V
7. Driver Current:2A
8. Logical Current:0-36mA
9. Maximum Power (W): 25W



**Helical spring like dispensing mechanism:**

This helical shape is used to dispense the medicines by means of pushing the medicines by a set amount with each rotation, this mechanism is fairly common in many candy dispensing machines and this allows us to refill the inventory in a very fast and in a manner which does not involve reloading or any other mechanical complexities, the material is sturdy enough to handle the load of the medicines and will not easily break or deform in a manner that will hamper the performance.

**Software**

**Front-End Languages:**

HTML and CSS5 are used for building GUI. The GUI consist of grid format which shows pictures, name, cost and details of each medicines and other medical aids along with an add button. On right hand side there is a checkout cart which shows selected meds and the total cost.

**Back-end Languages:**

Python language is used for backend and to control RFID module and stepper motor via Raspberry Pi 3B+. JavaScript is used for computing the data and AJAX is used to send data to python file.

**Flask Frame Work:**

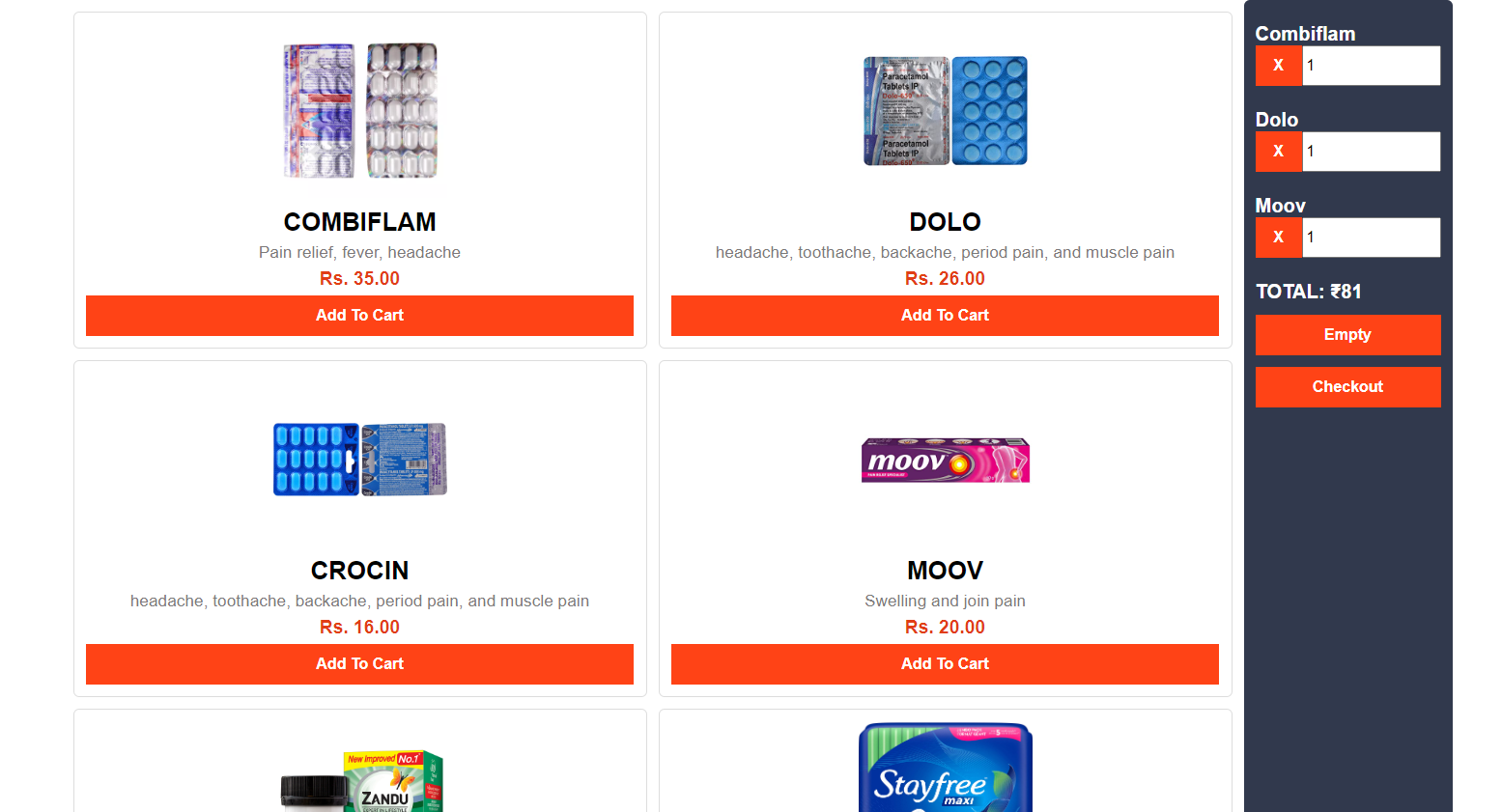
Flask is a web framework; it is a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it has a micro framework that doesn’t include an ORM (Object Relational Manager) or such features. Unlike the Django framework, Flask is very Pythonic. On top of that it’s very explicit, which increases readability.

We used flask frame work to interact backend with front end as our backend is written in python language and front end is in HTML and CSS language.

**Database:** MySQL database system is used to store data of users, inventory and bills.

**Experimental setup**

Software:



### Chapter 4

### Results and Discussion

**Description of Results**

The machine will provide medicines and medical aids as required by the user once the payment for the meds are completed using the RFID card. Once the process is completed the database will be updated.

**Application area**

This project can be installed at public places like airports, railway stations, malls, bus stands, hotels, restaurants, near highways, offices, playgrounds, gardens, libraries, schools, university and colleges.

**Conclusion**

This vending machine is an effective and helpful way of making it easier for user to buy any required first aid or medicines instead of going to pharmacies which can be quit far away from places such as railway stations, bus stops, schools, colleges, airports, etc.

Smart medicine dispenser can be also installed in colleges for teachers and students to get pain killers or sanitary pads without leaving campus.

This machine can also be installed in areas where pharmacies are far away from homes so that people don’t have to travel for just small medicines.

**Future Work**

Replace RFID card system with QR code money transaction or install currency detector. We can also replace our current LCD with a large size Touch Screen LCD and a big container for contain many medicines and first aids. Also provide receipt of the payment via a printer.

### 

### References

1. Lydia Kanguvi. “Problem Statements.” Slide Share. <https://www.slideshare.net/LydiaKanguvi/problem-statements-25028233> [accessed Aug. 07. 2013]
2. Ms. Disha Dilip Ved1, Prof. P. B. Ghewari. “RFID Based Shop Billing Machine using Raspberry Pi.” IRJET. <https://www.irjet.net/archives/V6/i9/IRJET-V6I904.pdf>.
3. Hoffman Arlan J. “Vending machine operating mechanism.” Google Patents. <https://patents.google.com/patent/US5070986A/en>.
4. [Pei-Hsuan Tsai](https://www.researchgate.net/scientific-contributions/Pei-Hsuan-Tsai-2107481030), [Tsung-Yen Chen](https://www.researchgate.net/scientific-contributions/Tsung-Yen-Chen-71013194), [Chi-Ren Yu](https://www.researchgate.net/scientific-contributions/Chi-Ren-Yu-74558758), [Chi-Sheng Shih](https://www.researchgate.net/profile/Chi-Sheng-Shih). “Smart Medication Dispenser: Design, Architecture and Implementation.” ResearchGate. <https://www.researchgate.net/publication/224177659_Smart_Medication_Dispenser_Design_Architecture_and_Implementation>. [accessed April 2011]

### Program Code

Cart.js:

var cart = {

  // (A) PROPERTIES

  hPdt : null,      // html products list

  hItems : null,    // html current cart

  items : {},       // current items in cart

  iURL : "images/", // product image url folder

  totalAmount : 0,

  set setTotalAmount(totalAmount) {

    this.totalAmount = totalAmount

  },

  get getTotalAmount() {

    return this.totalAmount

  },

  // (B) LOCALSTORAGE CART

  // (B1) SAVE CURRENT CART INTO LOCALSTORAGE

  save : () => {

    localStorage.setItem("cart", JSON.stringify(cart.items));

  },

  // (B2) LOAD CART FROM LOCALSTORAGE

  load : () => {

    cart.items = localStorage.getItem("cart");

    if (cart.items == null) { cart.items = {}; }

    else { cart.items = JSON.parse(cart.items); }

  },

  // (B3) EMPTY ENTIRE CART

  nuke : () => { if (confirm("Empty cart?")) {

    cart.items = {};

    localStorage.removeItem("cart");

    cart.list();

  }},

  // (C) INITIALIZE

  init : () => {

    // (C1) GET HTML ELEMENTS

    cart.hPdt = document.getElementById("cart-products");

    cart.hItems = document.getElementById("cart-items");

    // (C2) DRAW PRODUCTS LIST

    cart.hPdt.innerHTML = "";

    let template = document.getElementById("template-product").content,

        p, item, part;

    for (let id in products) {

      p = products[id];

      item = template.cloneNode(true);

      item.querySelector(".p-img").src = cart.iURL + p.img;

      item.querySelector(".p-name").textContent = p.name;

      item.querySelector(".p-desc").textContent = p.desc;

      item.querySelector(".p-price").textContent = "Rs. " + p.price.toFixed(2);

      item.querySelector(".p-add").onclick = () => { cart.add(id); };

      cart.hPdt.appendChild(item);

    }

    // (C3) LOAD CART FROM PREVIOUS SESSION

    cart.load();

    // (C4) LIST CURRENT CART ITEMS

    cart.list();

  },

  // (D) LIST CURRENT CART ITEMS (IN HTML)

  list : () => {

    // (D1) RESET

    cart.hItems.innerHTML = "";

    let item, part, pdt, empty = true;

    for (let key in cart.items) {

      if (cart.items.hasOwnProperty(key)) { empty = false; break; }

    }

    // (D2) CART IS EMPTY

    if (empty) {

      item = document.createElement("div");

      item.innerHTML = "Cart is empty";

      cart.hItems.appendChild(item);

    }

    // (D3) CART IS NOT EMPTY - LIST ITEMS

    else {

      let template = document.getElementById("template-cart").content,

          p, total = 0, subtotal = 0;

      for (let id in cart.items) {

        // (D3-1) PRODUCT ITEM

        p = products[id];

        item = template.cloneNode(true);

        item.querySelector(".c-del").onclick = () => { cart.remove(id); };

        item.querySelector(".c-name").textContent = p.name;

        item.querySelector(".c-qty").value = cart.items[id];

        item.querySelector(".c-qty").onchange = function () { cart.change(id, this.value); };

        cart.hItems.appendChild(item);

        // (D3-2) SUBTOTAL

        subtotal = cart.items[id] \* p.price;

        total += subtotal;

      }

      // (D3-3) TOTAL AMOUNT

      item = document.createElement("div");

      item.className = "c-total";

      item.id = "c-total";

      item.innerHTML ="TOTAL: &#8377;" + total;

      cart.hItems.appendChild(item);

      // (D3-4) EMPTY & CHECKOUT

      item = document.getElementById("template-cart-checkout").content.cloneNode(true);

      cart.hItems.appendChild(item);

    }

  },

  // (E) ADD ITEM INTO CART

  add : (id) => {

    if (cart.items[id] == undefined) { cart.items[id] = 1; }

    else { cart.items[id]++; }

    cart.save(); cart.list();

  },

  // var: total = 0,

  // (F) CHANGE QUANTITY

  change : (pid, qty) => {

    // (F1) REMOVE ITEM

    if (qty <= 0) {

      delete cart.items[pid];

      cart.save(); cart.list();

    }

    // (F2) UPDATE TOTAL ONLY

    else {

      cart.items[pid] = qty;

      var total = 0;

      for (let id in cart.items) {

        total += cart.items[id] \* products[id].price;

        document.getElementById("c-total").innerHTML ="TOTAL: &#8377;" + total;

      }

    }

  },

  // (G) REMOVE ITEM FROM CART

  remove : (id) => {

    delete cart.items[id];

    cart.save();

    cart.list();

  },

  // (H) CHECKOUT

  checkout : () => {

    // alert(this.getTotalAmount);

    var ans = JSON.stringify(cart.items)

    // console.log(ans);

    $.ajax({

      url: '/cart',

      data: ans,

      type: 'POST',

      contentType: 'application/json',

      dataType: 'json',

      success: function(response){

        console.log('response');

      },

      error: function(error){

          console.log(error);

      }

    });

  }

};

window.addEventListener("DOMContentLoaded", cart.init);