ISY5005-INTELLIGENT SOFTWARE AGENTS

CareBot

A Companion You Deserve!

TEAM_8 : SANGAM

SANTHOSH KUMAR MOHAN (A0198528L)
GAUTHAM BALASUBRAMANIAM (A0198478B)
MERCEDES PREMALATHA RAMESH (A0198411A)

TABLE OF CONTENTS

1. Executive Summary

According to Singapore's Department of Statistics (**SingStat**), Singapore's total pregnancies accounted a total of 39,039 births in 2018. Childbirth typically occurs around 40 weeks from the start of the **last menstrual period (LMP)**. **The American College of Obstetricians and Gynaecologists (ACOG)** has clinical guidelines related to pregnancy which is followed worldwide. For a healthy pregnancy, a preconception care visit to a health care provider is recommended as per ACOG (International Standard). Prenatal visits to a health care provider usually include physical examination, weight checks, and providing a urine sample. Depending on the stage of the pregnancy, health care providers may also do blood tests and imaging tests, such as ultrasound exams. These visits also include discussions about the mother's health, the foetus' health, and any questions about the pregnancy. So, getting early and regular prenatal care is must. Prenatal care may include taking extra folic acid, avoiding drugs and alcohol, regular exercise, blood tests, and regular physical examinations. But a hard stop cannot always fulfil if there are questions that arise then and there. Knowing when to have check-ups, what nutrients to have, when and how much exercise is required etc. during pregnancy is important for every pregnant woman.

The primary aim of the project is to manage the check-up appointments for the user using Robotic Process Automation (RPA). Robotic Process Automation (or RPA) is an emerging form of business process automation technology based on the notion of metaphorical software robots (bots) or artificial intelligence (AI) workers. We have automated the process which schedules doctor appointments as per ACOG guidelines automatically with respect to LMP of the user. We have integrated the process with a chatbot for a seamless user interface. The chatbot also assists the pregnant women with answers for their doubts and queries.

2. Introduction

Although enterprise interest in bots seems to be at an all-time high, Gartner reports that 68% of customer service leaders believe bots and virtual assistants will become even more important in the next two years. In exponentially increasing development of technology, updating ourselves saving more time and effort helps in producing better work rate with less errors. RPA can augment repetitive, mundane, and error prone clerical task effectively minimising the manual tasks by expanding the virtual workforce quickly and easily. An efficient RPA agent does not require to replace the

existing systems, instead it leverages the existing systems the same way a human does.

2.1. Minimum Viable Product - CareBot

Every product in the industry was started as something much simpler than they are today. A Minimum Viable Product is a version of the product which includes only the features that allows to release it to market to solve a core problem for a set of users. The goal is to provide immediate value, quickly, while minimising development costs. We have developed an MVP which allows to get a basic version of our project with standard guidelines, intelligence and the core set of features rather than a full-blown, feature-heavy product.

2.2. Problem Overview

Pregnant women should mandatorily schedule appointments with a midwife or a doctor (Gynaecologist and Obstetrician) in a specific hospital or clinic as per their preference and ease of access. This must be done by filling a form for each appointment in the clinic's website. There will be multiple appointments which has to be appropriately done manually for each schedule. Say, if there are 12 appointments to be made, then 12 forms must be filled manually and submitted individually which is time-consuming and prone to errors. The user may also fail to keep track of the appointments.

2.3. Problem Solution

An application was built to automate the process of filling the form for scheduling appointments with clinic. This is invoked through a chatbot user interface. Access to the user's calendar was obtained and successfully booked appointments were saved as a calendar event in the user's calendar. In addition to this, the doctor's calendar is also maintained in a multi-agent environment and verified before confirming a doctor's appointment.

3. Market Research

Before developing a minimum viable product, a market research was done. There are many phone-centric pregnancy tracker applications that are very helpful in monitoring and keeping track of the pregnancy.

For example,

- Ovia Pregnancy & Baby Tracker
- What to Expect – Pregnancy and Baby
- Wonder Weeks

The above listed applications included the delivery date calculator and week-by-week pregnancy tracker guide. Every application had its own description and tracking schema. A study was done beforehand, and many applications were tested manually. They did not give reliable or unified results.

Many calendar applications are available, but they are of manual entries. These applications do not integrate with the doctors' calendar

Inferences:

No products had the doctor appointment scheduling and RPA integration.

3.1. Field Works

We discussed our MVP and its use case with doctors from India who are experts in gynaecology and obstetrics. The doctors with whom we had discussions were listed below.

1. Dr. Usha Balan, M.D. Gyn & Obs

   Usha Clinic, Madurai, India

2. Dr. Keerthana Subramaniam, MBBS

   Keerthi Hospital, Madurai, India

3. Dr. Chandrasekaran Seeniappan, M.D.

   Academic Tutor, Thirunelveli Medical College, India

4. Business Value

- The problem specified above has been there for decades but currently there is no reliable solution that covers all the above stated problems. So, this application has immense value as a commercial product.

Targeted Customers: Hospitals and fertility centres

End Users: Pregnant women and family

- Furthermore tie-ups with hospitals can be a source of revenue.
- Link with nutrient supplement vendors to promote their products.
- In addition to this relevant ads can be displayed.

5. Resources for the Application

A resource is reliable only if it provides content with strong evidence or well-reasoned theory. Collecting information by surfing the internet that has lots of articles and resources is unreliable as there is no way to determine the author's expertise. Many articles may notoriously provide false information.

To achieve reliability the following were used as the **resources for the application**:

- To explore the views, experiences and knowledge of the Doctors in this field, **Telephonic Discussion and Interview** were made. Few doctors from India who are specialised in gynaecology and obstetrics were interviewed beforehand to obtain a qualitative content. The content was also derived from the doctors through email and messages.

- Additionally, **ACOG and WHO** guidelines were followed. ACOG Endorses the International Confederation of Midwives Standards. The World Health Organization (WHO) is a specialized agency of the United Nations that is concerned with international public health. Both the standards were followed worldwide.

6. Technical Sophistication

According to ACOG, prenatal visits must be done as follows:

- Weeks 4 to 28: 1 prenatal visit per month
- Weeks 28 to 36: 1 prenatal visit every 2 weeks

- Weeks 36 to 40: 1 prenatal visit every week

• Based on the ACOG guideline, the check-up dates were calculated from the user's LMP and the bookings were made in the clinic's website.

• The appointment timings were checked with the doctor's calendar for availability slots. If the slot is available for that doctor, the appointment was successfully made.

• The successfully booked appointments were synchronised with the user's calendar so that they can keep in track of all the appointments.

• Also, if the user cannot meet up the schedule, the appointment can also be modified or cancelled upon user request.

• A knowledge base with FAQs was created from the interview with the doctor and from the WHO articles. The user can get answers for their concerns related to pregnancy.

7. Use Case

For testing purpose, original hospital systems cannot be used so the front end and back end of NUHS websites for creating, modifying and cancelling appointments were mocked up. This website was mocked up using Django and hosted on Heroku. The website was integrated with SQLite (light version testing) database when run on localhost and with PostgreSQL (production) database when run on Heroku. The records of appointments booked were stored in the back-end so that they can be modified in future if required.

8. High Level Design

The high level design is explained below:

• User interacts with Google Dialogflow chatbot to ask queries or perform certain tasks like booking appointments.

• Dialogflow interacts with Google authentication system to access the users' calendar and information to manage the calendar.

• Dialogflow is also connected with the RPA agents to invoke the respective RPA agent based on the user requested tasks

• The RPA agent interacts with existing hospital systems and books or modifies appointments from their respective website, just as a user would do.

• The hospital system implements a multi agent calendar management system in the backend to co-ordinate the availability of doctors using google calendar.

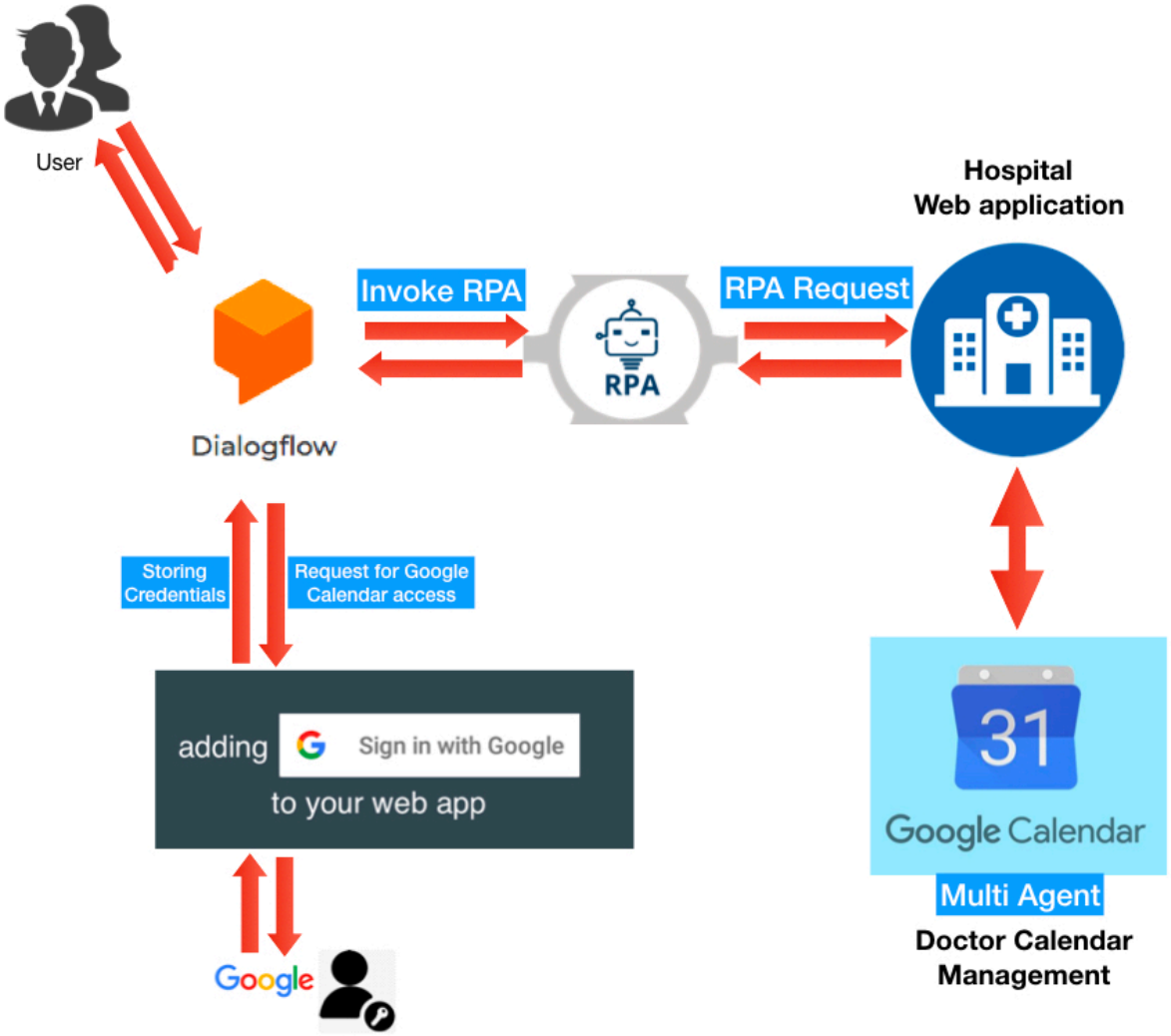


Fig.1. High Level Design

9. System Design

- The user interacts with our chatbot built on Dialogflow through Google Assistant

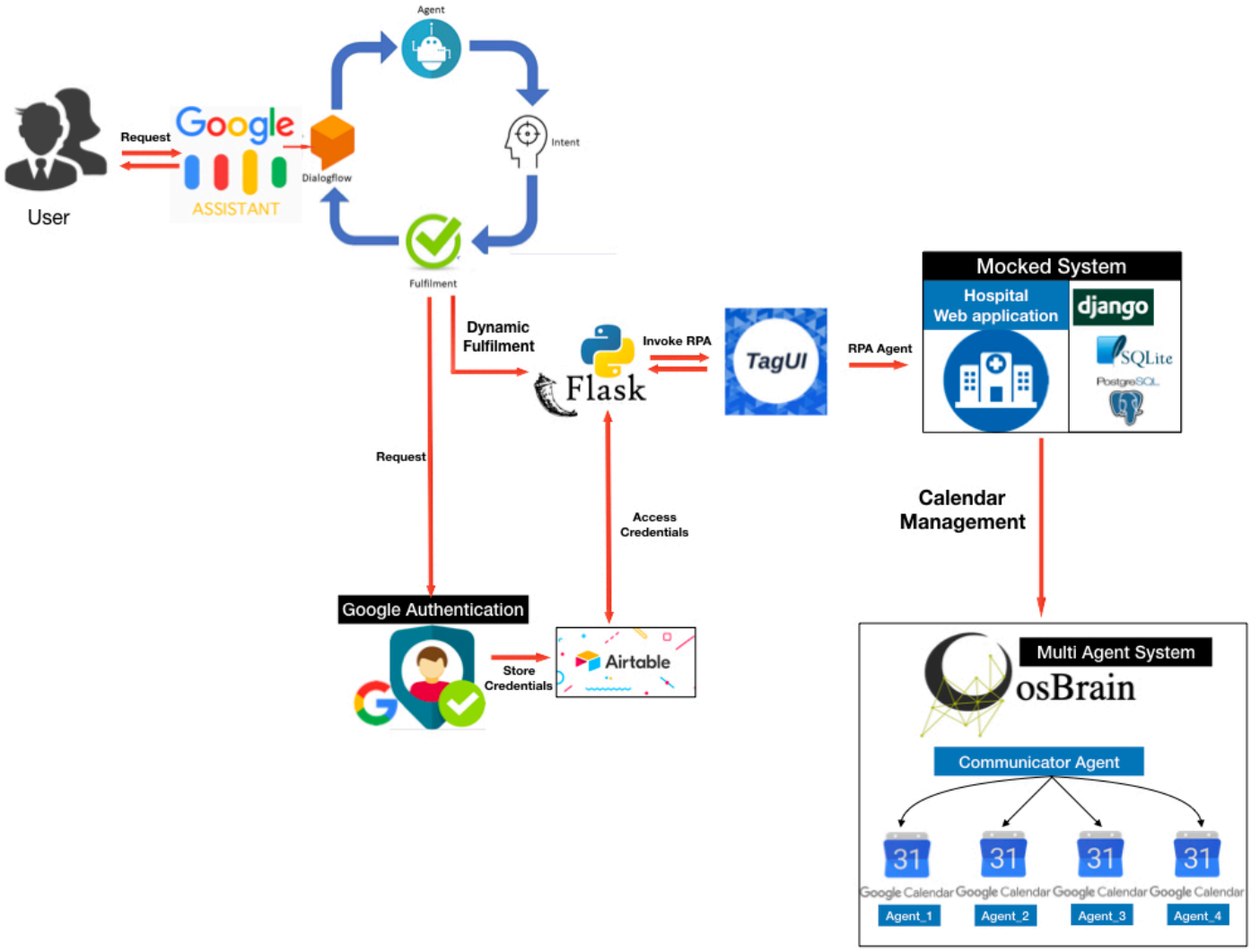


Fig.2. System Design

- Google Assistant send a http request to Dialogflow to satisfy the user request
- Dialogflow handles the natural language understanding part in human-computer interaction and sends the request to the appropriate agent
- The Dialogflow agent identifies the intent of the user
- Attributes are extracted from the identified intent and sent to fulfilment to a python-flask application hosted in Heroku
- Dialogflow is also connected to Google Authentication system to authenticate access for users' calendar
- The obtained credentials are saved in Airtable and used by the Flask application

- Based on the user request, the flask application calls respective RPA agent developed using TagUI
- The TagUI agent connects with the hospital website (Mocked up in this case) developed using Django, PostgreSQL and SQLite and hosted in Heroku
- This application also incorporates a Multi Agent Calendar Management system to manage the appointments of doctors using google calendar.
- After processing the user request, a fulfilment response text is sent to Dialogflow from the Flask application
- Dialogflow converts text to speech and sends the response back to Google Assistant
- The response for the user's question is then given both via text and voice by the Google Assistant

10. Chatbot

Goal : User Interface to access the automated process of scheduling the appointments and to interact with the user to answer the queries related to pregnancy.

Channel : Google Assistant helps in handling the speech to text conversion and text to speech conversion.

Platform : Google Dialogflow was built-in with intents and training phrases to handle the natural language understanding. Google Dialogflow was integrated with Google Assistant for user convenience.

10.1. Process Flow

The user either asks knowledge base questions or activate the RPA process through dialogflow.

External intent fulfilment was enabled to return dynamic responses through the web application. The above stated web application was developed using python-flask framework. To have a tight integration with google assistant all the responses including suggestion Chips, Lists and Text-to-Speech outputs were tailored into the response yet retaining support for basic text conversation. The output context of Dialogflow was altered

from Webhook so as to maintain a coherent conversation regarding a particular attraction and handle further queries.

To automate and activate the appropriate RPA, the request from the Dialogflow helps to get the basic input from the user like the kind of action to take(make, modify or cancel an appointment) and basic information like mobile number, preferred time to visit, preferred doctor.

11. Robotic Process Automation (RPA)

Analysis : The life cycle in RPA begins with Analysis. An appropriate study was made, tools to implement RPA were selected and the workflow was designed.

Bot Development : The RPA bot was developed using Python and TagUI

Testing : After development, the bot was tested multiple times to ensure reliability and logics. Testing repetitively helped in minimising bugs.

Support & Maintenance : After development and testing, it must be maintained and every change must be updated

12. System Workflow

Periodic prescribed check-up during pregnancy is compulsory for a healthy pregnancy. There will be nearly 15 check-ups during the entire term. Booking of appointments for the prescribed dates is a tedious, repetitive and time-consuming process. Automating this process using RPA will be most effective.

In this project RPA is implemented using TagUI. TagUI is a command-line tool for digital process automation. It is maintained by AI Singapore, a government-funded initiative to build local artificial intelligence capabilities. TagUI has built-in integration with Python (works out of the box for both v2 & v3). Hence TagUI is used as the base tool to develop RPA. The user can demand to book appointments or modify or cancel the existing appointments. The following three processes were automated using RPA in this project.

- Make Appointment
- Change Appointment
- Cancel Appointment

12.1. Make Appointment

When the user requests to make appointment through Google Assistant, the Dialogflow uses slot filling mechanism to get the details like Name, Phone number, LMP date, preferred timing to visit and preferred Doctor. From the LMP date, the check-up dates were calculated asper ACOG guidelines. The process to book appointments repetitively for all calculated check-up dates is automated using RPA. The availability of the slots from the doctor's calendar will be checked before booking the appointment. Once the appointment is successfully placed in the website, the respective doctor's calendar will also be blocked for that specific slot.

12.2. Cancel Appointments

When the user requests to cancel the existing appointment, the required details will be obtained using dialogflow and sent to the flask backend. The respective appointment will be cancelled in the hospital website using RPA. In addition to this, the calendar slot for this appointment will be removed and made free.

12.3. Change Appointments

When the user requests to change the existing appointment, the required details will be obtained using dialogflow and sent to the flask backend. The requested change of date will be applied in the hospital website using RPA. The calendar availability of the doctor for the old date will be cancelled and made free in addition to blocking the doctor's calendar for the new date.

13. Calendar Agent

Multi-Agent calendar management system is implemented to manage and schedule checkup appointments on behalf of doctors. It is considered as a distributed task where each agent knows the calendar availability and preferences of respective doctor. Python osBrain, a general-purpose multi-agent system module was used for this scenario.

Google calendar was used to maintain the preferences and schedules of the doctor. Google credentials of the doctor is available only to the agent

which maintains the calendar and only office hours have been accessed. Thus, the privacy of the doctor is upheld.

Actions such as make appointments, cancel appointments and get appointment details have been implemented for each agent.

A communicator agent is also implemented to co-ordinate agent and help in inter-agent communication.

13.1. Communication between agents

OsBrain uses ZeroMQ were used as a messaging system for flexible message passing between agents. ZeroMQ is a high-performance asynchronous messaging library to use in distributed or concurrent systems.

Json message format is used on top of the ZeroMQ message queue for easier transmission of complex messages and to co-ordinate with google calendar APIs. Request-Reply and Pub-Sub patterns are used as mode of communications.

14. Challenges Faced

- Google Dialogflow has been used to implement the chatbot interface. It imposes hard timeout of 5 seconds to send back a fulfilment response. But, the form filling and submission for multiple iterations which is automated by RPA takes few minutes depending on the time required to load the webpage and the number of bookings. To overcome this setback, the RPA process was carried out asynchronously and a fulfilment response is sent once the RPA process is invoked without waiting for its result. So, the status of appointments booked could not be directly given to the user in the conversation as the context could not be hold in the backend after sending the fulfilment response. So the status is saved in the backend database and will be presented to the user after few minutes through a separate intent when the user asks for the appointment status.

- Integrating google calendar with google Dialogflow and OsBrain system was a major challenge, since direct access of calendar from chatbot is restricted by google. A separate web-app was created to initiate a hybrid OAuth authentication flow so that user's google calendar can be used to create calendar events.

- Attempt was made to build the chatbot as pervasive as possible but there were complexities in matching the correct intent for the ambiguous questions.

- Testing the application in a real hospital system was not possible, as a work-around the front end and backend system of a real hospital was mocked up and hosted along with database to store the appointments which required a lot of time and efforts.

15. Limitations

- In case of complications in pregnancy the bot cannot handle certain circumstances like miscarriage, abnormal health issues like diabetes, pressure etc. which requires hospital visits frequently. The track of health of the user is not maintained.

- Failure of applications due to unavailability of doctors will be updated to the users. However, the system will not be able to retain the context and automatically rebook the appointments with change od date and time. Currently this must be done manually by the user, but this is one of the future scopes of this project.

- A study and a systematic effort were made in designing a system which includes **Machine learning algorithms** namely Decision Tree, SVM and Naive Bayes to predict diabetes at early stages of pregnancy. Due to time constraints, the integration of this system was unsuccessful. Further study was also required before implementation.

- Utilization of RPA is gradually increasing in various market worldwide which comes with its own share of challenges. With a new and innovative technologies, many factors in the design and execution of an RPA solution needs to pay attention to a few crucial aspects and update them then and there to increase the success rate. Though the websites do not change frequently, if the website is updated, then the RPA should also be altered.

16. Future Enhancements

In addition to the developed features, the following improvements are part of the roadmap for future:

- Developing a complete application end-to-end which includes:
    - Chatbot and RPA activation as per requirement

- Calendar and remainders
- Track of user health including health complications and solution and records of the health details like scans, medications etc.
- Nutrients, diets and exercises
- Online sale of nutrient supplements and medications
- FAQs

- The appointments booking and integration was implemented with the mocked website of NUHS-The National University Health System, Singapore. In future, the integrations can be done with multiple institutions and clinics so that wide number of users can be benefited.
- Implementing Machine learning algorithms to predict and detect abnormalities in health conditions like high/low blood pressure, diabetes, anaemia etc.
- Hold the context to automatically rebook failed appointments.

17. Conclusion

Minimum Viable Product – CareBot is a hard-headed implementation that can be commercially employed. Building an MVP is the essential part of our app development which is an initial point in the thoroughfare to accomplish a product. Application of our project helps wide range of pregnant women of various traits as we have followed International Standard. The built project can be extended worldwide with minimum alterations only in the process and workflow of RPA, as the Healthcare provider, their websites and functionalities changes for every hospital or clinic. Many fields are in the intermediate phase to understand the benefits of an RPA. This technologies and capabilities continue to increase, and the best complete product is yet to be created. This opportunity stimulated our thinking and consolidated our learning to implement our knowledge practically. We laid a promising path to add loads of feature for a successful start-up with this MVP.

Appendix 1: Installation Guide

Download the source code from GitHub repository

- https://github.com/sangam-iss/ISA-IPA-2019-10-30-IS01FT-SANGAM-CAREBOT.git
- (should not have space in the file-path, Tag-UI has been used)

Python Flask Setup:

1. Open folder ISA-IPA-2019-10-30-IS01FT-SANGAM-CAREBOT\system_code\fulfillment_with_rpa_system

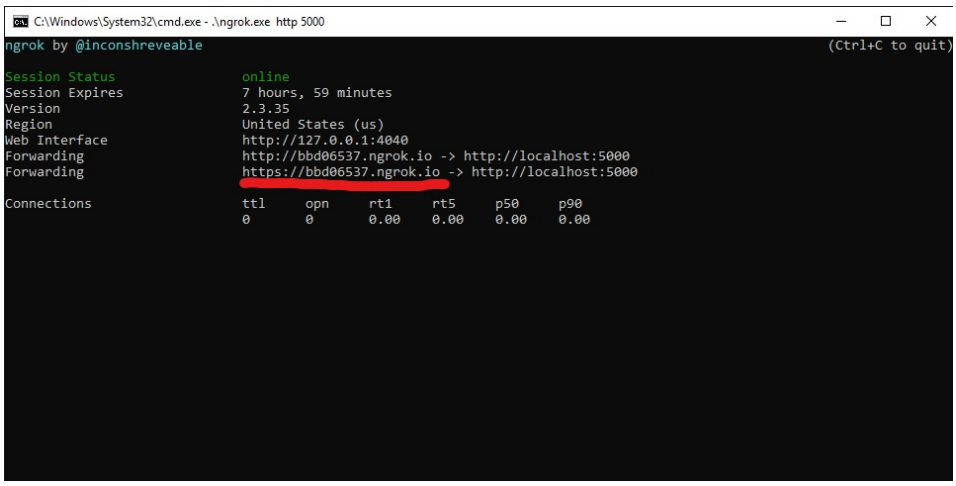
2. Copy the given credentials.json file into the above folder

3. Create Conda Environment:
    a. Conda create -n “sangam-ipa” python=3.6 ipython
    b. Conda activate sangam-ipa
    c. Pip install -r requirements.txt

4. Run the flask app (python app.py)

NGROK Setup:

1. NGROK is used to connect the local hosted flask app to the Dialogflow .
2. Download NGROK for windows:
    - https://ngrok.com/download
3. Unzip the downloaded NGROK and run the following command prompt:
    - Ngrok.exe http 5000
4. Copy the link shown below. This link needs to be used in dialogFlow later.

- 

DialogFlow Setup:

1. Go to URL https://dialogflow.cloud.google.com/#/login
2. Login using google credentials

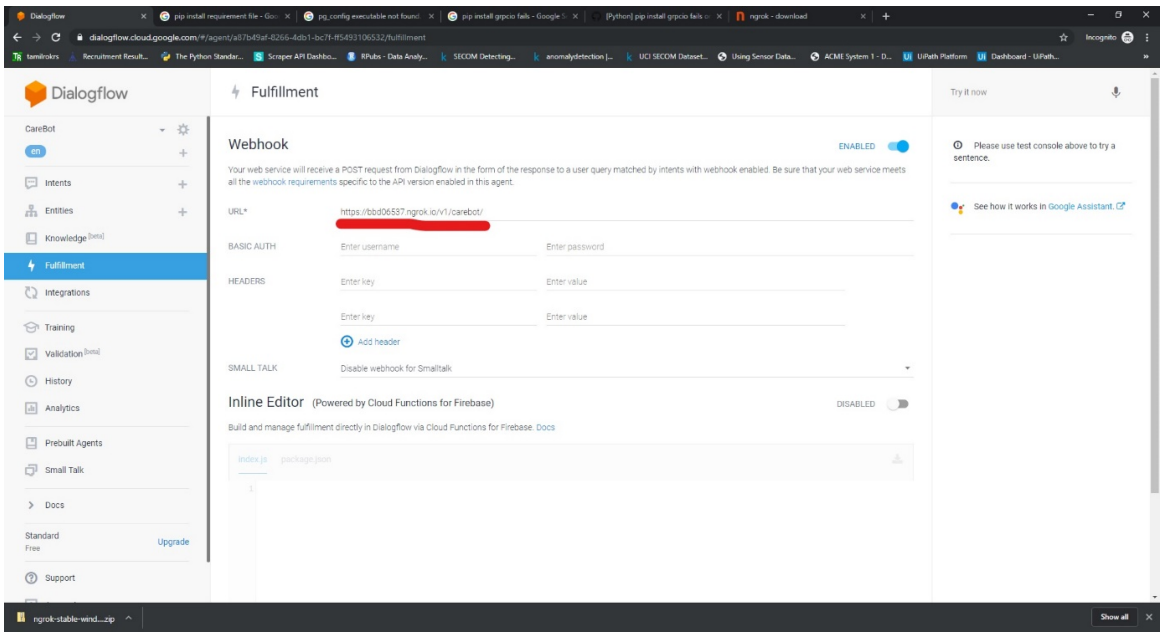
3. Follow the given video and create a DialogFlow agent with carebot.zip downloaded from github

    - https://www.youtube.com/watch?v=ii2woEMJHQQ&feature=youtu.be

4. Delete the “Default Welcome Intent" in Intents

5. Go to Fulfilment Tab in DialogFlow and enable fulfilment

6. Paste the copied NGROK link in the URL of fulfilment as follows:
7. NGORK_URL/v1/carebot



8. **Account Linking and Voice Invocation:**

   Follow the below video for google account linking and voice invocation

   - https://www.youtube.com/watch?v=3inKuZ2M6Pw&feature=youtu.be

9. **Email Permissions**: App needs Email permission so that we can identify and manage appointment based on email. System asks for email for every fresh user. Follow the below link to give permission to access email.

   - https://www.youtube.com/watch?v=OeaqJw-Y_QY&feature=youtu.be

10. **Manage the users' calendar events**: For the system to manage the user's calendar, the user should ask "Manage my calendar" so that the system will prompt the user to authorize this task and get authorization token to create modify and delete events from the users' calendar in future tasks. This is a one-time process.

    Follow the below video to initiate the system and give necessary permissions to the chatbot

    - https://www.youtube.com/watch?v=Hz_u2tYd8Ak&feature=youtu.be

Appendix 2: User Guide

The user interface for using our application is a self-intuitive chatbot implemented using Dialog-flow.

To initiate conversation with the chatbot the user has to say, ***"Talk to care test app"*** (based on the invocation name) from their google assistant which will give a welcome message to the user.

Initially when the user first uses our application, along with welcome message the chatbot will ask the user to allow access to his details such as email id to automatically fill up forms at later stage.

Some examples to achieve the implemented functionalities are given below

Get answers for frequently asked questions: At any point of time the users may ask their questions related to pregnancy in a simple conversational manner and get the relevant answers from the chatbot. For example, ***"What is LMP?"***, ***"Tell more about calcium intake during pregnancy"***, ***"What happens in week 8?"***.

Book appointment with the hospital system based on LMP date: The user can simply say ***"Book doctor appointments"***. The chatbot will gather the required information through simple questions and book the appointments in the backend using RPA.

Reference link: https://youtu.be/U0sLAGY3llE

Check whether the appointments are booked successfully: Few minutes after instructing the chatbot to book appointments the use can come back and anytime and ask, ***"Get me the status of my appointments"***. Based on the details collected before, the system will get the success status of the applications and if there are any failed appointments, it will also mention the dates for which the appointment failed due to doctor unavailability.

Get upcoming appointments for the user: At any point of time, the user can ask the upcoming appointments by asking ***"Get my upcoming appointments"***. Based on the user details collected before, the system will fetch the upcoming appointments for that user.

Reference link: https://youtu.be/8L4chkHqBNc

Change existing appointment in the hospital system: The user can say *"Change my appointment"*. Then the system will gather the required information and change the specified appointment in the hospital system using RPA.

Reference link: https://youtu.be/wHAfC7zOshA

Cancel existing appointment in the hospital system: The user can say *"Cancel my appointment"*. Then the system will gather the required information and cancel the specified appointment in the hospital system using RPA.

Reference link: https://youtu.be/reieAKDeH-c

Appendix 3: Individual Contributions

Individual Report: Santhosh Kumar Mohan, A0198528L

personal contribution to group project:

As a team, all three were involved in Requirement gathering, research regarding the product, architecture design, use case designing, validation and testing, and creating project report. Apart from these tasks, I was involved the implementation of the below tasks:

1. Implementation of the multi-agent system for doctor calendar management.
2. Developing the google authentication workflow for asking permissions from users for email and calendar access.
3. Integration of google calendar API with DialogFlow to maintain user's calendar by booking, changing, and cancelling their appointments in google calendar.

what I have learnt:

Working on this project opened a lot of interesting train of thoughts for me. Especially on how to analyse the market and how to properly plan for a project commercialization. Approaching a problem and a viable solution in terms of implementing the minimum viable product taught me how to incrementally develop a product keeping the business angle in mind. Also, I learnt the scope and possible usage of RPA processes and how to think about RPA solutions and effectively implement them. I also learnt about multi-agent systems, how to develop them and where these kinds of solutions would be useful. Implementation of multi-agent system also taught me about inter process communications and how we can user them for developing effective complex systems.

how you can apply the knowledge and skills in other situations:

In the current market scenario, RPA is one the key skills that could be used in any situation to replace mundane tasks done by employees. Pairing RPA with multi-agent system opens new opportunities and designs that can be explored to make RPA even more efficient.

Also, thinking about a project in terms of MVP is useful in a situation where a project must be started from scratch. Leading a team in either a start-up or a big company, MVP approach would be useful in attaining the goals of the project faster since it requires concise planning and resource allocation before even starting the project.

Individual Report – Gautham Balasubramanian – A0198479B

Personal contribution to the group project:

- Actively took part in brainstorming for selecting a problem that has a huge socio-economic impact
- Took part in designing technology-oriented design to solve the problem that was selected
- Developed the mock website for RPA using Django Framework and connected it with SQLite Database for storing the appointments
- Developed front-end for the website to replicate the form filling process of real-world systems
- Hosted the Django app in Heroku cloud for round the clock availability and integrated PostgreSQL Database to replicate a production environment
- Developed Dialog-flow intents for the chatbot to address the user requests
- Created effective training phrases to effectively capture user interests and developed slot filling mechanism to capture the required attributes from user requests
- Developed the Flask backend for dynamic fulfilment of Dialog-Flow requests using webhook
- Took part in testing the application and project report writing
- Proactively lent helping hand wherever required

Things that I learnt:

First and foremost, I have learnt the importance of developing a Minimum Viable Product for any real-life ideas around us. I also understood the scope of disruption that RPA could bring to any industries thereby improving the currently followed manual process. I also understood the difficulties posed by unstructured data which is the most prevailing form of data as of now. Technically I have learnt to build chatbots using Dialog-flow which is one of the most sought skillsets in the industry. I have also learnt to develop and host web applications in public clouds which could help me to become a T - shaped expert. In addition to that I have also learnt implementation of multi agent systems and RPA application development from my peers in this project.

Where can I apply these?

I believe that the skills I have acquired from this project would help me in most of my future endeavours. Most of the process followed in the current industry are repetitive, manual and time consuming. I believe I can automate frequent repetitive tasks to improve the work efficiency. Gaining knowledge in web app development would help me to present Data Science POCs to clients in an impactful manner. Finally, chatbots are transforming from the state of sophistication to the state of necessary in Industry 4.0 which creates a huge demand for this skillset. I believe I could fit in to those tasks and develop them with ease with the experience gained from this project

Individual Report - Mercedes Premalatha Ramesh – A0198411A

1.Personal contribution to group project.

As a team we had many discussions beforehand and designed the basic flow and architecture.

Individually,

took few interviews with the doctors from India before starting the process.

I also did few research as per Experts' ideas.

Tested many apps in the industry before automating the workflow(Market Research).

Developed RPA using Python and TagUI.

Integrated RPA with Flask.

Did knowledge base integration for FAQ in dialogfow.

Testing the application workflow

Project Report preparation

Marketing Video preparation

2.what you have learnt?

By developing a project from scratch using MVP, I understood the importance of analysing the market and understanding the business model before implementation. I also understood the impact created by time to market. I developed my skills in implementing RPA solutions using multiple tools. I learnt to create knowledge base that could be readily consumed by chatbot frameworks to develop conversational interface for FAQs.

3.How you can apply the knowledge?

I underwent a situation where my sister was undergoing a helpless and confused state while she was pregnant. I felt that something was lacking in the industry in this field. On discussion with the teammates it seemed that they also faced similar situation with their family members. So we ended up in developing a MVP which assists pregnant women. We can also be benefitted from that product commercially. In the IPA module coursework

we had a briefing on developing an MVP. So, this gave us a clear solution to develop this project. I will definitely be developing this project furthermore.