Virtual Doctor-

A web app for instant patient diagnosis!

<https://marutitech.com/build-a-chatbot-using-dialogflow/>

Wotnot is an Alternative Development application stack that can be used for building the chatbot frameworks.

Advantages-

* Pregnant woman
* Unborn child
* HCP
* Healthcare Personnel
* Community and Society
* Health of the baby and coming generations

IT-

* AI Anticipate
* Nausea and Nutrition
* Behavioral Changes
* The time structure healthcare industry to be able to optimize value resource
* Patient Contact time for repetitive info can be reduced for providing more opportunities for diagnosis and management
* Improves emotional quotient of Doctor and patient and drives away redundancy.

Topics covered -

* Information technology in healthcare
* Deep Learning and Information.
* Communication in Healthcare
* Ethics in Healthcare.

Ethics fillers- Code of conduct, 4 pillars, autonomy, maleficence-No harm justice-caste- religion no biasing, beneficence- , MCI Oath taking. (Extensions-> Informed consent and debriefing.

For Midsem -

Some of the other projects of this domain that are discussed are pen that writes and sends data to cloud, x ray image scanning, Suggestions of treatments to doctors for treatment of COVID, Disease Detection, classify skin cancer, Diabetic retinal screening, Lifestyle improvement suggestions, NLP to document and write a medical diagnosis from conversation b/w doctor and patient.

Objective

This project objective is to improve doctor patient communication to enhance patient awareness to directly impact positive behavioral change to improve health

Use chatbot (static-not storing information) -> Dialogflow -> whatsapp integration with twillo used for sending whatsapp text, msgs, calls etc. 500 mb traffic -> website integration. i/p predefined answers->contact doctor/book an appointment.

1. NLP Model if need to store chatbot data will have to use firebase.

2. Backend Django- Routing Service SQL Lite inherit Database Sequential database.

Store Userlogin, id-password, current-status, address, login-page.

Predefined Doctor- Timestamp, store data in database.

Making table (4-week max) ->checkup admin login to doctor (doctor can put up that he wants to setup for patient.)

DialogFlow

1.can store variables in it

1. Chatbot month – {1-2 vomits),{5-9}, (3-4 vomits}

2. States-1 Intent leads to another

3. 1 Condition

1. months
2. frequency
3. 1day (or) consecutive days->contact doctor (or) appointment

* Tech stack
* Abnormal behavior
* Normal behavior
* Remedy

First send and create workflow then ask for data entries. Then decide SQL Lite or non-SQL Lite database. Then Decide Tech-Stack.

Creating a website chatbot that will connect all the stakeholders in the value chain of doctor-patient communication and ease the entire process by improving the emotional quotient of Doctor and patient and drives away redundancy directly impact positive behavioral change and improve health.

Project Timeline-

The Deadline for the Project is (3-4) months-

* Prototyping, Designing, Wireframes/Mockups
* Development
* Testing
* Deployment and Optimization

**Initial Planning-** So To Make Our Idea A Reality The First Step Conatins Initial Planning. A Clear & Transparent Discussion For Preparing The Roadmap For Our Idea Right From It's Development Process To It's Optimization Process

**Wireframing and Mockups-** The next step is to design the application and provide mockups to see how the idea looks in practicality. The design focus is on to meet intuitivity, simplicity and convey the idea minimalistically.

**Development-** Besides design the most important parameter is the technology that should be used so that the product has scalability and reliability.

**Deployment/Optimization-** The next stage is testing along with to scale the code by following proper linting and version control so that there is no issue reading/changing it.

**Development Stages**

* Basic functionality of the application added.
* Frontend and backend architecture synced.
* Code linting & Development testing.
* Collecting the Dataset
* Pre-processing the dataset
* Training the model
* Testing the model
* Hyperparameter tuning of the model
* Checking the underfitting / overfitting of the data

\* Identify the top 5 to 10 items that customers want to know about. Limiting the topics allows for finer tuning of the bot to understand the subject better which in turn allows answers with higher confidence.

\* Construct Frequently Asked Questions responses (FAQs). These are normally 10 to 25 topics and answers with short text responses along with follow up links as needed which contains additional details and information. However, note that the chat medium demands a more conversational approach, so make sure responses are less regurgitation of long explanations, and more chat.

\* Match the chatbot to your existing voice self-service platform capabilities. Recreating well-defined and understood topics creates consistency across all modalities for the customers.

\* Identify the transactional use cases beyond merely informational ones. This would be for creating a more fully functional bot.

\* No matter which direction is decided upon to get started, it is important that the bot-customer transcript be provided to the agent, so they have context to help the customer and provide a higher touch experience.

**Chatbot channels-**

Chatbots are about a continuous conversation allowing for any number of turns taken between the bot and the user. They are story- or flowbased, where all previous interactions are always visible to both interlocutors, and linguistic constructs such as pronouns (“and what is the price of that”) or ellipses (incomplete sentences such as “yes, but on the aisle, please”) require you to maintain context so that that messages can be resolved using information previously provided. What this means for your chatbot design is that user messages can never be analyzed in complete isolation – they are part of a conversation. As a first design step, you therefore want to create a Conversational Architecture. This is similar to an Information Architecture for a GUI, which puts the website content into a hierarchy of web pages (a site map). If the basis for your bot is an existing set of answers to FAQs, typically exposed on your website today, you would take this set of answers and consider which follow-up questions a user might have based on a particular bot response. Without knowing that we are now in the context, the bot cannot provide an answer to this question. The Conversational Architecture tells the bot developer which context to consider in which step of the overall dialog.

**Design DialogFlow and StoryBoard**

In the dialog flow design, you will want to show representations of what the bot will say at each step. However, the detailed message design should happen outside of the flow diagrams, as you will want to design variations of the same message for frequently occurring dialog steps. This is a technique called random prompting, where you make the bot use wording variations to essentially say the same thing. This will make the whole experience less robotic and more human-like, something we should strive for in bot design.

**Chat Data-Collection**

One of the most important resources you will need is a collection of sentences reflecting different ways to express each one of the intents and slot values your bot will need to recognize. This resource can be difficult to create because of the amazing diversity of human language – and the limitations of the developer’s imagination. The ideal resource is actual text from conversations with customers. If you need to create your initial data by hand, make certain that your collection includes not just one developer’s intuitions but also input from as large a variety of people. Remember that diversity is the key. One useful approach is therefore to work with QA and crowdsourcing companies who have access to large number of people from around the world. Finding people with the same linguistic background as your target users is helpful.

**Platform and Development Approach**

Chatbots consider the key tasks to be performed on natural language sentences to be: (1) to determine the intent of the sentence (2) to extract data from the sentence (what options has the customer requested? What data is he providing to you?. There are essentially two different approaches to these tasks: one based on explicitly creating rules from the top down, and one using machine learning algorithms to learn the task from a large corpus (a collection of written texts) of transcribed interactions.

An advantage of the linguistic top-down approach is that you have full control over how a message is understood. A neural network created by a machine learning algorithm is often a black box that doesn't let you go in and surgically change how one particular message is understood. Nuances in natural language – such as the fact that “I want to transfer my data” and “How do I move my files?” are about the same intent, but “How do I move this to my file?” is from a different one – are hard to learn with machine learning, but easier to distinguish with a linguistics-based platform

**Implementation**

It all comes together in this step: the conversational architecture, the dialogue flow and storyboard, the platform you have selected, and the data you have collected. Your essential task is to use these to create a classifier that will map an incoming text to the system’s response. If you selected a platform based on machine learning, you will provide this platform with your example sentences for each possible intent. The more examples you provide, the better the algorithm will learn the variations of linguistic expressions that can be used for each intent, and the better it will learn how to distinguish between intents. Note that you will want to reserve some of your example sentences for the next step (testing). If you are working with a linguistic rules-based platform, you will use the sentences in a different way. The rules you craft will explicitly represent the characteristics that determine that a given sentence belongs to intent A or intent B, leveraging the tools and abstractions mentioned earlier. In either situation, this is when it is centrally important to have a diverse set of examples as close to real user utterances as possible.

**Deployment and revision**

Even if your bot employs unsupervised or semi-supervised learning to adjust its own behavior over time, monitoring the first interactions with real users will yield very useful information and may signal that explicit adjustments should be made. Typical adjustments are in the wording of your bot’s responses, as they might yield follow-up clarification questions by your customers that wouldn’t be necessary if the bot’s answer were clearer. You may need to adjust the logic of your intent classification, either through explicit manipulation of the rules or through providing more example sentences. Finally, you may need to add new use cases if the designed use cases do not cover the majority of user requests. If you truly started small as recommended, then this is the time when you are collecting the vital information about which use cases are the key ones to cover.

**Chatbot Parameters**

1. The Intent: An Intent is the intention of the user when he/she interacts with the chatbot. What does he want from the chatbot/ What is his intention? As businesses consider how best to effectively use the new Bot technologies for their business, they need to identify the area that can be automated or improved by AI solutions. Artificial Intelligence solutions can be classified based on two criteria:  
a. Work complexity  
b. Data complexity

2. Designing a chatbot conversation: For a chatbot to be useful and evolve, engaging with the consumer and business is the key. Therefore, the chatbot conversation should be well-defined and can be segmented into structured and unstructured interactions.  
a. Structured: The structured conversation is more defined and follows a set pattern. There is a logical flow of information  
b. Unstructured: The conversation flow is more random or freestyle, just like you have a conversation with a friend or family

3. NLP/Deep Learning-Machine learning is an algorithm that helps the bot to solve problems that are too difficult. They learn from previous queries or data. When a query is triggered, the machine learning system before giving a response checks the past conversations it had with the user. The performance of the machine learning systems can be enhanced by exposing the algorithm to more data. 4. Natural Language Processing (NLP) provides assistance to the bot to communicate with the user. It helps you understand and interpret the information to deliver a new level of experience desired by companies and customers. With NLP, the chatbots are trained to understand the intent of the user from the conversation, based on which they can respond to the query.

**Chatbot Development steps**

**\*\*Step 1: Define User Experience\*\***

**\*\*Step 2: Define the Devices and channels\*\***

**\*\*Step 3: Gather the usage data from analytics\*\***

Analytics is the better data to tell you what your users are looking for in your applications.

Get all the past data to figure out all the keywords your user use. If you already have a live chat or customer service systems try and see all the historical data.

Use this data to design your features for the chatbot. And use the same data to train your bot if it is structured.

You can also get more data from social trends or public dataset for your niche and use that.

**\*\*Step 4: Pick the right NLU (chatbot engine)\*\***

Once you know the exact feature set you need to implement in the chatbot. Pick the right chatbot engine that supports that.

**\*\*Step 5: Design your backend services (For automation) \*\***

Chatbot should be used for automating process end to end.

There are many functions that can be automated using the chatbots. So, the chatbots should be able to accommodate the end-to-end automation.

**\*\*Step 6: Design Contexts for conversations\*\***

The primary goal of chatbots is building conversational applications. Every conversation should be tagged with some context and the bot should be able to understand and switch between contexts elegantly.

If the bot sounds or feels like a machine, the users are not going to interact much with it.

It should be fun and interesting for users to continue to use chatbots.

**\*\*Step 7: Input all the data in the right format (entities, intent) \*\***

Once all the design is done, then you have all the information to build your bot. Most of the chatbot engines have these most basic elements.

Entities- For defining data and keywords

Intents - The user intent classifications based on the questions or sentences.

Use the data collected earlier to enter intents and entities.

**\*\*Step 8: Train & Test\*\***

Once the data is entered. Train it.

If you are using custom chatbots, then train it by altering the parameters until you see good results. Test. Many NLU engines are built for training with data. They also learn as we keep hitting it hard. Test it as much you could and make sure the behavior is consistent. If you see inconsistency, then it means there is a problem with data or training.

**\*\*Step 9: Implement a good analytics system.\*\***

You need analytics system to measure every single event in your chatbot.

Without knowing what users are experiencing its hard to optimize the chat.

**\*\*Step 10: User Testing & optimize it.\*\***

The last step is to have users test it.

Use the data from analytics to understand how people really use it and make sure the bot is optimized for it.

**Front-End Development:**

* Application Pages coding on DialogFlow.
* Creating as aesthetic UI for a better user experience.
* Coded as to increase SEO merit of the application.
* Creating features or interactive tools for the user.
* Upgrading the traditional look of the application and enhance/increase the CXM/retention rate.

**Back-end Development:**

* Structure Database
* Programming for the admin interface.
* Integrating features
* Integrating modules/Functionalities
* Fixing loopholes in the work, if any.

Chatbot designing comprises two form factors- front end look and backend functionality.

Whenever starting from scratch, we see to it that we build these two in sync to avoid discontinuity during the development process.

**Testing**

* Testing to ensure no errors in code and the application is generally bug free.
* Configuration and setup of the Client-server communication network.
* Updating content/information on the application to train the model further and get better accuracy results
* Documentation and commenting of code for future updation.

***Scope of Work***

Design Problems and develop strategy according to the problem statement given for solving the problems effectively and memorably. The tiered content approval levels will be established by

understanding the process and incorporating it into the application design.

● Determine the main goals of user pathways.

● Determine the user/application flow

● Prepare wireframes

● Optimise the NAV - Navigation [ (Site-map) + (navbar & footer link focus ->

user focused) ]

Using the insight gained from the analysis of the required content for the site, make designs to communicate the material, with minimal modification, to visually express the core message with intuitive navigation.

This schedule gives the major tasks to be completed during the life of the project. The times listed below are estimates.

**PROJECT TIMELINE SCHEDULE**

***Phase I: Strategy & Concept Development***

**Begin work on site map/architecture** 16 January

**Content / Images (photos/logos/etc.) Curated** 23 January

**Basic codebase repository created** 30 January

***Phase II: Application Design & Development***

**OAuth user module implemented** 7 February

**Application Functionality added(Type of input, intent added)** 14 February

**Product functionality added. (Type of Product, Type of input)** 21 February

**Application Functionality added(Type of input, objects created, workflow defined)** 28 February

**Application Functionality added(Type of input, Routing established, Different states established)** 7 March

*These Implementation will require code review-(Will have to look at the code for implementing these phases)*

***Phase III: Production Management & Implementation***

**Content migration to the website** 14 March

**Final Touch-ups of the Application, Implementing the nitty-gritty** 21 March

**Beta testing of the Application on production server** 28 March

**Integration into the Client side-website** 6 April

**Implementation of final design refinements** 13 April

**Final Project Completion** 20 April

**Technology Stack**- DialogFlow, NLP, Recommender System, Data Analytics, Deep Learning, LSTM, Twilio, DBMS, Firebase, Django, HTML, CSS, JavaScript

https://www.facebook.com/Lenest-Chatbot-106431041429565/?view\_public\_for=106431041429565

The Project is Divided into 4 Tasks-

|  |  |  |
| --- | --- | --- |
|  | Plan Of Task | Date |
| * Task1 | * Technical and Design Documentation | * 07/02/21 |
| * Task2 | * Wireframes and Mockups | * 07/03/21 |
| * Task3 | * Development | * 07/04/21 |
| * Task4 | * Testing | * 28/04/21 |