University of Massachusetts Boston

Medpick

Pinaka

Students

- Abdulaziz Al-Boainain `a.alboainain001@umb.edu`
- Dor David Vankin `dor.vankin001@umb.edu`
- Son TranDuc, `son.tran001@umb.edu`
- Antoine (Tony) Rahme, `antoine.rahme001@umb.edu`
- Rahul Nagesh, `rahul.nagesh001@umb.edu`

Clients

- Ranjit Makam, `rkmakam@medorna.com`
- Soujanya Parvataneni, `anya@medorna.com`

Pinaka Project Final Document v1.1.0 05/22/2020

1 Introduction

1.1 Purpose

In *Medpicks*'s current operation, medical prescriptions get delivered to pharmacies using the company's *Pinaka* app.
In this workflow, doctors hand over prescriptions to their patients, who upload the prescription to the company's database via *Pinaka* app. Finally, the prescription is sent from the database to the patient's chosen pharmacy.

We offer a workflow that minimizes the patient's steps in delivering the prescription to the pharmacy.
The proposed software will create an easier experience for patients in India, in which their prescriptions will be sent directly to the patients' desired pharmacies without effort on their part.

Intended audience

- Doctors that will use the web application.

1.2 Scope

Pinaka is an application created by Medpick based in India. The software will allow doctors to either upload a prescription or draw a prescription on the web canvas. The prescription, whether uploaded or written using the canvas, will be sent directly to the patient's preferred pharmacy.
Our software will be usable for all platforms that support web applications (iOS, Android).

1.3 Overview

The aim of the project is to eliminate steps between the doctor prescribing the medicine and pharmacies receiving the prescribed medication.
Our two objectives are the following:

- Creating a sketchpad feature that allows doctors to draw prescription drugs directly in the app and upload to the database, which will then be sent to the pharmacy.
- Allowing doctor/user upload from Google Drive.

1.3.1 Existing Work

- e-Prescribing
  ”e-Prescribing is defined as the process of electronically generating and sending a prescription order, so that physicians and other medical practitioners can transmit an electronic prescription to a pharmacy directly from the point of care” (https://www.practicefusion.com/e-prescribing/)

- Notable mention: Amwell
  Amwell is a service that allows users to get online prescriptions and also allows them to select pharmacies. (amwell.com)

1.3.2 Limitations

- Only works for *Medpick*'s associates

1.4 Definitions

- API: Application Programming Interface.
- Pinaka App: Medpicks's own Android app for medial purchases from local pharmacies.
- MedpickDB: Medpick's Database
- GDR: Google Drive Retrieval

- WAD: Web App Drawing.
- JPEG: Joint Photography Experts Group. File extension for processed images.
- PNG: Portable Network Graphics. File extension for processed images.
- PDF: Portable Document Format. File extension that presents texts and images formatted independently across all platform files.
- GDrive: Google Drive

2 Requirements

For GDR user

- **R1** Can run on Windows and Mac.
- **R2** Allows retrieval of prescription image from GDrive and send to preferred pharmacy

For WAD user

- **R3** Can run Android and Apple Mobile devices.
- **R4** Doctor can write the prescription on the digital canvas and send it directly to patients' preferred pharmacies.

3 Specifications

- **R1** GDR can run on Windows and Mac and supports the following browsers: Internet Explorer 9+, Firefox 19.0+, Chrome 25.0+, Safari 5.1+
- **R2** GDR Requires GDrive to be mounted to PC as a Network drive. The operator selects the prescription image in this Network drive and submit to the preferred pharmacy.
- **R3** WAD supports the browsers mentioned in **R1**, and is designed to work on bigger mobile devices (tablets). WAD also responses to mouse inputs.
- **R4** The Web App (WAD) has the digital canvas built-in. Doctors write prescriptions on screen and send it to any pharmacy where the patient requested. The prescription image can also be saved to, and printed from the devices
- **GDR** and **WAD** can retrieve any image format that Pinaka supports.
- **WAD** can be integrated via iframe in the Android app

4 Design

4.1 Use Cases

4.1.1 Use Case : Syncing prescription with database via WAD

Pre- Conditions for WebApp:

- Requires internet connection
- Requires a device that can access the web
- WebApp can be accessed through *Pinaka* app, or web browsers
- The user has to provide a valid phone number that is associated with a registered *Pinaka* user

WAD requires the user to choose targeted pharmacy from a list, provide *Pinaka* user's phone number, and draw the prescription on a digital canvas. Once the prescription is submitted, the user will view a success message.

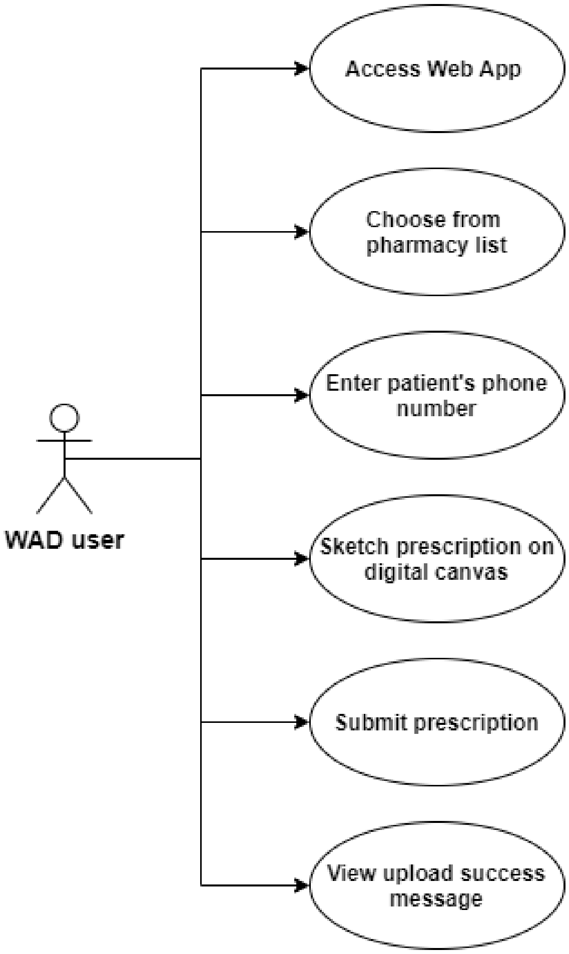

Access Web App
Choose from pharmacy list
Enter patient's phone number
WAD user
Sketch prescription on digital canvas
Submit prescription
View upload success message

Figure 1: Use case for WebApp user

4.1.2 Use Case : Syncing prescription with database via GDR

Pre Conditions:

- Requires internet connection
- Requires the use of a computer
- The user should have a Google Drive account
- The user should mount GDrive folder prior to using GDR
- The user has to provide a valid phone number that is associated with a registered *Pinaka* user

The user is required to mount the source Google Drive folder into his/hers local file system on the computer (see ”How-toMountGDriveToPC.txt”).
After accessing GDR, the user will choose the targeted pharmacy, provide *Pinaka* user’s phone number, and upload the prescription image from the previously mounted GDrive folder using the ”choose file” option.
Once the prescription is submitted, the user will view a success message.

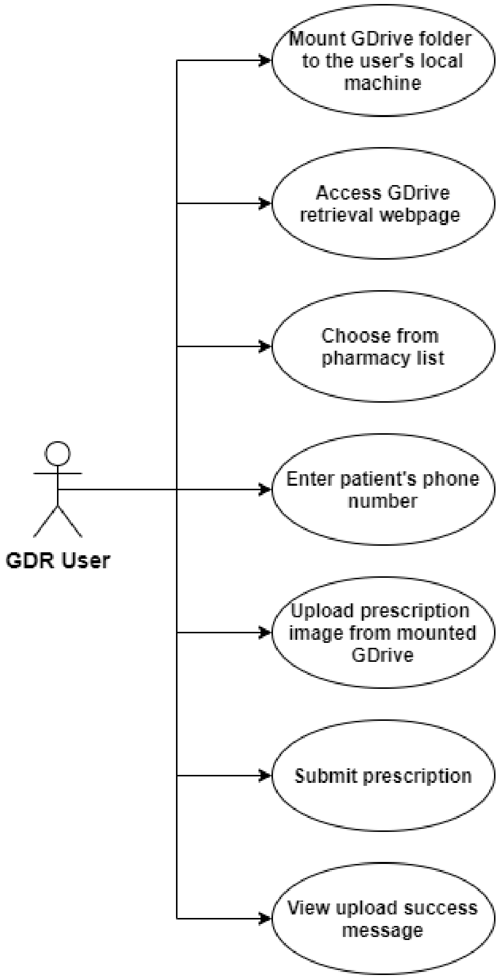

GDR User
Mount GDrive folder to the user's local machine
Access GDrive retrieval webpage
Choose from pharmacy list
Enter patient's phone number
Upload prescription image from mounted GDrive
Submit prescription
View upload success message

Figure 2: Use case for GDR user

How To Mount GDrive:

Run the following commands on command prompt:

1. subst g: ”C:Drive”
2. Type ”subst x: /d”, replacing ”x:” with the drive letter you want to remove.

4.2 Architecture

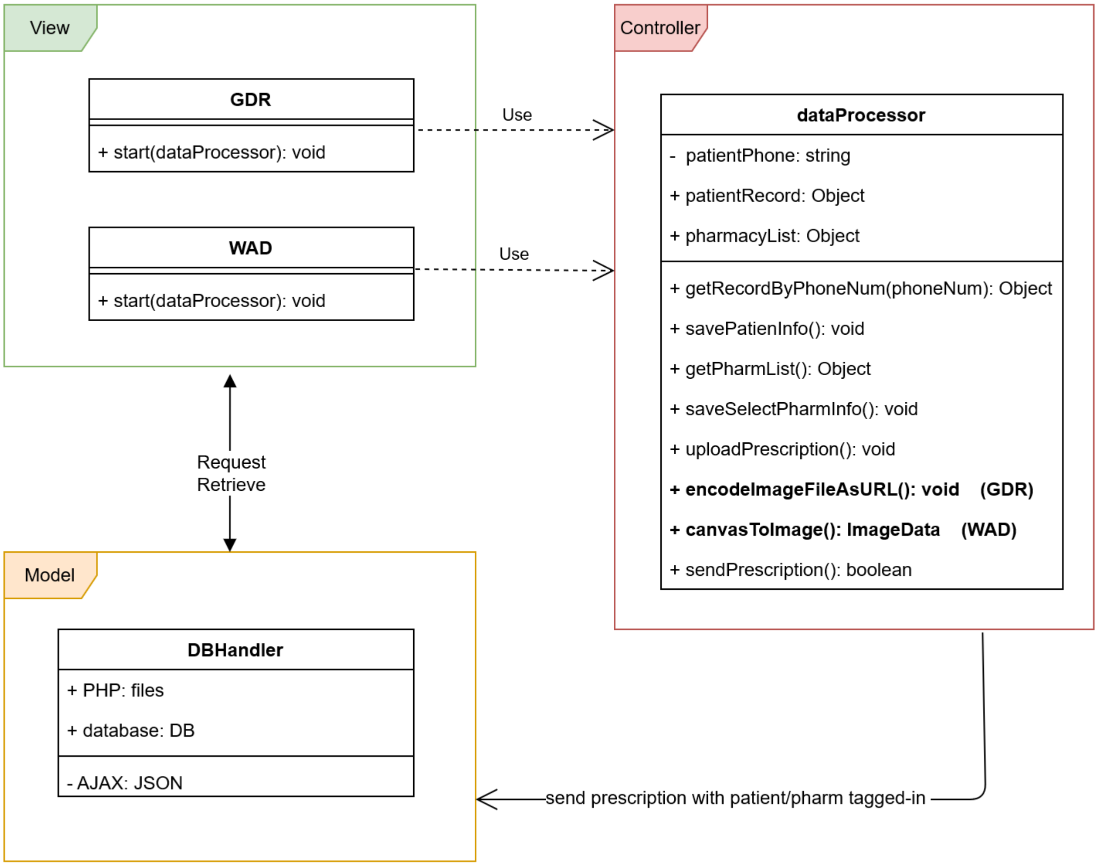

View
GDR
+ start(dataProcessor): void
WAD
+ start(dataProcessor): void
Use
Use
Controller
dataProcessor
- patientPhone: string
+ patientRecord: Object
+ pharmacyList: Object
+ getRecordByPhoneNum(phoneNum): Object
+ savePatienInfo(): void
+ getPharmList(): Object
+ saveSelectPharmInfo(): void
+ uploadPrescription(): void
+ encodeImageFileAsURL(): void (GDR)
+ canvasToImage(): ImageData (WAD)
+ sendPrescription(): boolean
Request
Retrieve
Model
DBHandler
+ PHP: files
+ database: DB
- AJAX: JSON
send prescription with patient/pharm tagged-in

Figure 3: GDR and WAD Class Diagram

.

4.3 Workflows

The workflow of our project can be described with activity diagram in Figure 4.
The main component of the workflow is *MedpickDB*.
Web App Drawing and *Google Drive Retrieval* will act as two methods to enter and update the database. WAD's user draws the prescription on the digital canvas, while GDR's user is required to mount GDrive folder to his/hers local machine and use it as the source folder when uploading prescription image.
Both WAD and GDR require the user to choose the intended pharmacy and provide a valid phone number associated with a registered *Pinaka* user. Then, WAD/GDR will retrieve the relevant user data from *MedpickDB*, and send a prescription order request to the database.

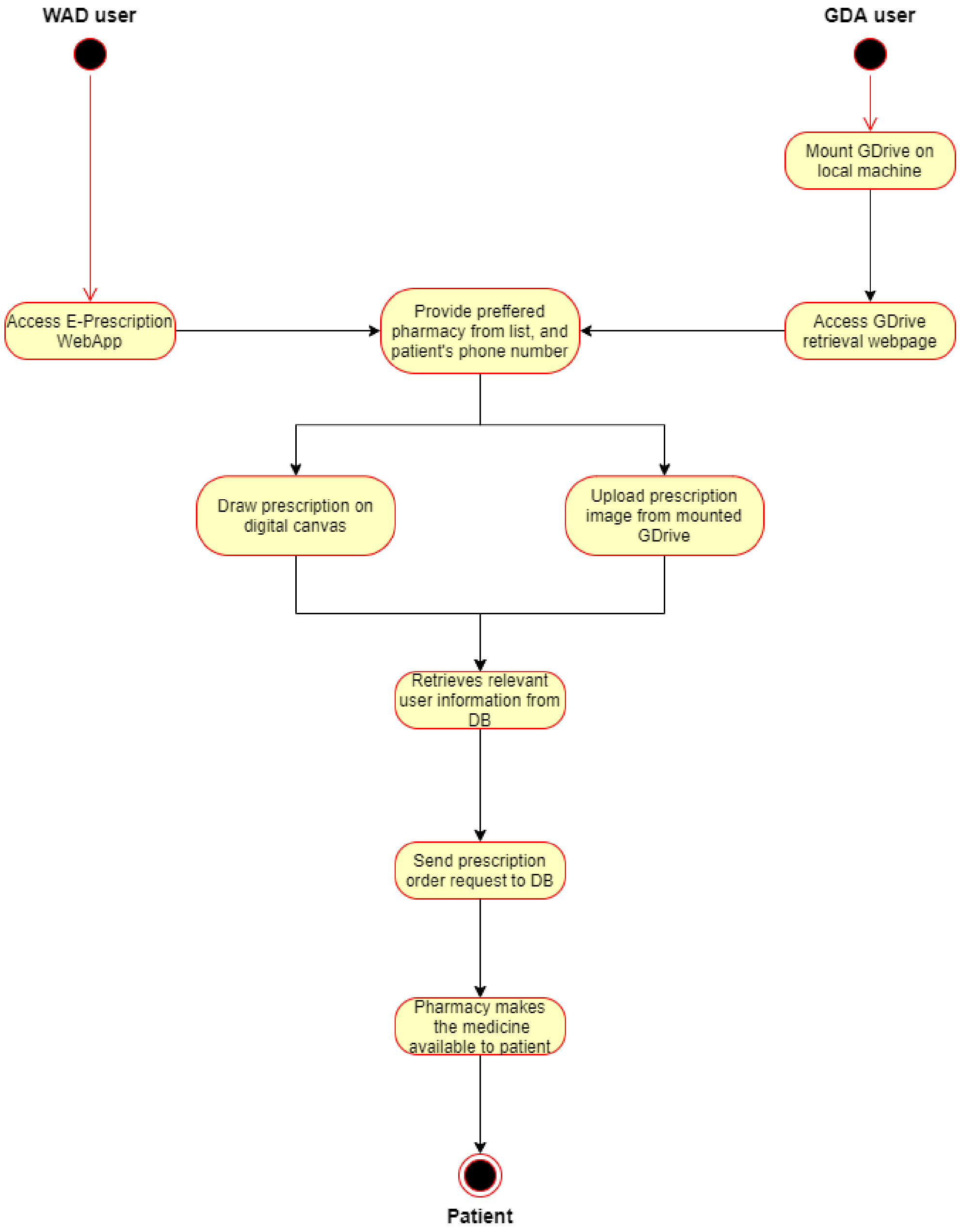

WAD user
GDA user
Mount GDrive on local machine
Access E-Prescription WebApp
Provide preffered pharmacy from list, and patient's phone number
Access GDrive retrieval webpage
Draw prescription on digital canvas
Upload prescription image from mounted GDrive
Retrieves relevant user information from DB
Send prescription order request to DB
Pharmacy makes the medicine available to patient
Patient

Figure 4: Activity diagram for main workflow

4.4 Technologies and Implementation Details

4.4.1 Technologies

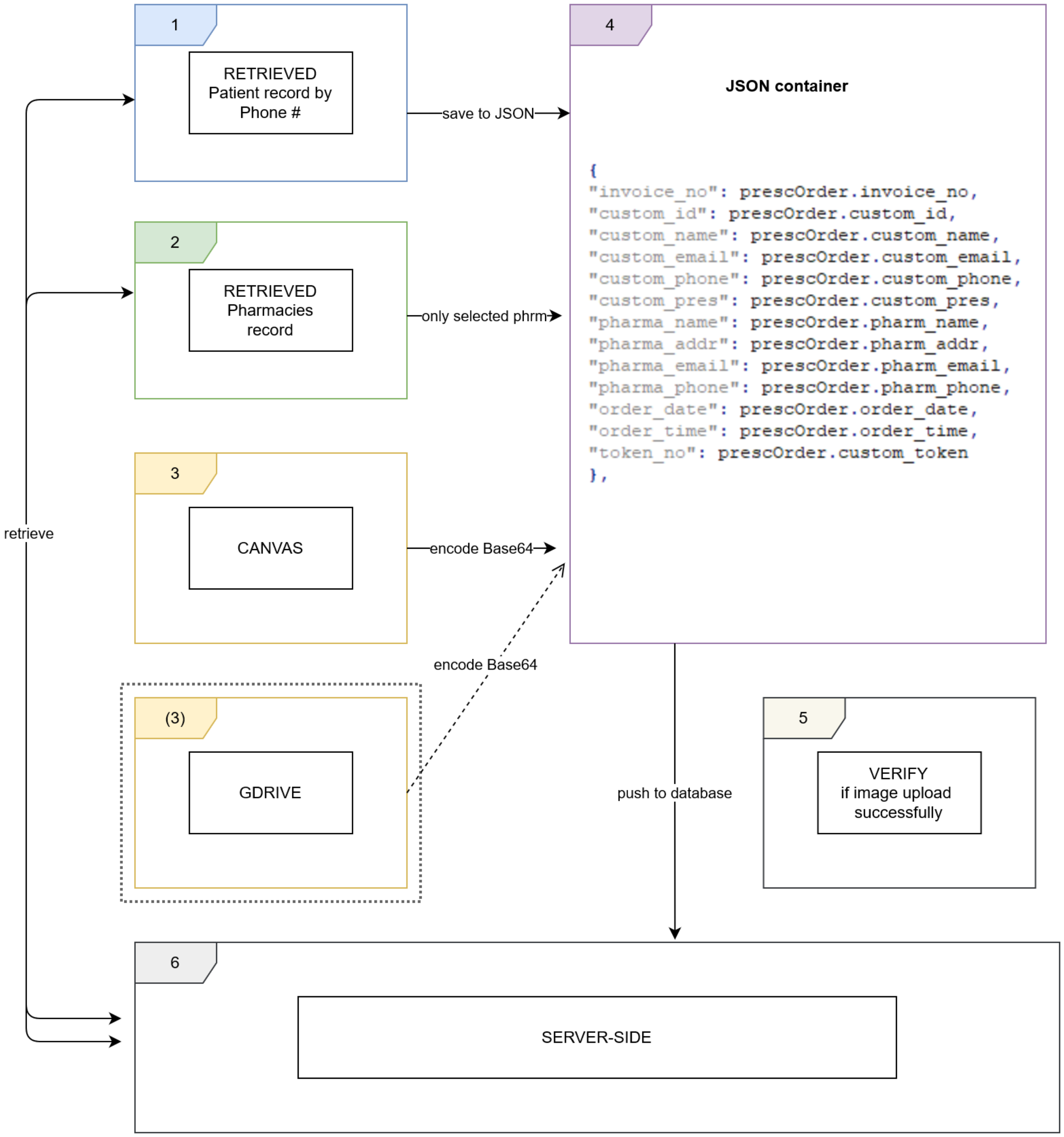
- HTML5 and CSS for UI
- HTML5 CANVAS
- Javascript
- MVC Modeling
- OOP - Classes and Objects
- Singleton
- AJAX
- JSON

4.4.2 Implementation

- **GDR** Retrieve the prescription image from Google Drive and insert it to existing Medpick database table. Implemented in Javascript languge.
  First, retrieving patient's record by patient phone number. Function getRecordByPhoneNum() uses AJAX to communicate with server thru the file retrieve_user_details.php to request the patient's record from database table "prescription_customer_details". Data is saved in JSON container (prescOrder) as tagging to new prescription for sending back to server later on . In the same way, the preferred pharmacies information are retrieved into the list (using function getPharmList()) , from data table "wholesale_users_table", for the doctors to select and the selected pharmacy is also saved it to JSON container prescOrder as tagging. The prescription image is selected from PC network drive and converted to base64 string, and saved in JASON container as "custom_pres". Using AJAX protocol, the JSON object with complete information is finally pushed to the table "prescription_order_consumer" thru the processing of file insert_prescription_order.php.

- **WAD** Use HTML5, CSS and JavaScript to build the canvas. Output from canvas module (image) is converted to base64 string and saved into JSON container ready to push to server-side database table. Processes of retrieving patient record and pharmacy information, and sending JSON to server are same as GDR implementation.

BLOCK DIAGRAM



```
{
"invoice_no": prescOrder.invoice_no,
"custom_id": prescOrder.custom_id,
"custom_name": prescOrder.custom_name,
"custom_email": prescOrder.custom_email,
"custom_phone": prescOrder.custom_phone,
"custom_pres": prescOrder.custom_pres,
"pharma_name": prescOrder.pharm_name,
"pharma_addr": prescOrder.pharm_addr,
"pharma_email": prescOrder.pharm_email,
"pharma_phone": prescOrder.pharm_phone,
"order_date": prescOrder.order_date,
"order_time": prescOrder.order_time,
"token_no": prescOrder.custom_token
},
```

Figure 5: Google Drive API and Web App API Class Diagram

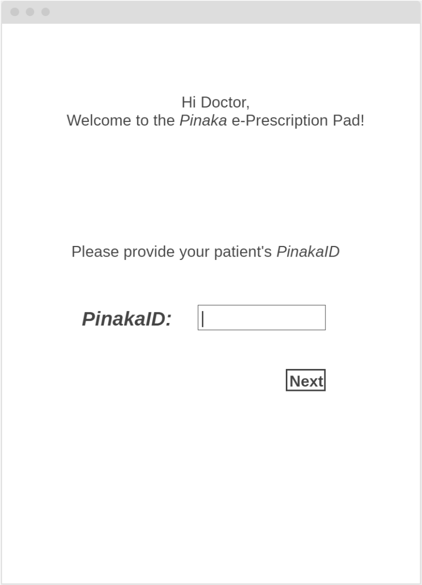
.

4.5 Initially proposed User Interface

Initially, we proposed two user interfaces corresponding to the updating methods: Web App, and Command Line Interface.

- **WAD:** The Web App's main webpage has the digital canvas as its main component.
  in order to continue to the prescription writing page, the doctor has to enter the patient's *PinakaID* in the first page. Once the "Submit" button is clicked, a success message window appear. See Figures 6-8

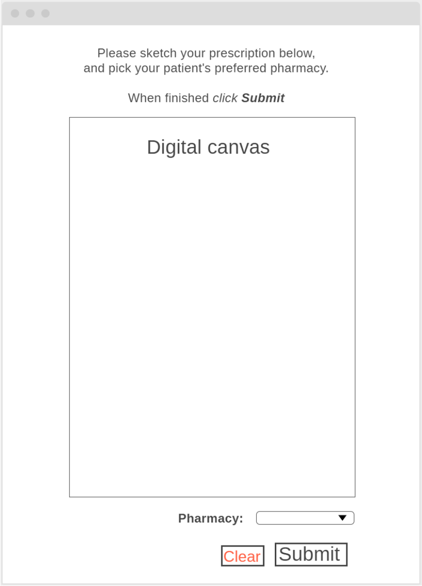
- **GDrive API:** the UI is the computer's own Command Line Interface. Once the program is called with the required arguments, a success message is printed to standard output. See Figure 9



Hi Doctor,
Welcome to the *Pinaka* e-Prescription Pad!

Please provide your patient's *PinakaID*

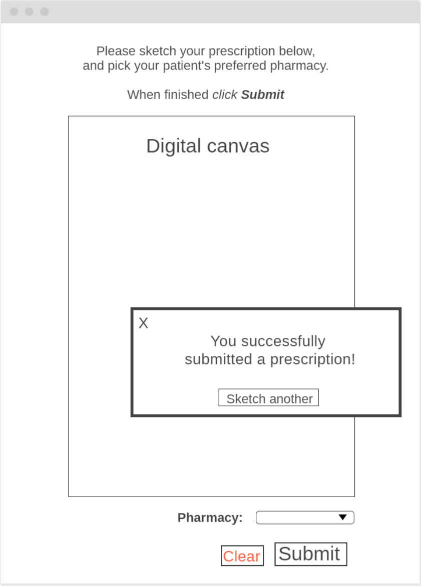
Figure 6: Welcome Webpage



Please sketch your prescription below,
and pick your patient's preferred pharmacy.

When finished *click* ***Submit***

Figure 7: Digital canvas Webpage

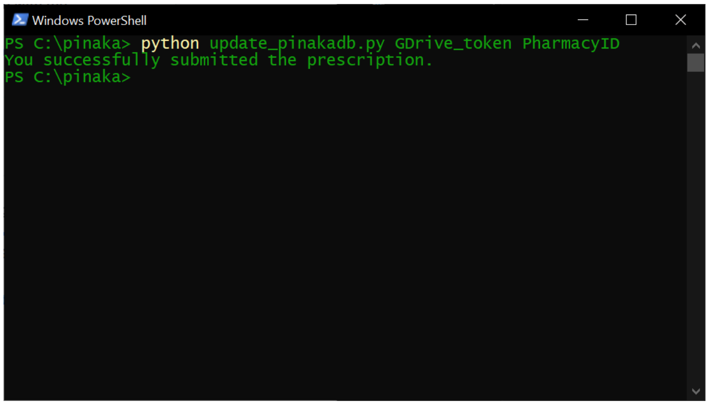


Please sketch your prescription below,
and pick your patient's preferred pharmacy.

When finished *click* ***Submit***

You successfully
submitted a prescription!

Figure 8: Success message



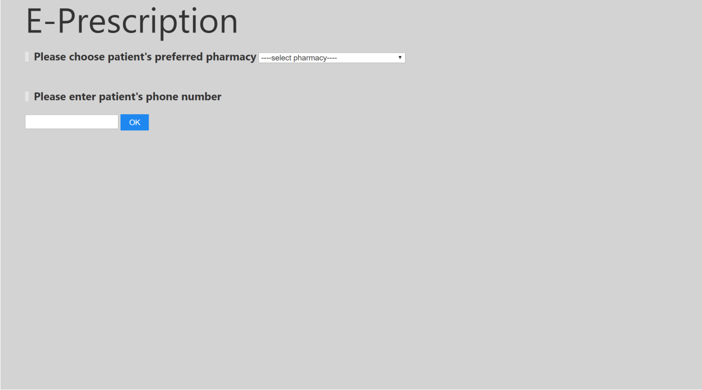
```
PS C:\pinaka> python update_pinakadb.py GDrive_token PharmacyID
You successfully submitted the prescription.
PS C:\pinaka>
```

Figure 9: Command Line Interface UI

4.6 FINAL User Interface

In our final product, we offer two user interfaces corresponding to our two WebApps: Canvas based WebApp, and GDrive retrieval WebApp.

- **WAD:** When users access the WebApp, they are asked to choose a pharmacy from a list, and provide their patient's phone number. Once each of the information above is provided, a list of information about the entity (i.e. pharmacy or patient) is displayed onto the screen for user's assurance. Then, once both requirements are fulfilled, the digital canvas is displayed for the user to draw the prescription on.
  To finish the process and send the prescription order, the user will be prompted to click the "SUBMIT" button and will be greeted with an upload success message. See Figures 10-14
- **GDR:** As with WAD, when users access GDR webpage, they are asked to choose a pharmacy from a list, and provide the patient's phone number. The information about each entity (i.e. pharmacy or patient) is displayed onto the screen, and when both requirements are fulfilled, the ability to browse the computer and retrieve an image appears. The user can then choose to upload directly from the previously mounted GDrive folder on his/hers machine. Once the "SUBMIT" button is clicked, the user will be greeted with an upload success message. See Figures 15-17

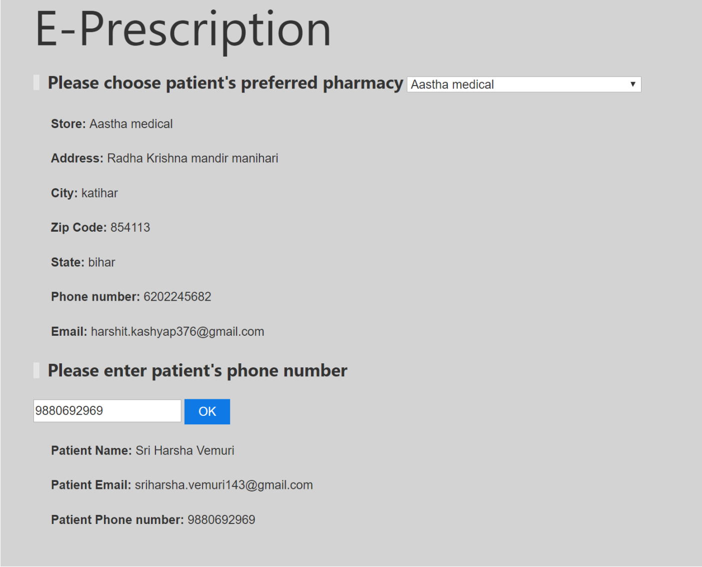


E-Prescription

Please choose patient's preferred pharmacy ----select pharmacy----

Please enter patient's phone number

Figure 10: WAD Welcome Webpage



E-Prescription

Please choose patient's preferred pharmacy Aastha medical

Store: Aastha medical

Address: Radha Krishna mandir manihari

City: katihar

Zip Code: 854113

State: bihar

Phone number: 6202245682

Email: harshit.kashyap376@gmail.com

Please enter patient's phone number

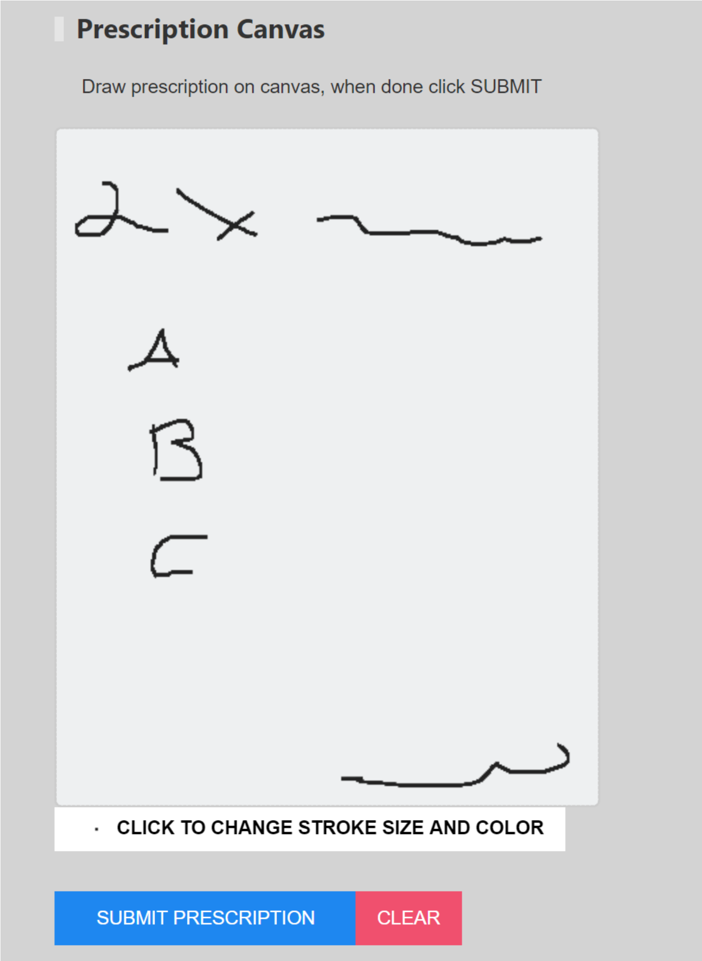
9880692969

Patient Name: Sri Harsha Vemuri

Patient Email: sriharsha.vemuri143@gmail.com

Patient Phone number: 9880692969

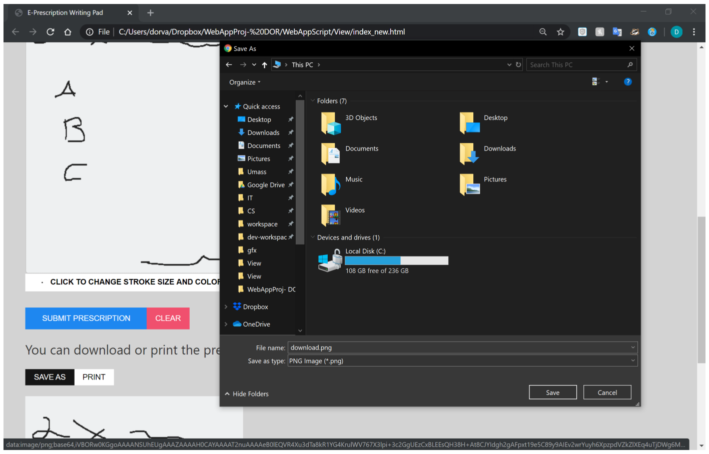
Figure 11: WAD entity information



Prescription Canvas

Draw prescription on canvas, when done click SUBMIT

Figure 12: Digital canvas



File | C:/Users/dorva/Dropbox/WebAppProj-%20DOR/WebAppScript/View/index_new.html

You can download or print the pre

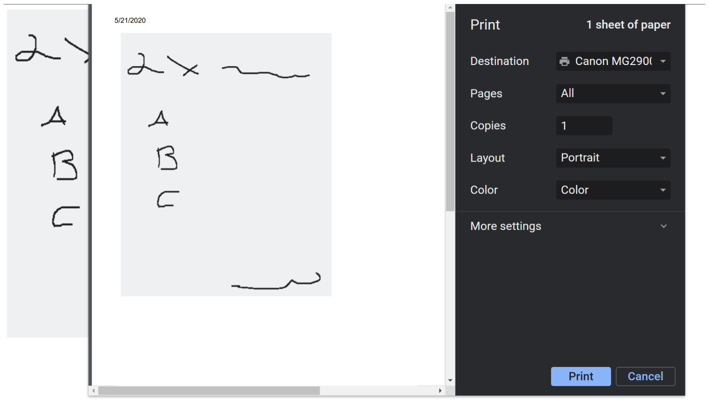
File name: download.png

Save as type: PNG Image (*.png)

Local Disk (C:) 108 GB free of 236 GB

data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAZAAAAH0CAYAAAAT2nuAAAAeB0lEQVR4Xu3dTa8kR1YG4KrulWV767X3lpi+3c2GgUEzCxBLEEsQH38H+At8CJYldgh2gAFpxt19e5C89y9AIEv2wrYuyh6XpzpdVZkZlXEq4uTjEQ4uTjDWg6M...

Figure 13: Save canvas image



Print 1 sheet of paper

Destination: Canon MG290(

Pages: All

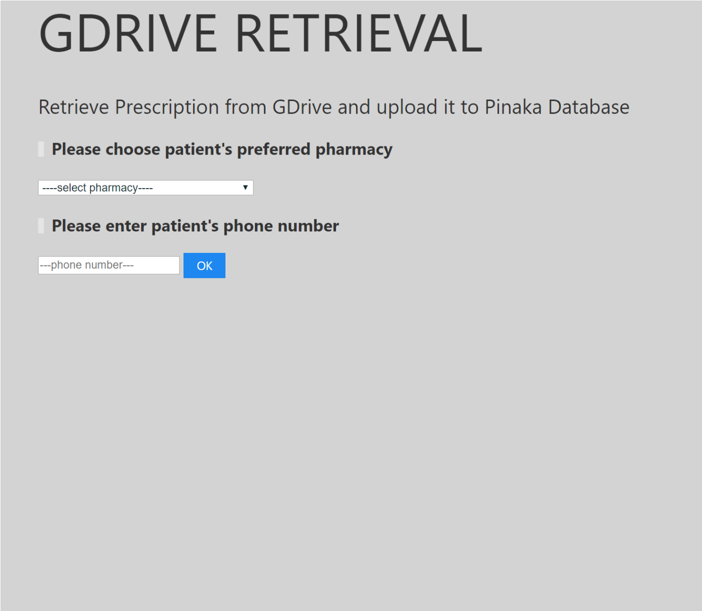
Copies: 1

Layout: Portrait

Color: Color

More settings

Figure 14: Print canvas image



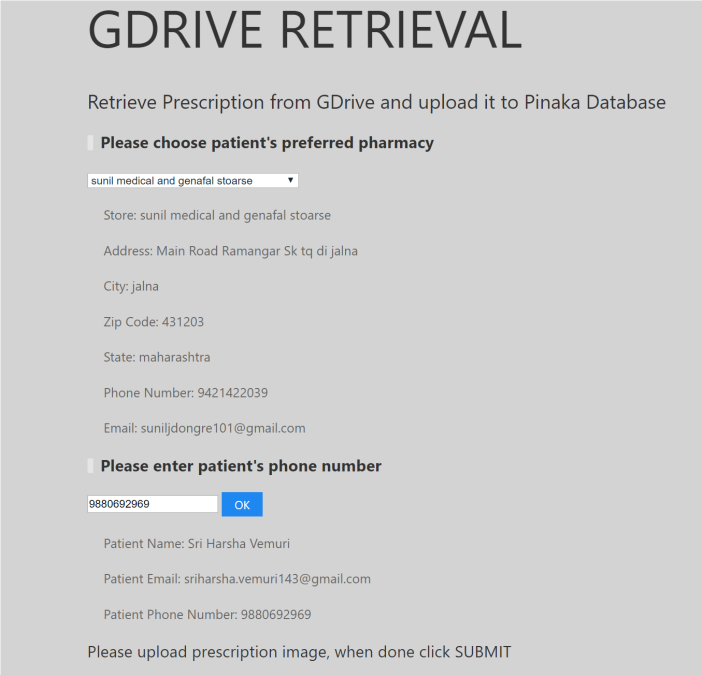
GDRIVE RETRIEVAL

Retrieve Prescription from GDrive and upload it to Pinaka Database

Please choose patient's preferred pharmacy

Please enter patient's phone number

Figure 15: GDR welcome page



GDRIVE RETRIEVAL

Retrieve Prescription from GDrive and upload it to Pinaka Database

Please choose patient's preferred pharmacy

sunil medical and genafal stoarse ▾

Store: sunil medical and genafal stoarse

Address: Main Road Ramangar Sk tq di jalna

City: jalna

Zip Code: 431203

State: maharashtra

Phone Number: 9421422039

Email: suniljdongre101@gmail.com

Please enter patient's phone number

9880692969

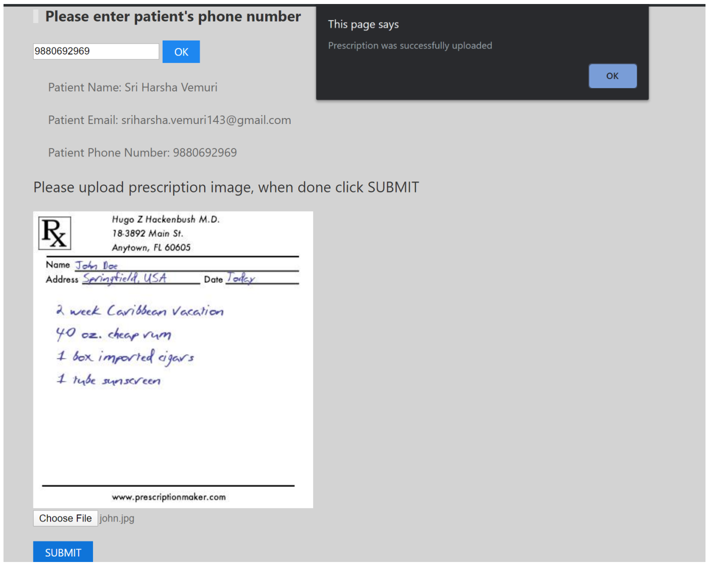
Patient Name: Sri Harsha Vemuri

Patient Email: sriharsha.vemuri143@gmail.com

Patient Phone Number: 9880692969

Please upload prescription image, when done click SUBMIT

Figure 16: GDR entity information



Please enter patient's phone number

9880692969

Patient Name: Sri Harsha Vemuri

Patient Email: sriharsha.vemuri143@gmail.com

Patient Phone Number: 9880692969

This page says

Prescription was successfully uploaded

Please upload prescription image, when done click SUBMIT

Rx

Hugo Z Hackenbush M.D.
18-3892 Main St.
Anytown, FL 60605

Name John Doe

Address Springfield, USA Date Today

2 week Caribbean Vacation
40 oz. cheap rum
1 box imported cigars
1 tube sunscreen

www.prescriptionmaker.com

john.jpg

Figure 17: GDR success message

5 Time Frame and Milestones

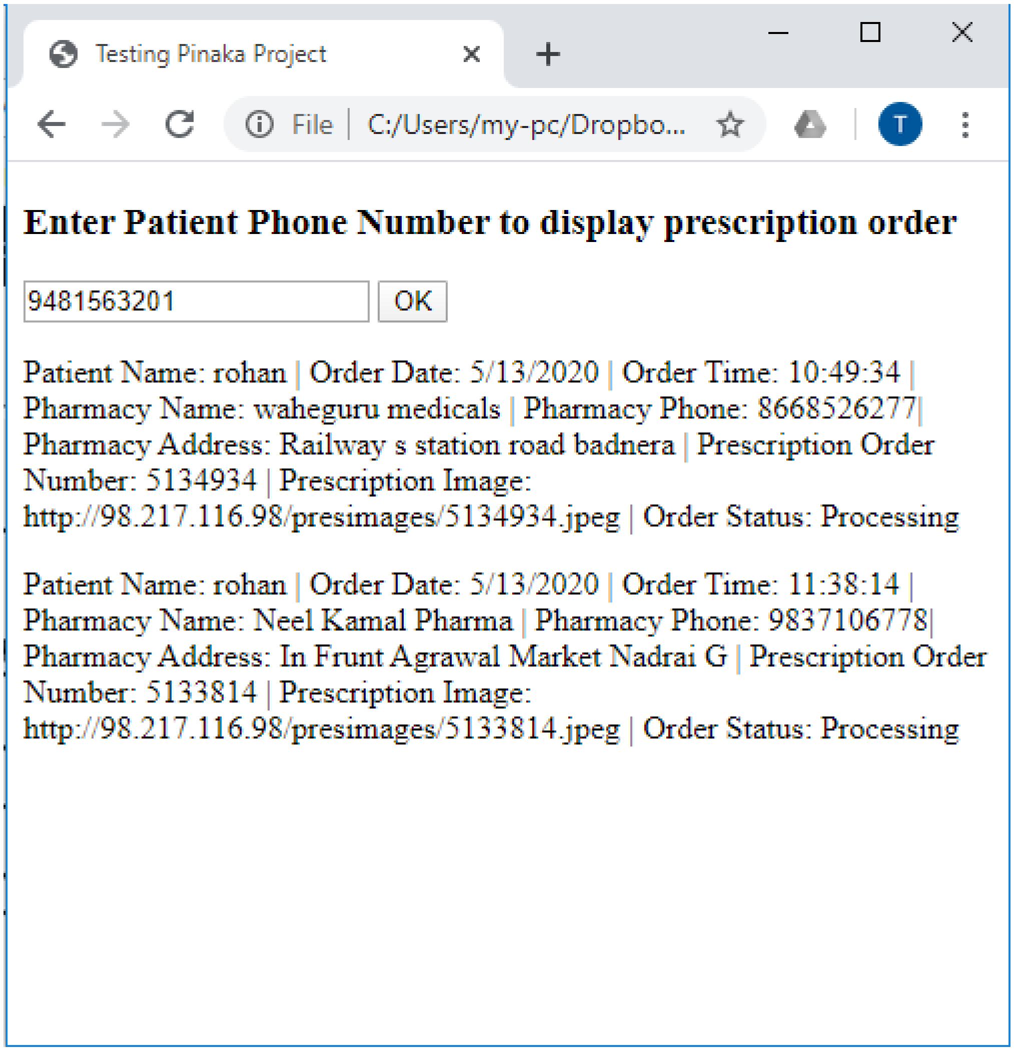
- 03/27 - 04/03:
  - Research on JavaScript implementation of the web-based UI drawing sketchpad.
  - Research on using third-party webpage to send prescription and store it in GDrive folder.
- 04/03 - 04/17:
  - Work on the webpage to implement a sketchpad used for writing down prescription drugs.
  - Work on the third-party webpage to send image to the GDrive folder.
- 04/17 - 05/13:
  - Test feature with dummy database; debugging.
- 05/13:
  - Presentation day!
- 05/13 - 05/22:
  - Test with different devices and finalize documentation.

6 Testing

We used different devices for the webapp starting with the desktop, connecting a tablet to the desktop, phones and tablets. To test compatibility and track any bugs that show up on different devices.

- **Desktop:**
  Tested with different browsers including Edge, Chrome, Firefox and Safari.
- **Desktop with drawing pad:**
  Same browsers but instead of using the mouse the pen was tested and worked smoothly.
- **Tablets:**
  Works perfectly when testing with touch.
- **Phones:**
  Although the touch functionality works, it is recommended to use tablets with bigger screens for better compatibility.

To test uploads to the database, we created **testing.html**, a webpage that shows a summary of information associated with a *Pinaka* user. The entries displayed confirm that prescription orders were successfully sent to the database. See figures 18,19.



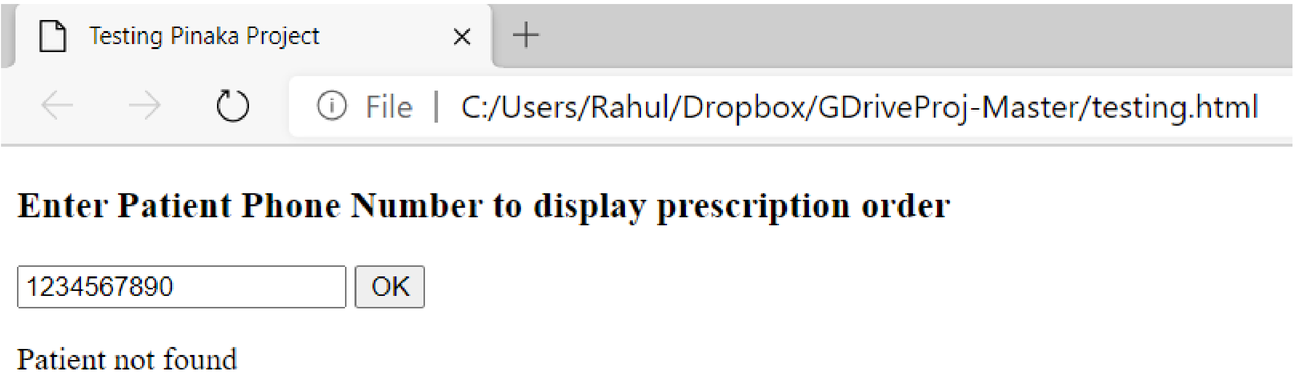
Enter Patient Phone Number to display prescription order

9481563201

Patient Name: rohan | Order Date: 5/13/2020 | Order Time: 10:49:34 | Pharmacy Name: waheguru medicals | Pharmacy Phone: 8668526277| Pharmacy Address: Railway s station road badnera | Prescription Order Number: 5134934 | Prescription Image: http://98.217.116.98/presimages/5134934.jpeg | Order Status: Processing

Patient Name: rohan | Order Date: 5/13/2020 | Order Time: 11:38:14 | Pharmacy Name: Neel Kamal Pharma | Pharmacy Phone: 9837106778| Pharmacy Address: In Frunt Agrawal Market Nadrai G | Prescription Order Number: 5133814 | Prescription Image: http://98.217.116.98/presimages/5133814.jpeg | Order Status: Processing

Figure 18: Change Log Showing Prescription Images Uploaded to DB


Testing Pinaka Project
File | C:/Users/Rahul/Dropbox/GDriveProj-Master/testing.html
OK

Enter Patient Phone Number to display prescription order

1234567890

Patient not found

Figure 19: Throws error when an invalid patient ID is entered

7 Conclusion

In any team, communication is the most important. Having everyone to be on the same page and stick to the outline can be challenging. To overcome this challenge, we created a group chat on **whatsapp** as our main communication platform. With the pandemic it was hard to keep up or do frequent in person meetings. When we wanted to have a group meeting, due to whatsapp's limitations we switched to **Slack** to have multiple users on the same call and have screen sharing conveniently. For coding platform we saved changes on **Dropbox**. The way we avoided any overwriting of other's edits was having each developer copy the original folder, and then present to each other the changes. The pandemic has restricted our more frequent in-person-meetings but thankfully the workflow, deadlines and communication were not affected. Issues we ran into were notably graphical glitches, server glitches including accidentally sent prescription to pharmacies in India. We were able to patch up the testing version. We had to create our own testing server built on XAMPP used to run SQL, PHP, AJAX scripts. The client has sent us PHP files to work with which we have modified to fit requirements of this project.

Overall it has been a great experience nevertheless to work in a team and with a client on a real application.