**CSE 546 — Project Report**

*Akash Patel Disha Bhukte Sapan Desai*

*1219522499 1219495225 1219080070*

# Introduction

We have built an elastic web application using PaaS. We have used the PaaS resource of Google Compute Engine that is Google App Engine. This web application (Medicine Expiry Reminder) will help users to remind them about the expiry date of the medicine. Taking the scenario, no one can buy the medicine without a doctor's prescription. Doctor needs to fill prescriptions that will be transferred to the desired pharmacy company of the users, and every bill that is generated has some kind of barcode or QR code. Now, our system needs certain information such as Medicine name, expiry date, doctor’s name, doctor’s email id, manufacturer of the medicine. So, we proposed a solution that above mentioned information can be easily stored in some form such as QR code or Barcode of the bill. Users just need to scan QR code or Barcode and all the reminders will be set in the google calendar of the users.

*Requirement*:

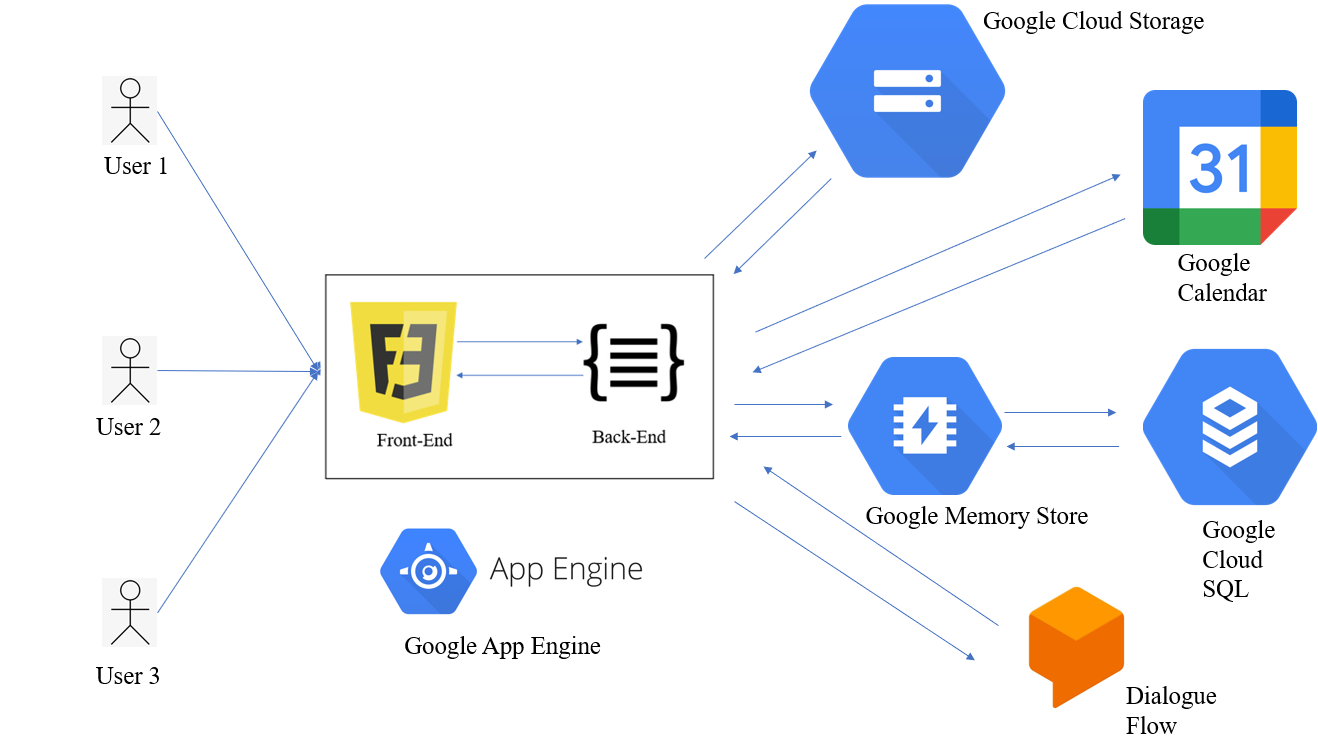
1. The application should be able to scan the QR code or Barcode and extract data.
2. All the reminders should be set correctly in the user's google calendar.
3. The application should be able to delete the reminder if the user wishes to.
4. The application should be able to take feedback about the medicine from the user.
5. The application must be able to send user feedback to their doctors email id.
6. The application should be able to fetch the feedback from the database and should only show that feedback from those users who made their feedback public.
7. The application should not show those feedback from users who have not made their feedback public.
8. The application should be able to answer user queries vai chatbot.
9. The application should also be able to take and send feedback to doctors email id via chatbot.

**2. Background**

* The nearest implementation of our idea is the Fodaffy. It scans the barcode and places the information of the expiry of the food items in the user’s calendar. We have used this idea for our application making it a variant in the field of medicine along with other features which will assist the user in contacting the doctor as well as giving feedback on the medicine.
* Till this date there is no application as such created specifically for medicines which is similar to our idea. The existing medicine applications mostly support the activities for a particular appointment. If extended for user specific scope, there are medicine management applications which remind the user to take the medicines on time.
* This problem is important because usually, we forget to throw away the expired medicines which can be harmful if consumed without checking the expiry date. Secondly, it also provides a feature of providing an immediate feedback to the doctor in case of any reaction to the medication. The application provides a smart way of managing the medicines. It scans the QR code of the medicine The medicine feedback feature is designed to report the doctor immediately regarding any reactions reported by the patient after taking the prescribed medicines
* The existing solution does not provide a feature of scanning the bill for updating information of medicines as well as the doctor’s information to the user’s calendar. Our solution makes the bills user friendly by scanning the QR code provided on the bill and extracting the expiry date of the medicines along with the doctor’s contact information and updating it to the user’s calendar.

# Design and Implementation

# 2.1 Architecture

****

● Frontend:- The frontend has been coded in Javascript. It provides a user-friendly UI for the user to login to the application and scan the QR code on the bill having all the information. It also provides a feature for deleting the calendar reminders for specific as well as all medicines. On opening of the login page, the user is prompted to login via google auth and then he is redirected to the home page where the QR code can be scanned. From here for all the services the requests are sent to the flask API which is the backend. In the medicine feedback feature, the email with the feedback is sent directly to the doctor.

● Backend:- The backend is developed in python using flask. If the request is related to the qr code scan, the backend enters the QR code data as a record in the Cloud SQL database and then updates the user’s calendar with the corresponding medicine records. When the user requests for deletion of any set reminder in his calendar based on the request the flask api communicates with the Google Calendar API. If the user request contains a request for deletion of a specific record only that specific record is deleted. If the user requests for deletion of all records the backend deletes all the records from the user’s calendar. Users can also review the feedback of other users if the feedback is made public by the users.

**Google Cloud Services Used:-**

1. **Google App Engine:-** It is a Pass service which we have used in our project. GAE is deployment and hosting of the web application. In the project, it receives the request from the user and communicates them to other Google services used. The APIs developed using flask have been deployed in GAE. The APIs communicate the user request from the frontend to the other google services and send appropriate responses.
2. **Google Cloud SQL:**- Another service used in our project is google cloud SQL. It is used to store and query SQL data. We are using it as primary storage. All the details of the user specific medication bill information as well as the feedback information is stored in the Cloud SQL
3. **Google DialogFlow:** It is a service offered by google which facilitates easy development of chatbots. We have used this chatbot in order to take feedback from the user about a particular medicine especially in case he experiences any issues and needs immediate medical assistance. On taking the feedback the user will send an email to the doctor notifying him about the emergency.
4. **Google Identity:-** It is a service provided by the google for enabling google sign in as well asking the user for permissions to use data needed by the google services to the user. We have used this service for logging the user into the application and asking thin the permission to put reminders into his calendar
5. **Google Calendar:-** It is a service provided by google to keep a track of reminders and meetings. The meeting and reminder events can be shared with each other and is very convenient for the user to book or schedule appointments.
6. **Google Cloud Memorystore:- I**t is a service provided by google that supports in-memory store for faster performance, high availability and scalability.
7. **Google cloud Storage:-** It is a service provided by Google for reliable and secure object storage.

# 2.2 Role of Google App Engine

For hosting our application, we have used Google App engine. API has been written using python flask module. This has been pushed on GAE using Google cloud shell command. We have also created a folder which has all our HTML/CSS files which are also hosted on our GAE. This is MVC architecture.

# 2.3 Autoscaling

The policy used by the application (present in yaml file) is if the CPU utilization of the machine goes above 50% or if there are more than 10 concurrent requests for the application, the application will scale up. The minimum instances have been kept as 1 and it can scale upto 50 instances. Later if the user demand increases , the maximum limit of 50 instances can be changed to deal with the new user base for the application.

# 2.4 Solution proposed to the problem

The solution that we are currently proposing is a very unique idea that is yet to be implemented by anyone in the field of medicine. There are millions of people who are still not aware about the medicine consumption or rather forget to check if the medicine they are consuming is expired or not. This proposed solution easily helps to eliminate the main problem that is the consumption of medicine before it’s expiry date as the user will get a reminder about the medicine on the day of expiry. Moreover, it gives an additional medium for the patient and doctors to communicate about their problem. The doctors will have all the details and records about the patient and can analyse the situation in a much better way with the help of just a mobile device. This application will not only promote consumption of medicine before its expiry but also will eliminate any chance of consumption after its expiry. Generally after the expiry of medicine the medicine is not effective as it used to be. As the time passes on, the effect of medicine starts deteriorating and at one stage it won’t be beneficial to the user. Thus this application is useful in many certain ways as described above.

**2.5 Explain why your solution is significantly better than the state-of-the-art described**

Our solution is very unique and significantly better than what already exists in the world. It is very easy for the user to use and with just one scan of QR code, the user can set up all the medicine expiry reminders. This will not only save the user’s time but also reduce the workload of setting in the reminders and making sure each of the remainder is ticked off once consumed. The proposed solution also provides one easy way to delete all the medicines that are still on the calendar. Thus the solution provided by us is significantly better than the state-of-the-art described.

# Testing and evaluation

If you have access to a Mac or Linux server, chances are you may already have a really simple http load generating tool installed called Apache Bench, or ab. If you are on windows and have Apache installed, you may also have ab.exe in your apache/bin folder.

Apache bench helps to send the number of desidered requests on the specific web server. With the help of this, we can analyse the autoscaling of the application. As the number of requests increases, the application tends to autoscale the request and scale out the process. Usually with 5000 requests, our web application was able to generate approximately 7 instances to handle the load of those requests. With 10000 requests, around 9 instances were created. When the application was hit by 50000 requests, with every hundred requests concurrently around 50 instances were running in the app engine serving the requests.

Later we tested the system with QR codes having different data. The data was altered in terms of Medicine number ,medicine name, medicine expiry date, doctor name and doctor email id. All of the QR codes resulted in the desired output and we were able to achieve the addition of details in the calendar of the patient.

Moreover, The feedback system was also working as expected as it was sending emails to the doctor for the complaint for any reaction to the medication. Later, The user was also able to review the feedback that was submitted earlier.

# Code

**4.1 Explain in detail the functionality of every program included in the submission zip file.**

**Front-end (HTML)**

* index.html: This file contains the home page code of the web app. This is the initial page where the user can see all the features that can be integrated. Moreover, the QR code can also be scanned and the data will be stored in the user’s calendar via this page.
* deletereminder.html : This file contains the page where a user can easily delete the reminders that were earlier set via the index.html
* medicinefeedback.html: This file contains the page where medicine feedback can be given by the user depending on the medicine added in the calendar.
* medicinereviews.html: This file contains the code for the medicine review, once the user has submitted the feedback on the medicinefeedback.html, it can access those feedbacks on this page.
* qrcodescanner.html: This page helps the user to scan the QR code for storing data into the google calendar
* bootstrap.html: This is a bootstrap file where all the pre-compiled files are stored that enables the user to quick usage in any web project.

**Back-end (Python)**

* Google.py: This file takes in the date input data date, moreover, it also analyses credentials tokens (auth). The python script is run as the backend of the project.
* main.py: This is the main file of the project. It contains all the functions that the system will execute. All the API’s are run in this file. Moreover, the feedback mail is also sent via this set of code.

**4.2 Explain in detail how to install your programs and how to run them.**

**Steps to run the program:-**

1. cd to the app folder in the code
2. Run the following command in on the terminal in the app folder:-

$ gcloud init

$ gcloud app deploy app.yaml

# Conclusions

From this project, we have successfully completed building a scalable web application that can scan QR code from the user’s bill and store data into their google calendar. Before scanning the QR the user will have to authenticate thus by allowing access to the google calendar.

Future implementations: This idea can be implemented for medicine as well as groceries. As the user will just have to scan the QR code from the bill, all the product’s details will be stored in the user’s calendar via our system. In our personal opinion, this will be a very healthy option for the users to switch to as they will be avoiding consuming expired products.

There is a huge scope to this idea as there are many other features that can be added such as

* Personal reminder: The user can set the personal reminder according to their choice. Currently, the reminder is placed on the expiry date of the product, while the new system can allow the user to set customized reminders.
* Doctor-Patient-Portal: The scope of the project is vast, there is one feature specifically we are looking to add later on, is about adding a portal where the doctor can monitor and pay attention to the user easily via the web application. The doctor and patient can chat with each other and convey the problems remotely and without the spread of infection in the public.

# Individual contributions (optional)

**Akash Patel (ASUT ID:- 1219522499)**

I. Design

The whole architecture was actively discussed by all the team members. There were a number of small features that we had to implement so as to get the application running. I have majorly contributed to three of the features that were thoroughly used throughout the project. They are Google Authentication for sign in, Google calendar authentication, and cloud memory store using redis.

II. Implementation

In terms of implementation, I have participated in all the aspects of the project in one way or the other. I have set up one of the basic requirements of any application that is authentication. I have suggested using Google sign in instead of normal sign in as our application heavily relies on Google calendar. To integrate Google sign in, I have used Oauth 2.0 to access Google API in order to authorize credentials that identify the application to Google’s OAuth 2.0 server. The advantage of using OAuth in our application is that we can store user email and name in session in order to perform insertion and deletion in user Google calendar without need to worry about verification of user email addresses as it is already verified by Google’s OAuth 2.0 server. Another advantage of using OAuth in our application is that none of our backend API is exposed, we have used flask as backend and also for redirecting from one web page to another, so users cannot hit any of our application API(web pages) without logging in. I have also set up calendar authentication for our application in order to correctly set the reminder in the user's Google calendar. Apart from this, I have also set up a cloud memory store using redis. The reason for using a memory store is because our application also takes input from users for giving feedback on their medicines. Users can only give feedback for medicine which they have purchased and not any random feedback. So, this service requires too many database calls. So, to reduce the number of database calls and improve the performance of the system, I have set up cloud memory stores using redis. I have also made a feedback form page and a medicine review page. The feedback form page will fetch the medical history of the user and will show it to the user. The user can select the medicine from the drop down list and can submit the feedback. The feedback then will be sent to the doctor who had prescribed the medicine to the user using gmail API. Here I have used redis to store the medical history of the user in cache. Under the medicine review page users can review about the medicine from other users. This web page also makes heavy use of SQL which was one of the major reasons to use a cloud memory store in our project.

III. Testing In the testing phase

1. I have tested google authentication using different google accounts to log in.
2. Tested google calendar authentication using admin google account.
3. Tested the performance of the application using cloud memory store and checked whether the application does not make DB hit every time to fetch the record via logging.
4. Checked the correctness of feedback form via concurrent request.

**Disha Bhukte (ASU ID:- 1219495225)**

I. Design

Among the various modules in the project, I was incharge of decoding QR code, deleting all the calendar events, setup and implementation of Cloud SQL and Google Cloud Storage and deployment of application in GAE. In the design phase, all the team members along with me had contributed to the finalizing of the services which we were going to use in the application. I was incharge of designing the database and qr code decoder for the application.

II. Implementation

I have contributed to all the aspects of both the frontend and backend of the application. In terms of implementation of our main module of updating the calendar according to the user’s medicine bill, I have built a QR code decoder in java and set up the Google Cloud Storage and used it in case multiple images are sent to the API. When the user would upload multiple files, the images were stored in the Google Cloud Storage. Once the image was stored in GCS, the API would fetch the image from GCS and decode the information from it using the jar which was designed and coded by me. The reason why it was necessary to make a separate jar for decoding the qr code image was that the pyzbar library which is used for decoding the qr code in python was not supported by google cloud services. This was a major blocker in our QR code decoding for which I had to work around the solution. After the image was decoded it was then utilized by the calendar reminder insert module for inserting the data into the user’s calendar. Now in case of deleting all the reminders from the calendar, the calendar id for all the medicine records of the user were fetched and the corresponding reminder having the calendar ids fetched previously were deleted. This was done if the user had a lot of reminders in his calendar and wanted to clean everything up in one go.

In terms of database design, I have designed and set up both the storage in Cloud SQL as well as Google Cloud Storage. The tables required for storing information for both the calendar reminder and feedback system were created by me in the google cloud SQL as well as I had also created a python code for fetching, editing and deleting the records of the Cloud SQL tables from our API. In terms of Google Cloud Storage, was storing the images in key value pair i.e. a unique id of the image with its value as the image in the storage bucket.

The deployment of the application was handled by me in GAE. I had created the app.yaml file for the application

III. Testing

I have actively contributed along with the other members towards the testing efforts of the project. On an individual basis, I have tested the modules that I have implemented. I had also contributed to testing the application with apache benchmark to show the scale in and scale out of the app instances.

**Sapan Desai (ASU ID:- 1219080070)**

I. Design

In the application I was the one in charge of setting up the designing QR code reader and decoder in javascript and the setup of the Google Calendar API and insert/delete reminders in it. Besides that I am responsible for making the feedback system smart by using the Google Dialog Flow. The designing process involved all the team members. However, I had designed the modules of which I was in charge.

II. Implementation

In terms of implementation, I have coded the frontend and the backend for the calendar insert module. I have set up the Google Calendar API for our project. When the user send a request to add all the medicines mentioned in the bill, the image is scanned using the QR Code scanner page created by me which has the image scanner as well as decoder in javascript coded by me. Then the request is sent to the API. The decoded QR code information is then fetched and inserted into the Cloud SQL and then it is inserted in the calendar of the user using the Google Calendar API. In case of deletion of a particular remider, the calendar id specific to that reminder was fetched and then deleted from the user’s Calendar. This was done in case the user does not need a particular entry in his calendar.

In terms of the feedback system, our requirement was that the user should be able to directly contact the doctor in case he has experienced a reaction or any other side effect of the medicine prescribed by the doctor. For this I proposed a chatbot in the project, which would smartly address users' concerns and send his feedback to the doctor. The best solution in this case was to use the Google Dialog Flow in our application. I set up the Google Dialog Flow agents, the needed intents for receiving user’s messages and responding to him. I have also added a functionality of directly emailing the doctor, informing him about the concerns that the user has with the medicine prescribed by him. So for example a user complains about any rash that he is experiencing after the medication, his complaint is directly sent to the user’s email.

III. Testing

For me I had tested the calendar reminder insertion and deletion functionality. It was a bit challenging in testing the chatbot as I had to handle the multiple possible test cases that would occur due to the user's response in the chatbot. I had to train the chatbot accordingly. Besides this, for testing the whole application I had created multiple QR codes in order to test the application and measure its accuracy when different QR codes are given to the application simultaneously. I had also participated in the testing of the application using the apache benchmark application.