High Level Design Document

Project: VAM Insurance Bot

Angsuman Ray
Shradha Shankar

Version: 1.0 **Date:** 01/07/2016

Table of Contents

1. Introduction

1.1 Overview

Contacting the toll numbers provided by the insurance claims department may not always be the easiest way to file a claim for an insurance policy. These numbers may either be subjected to heavy traffic or may be unresponsive (a possible event during weekends and public holidays) and hence make the process of FNOL (First Notification of Loss) cumbersome and frustrating. VAMInsuranceBot is leveraged to address this concern. The bot captures the necessary details about the loss through a friendly conversation and gives the user the impression of an actual telephonic conversation with an officer from claims department through a chat.

1.2 Abbreviations

- PAN : Permanent Account Number
- AI : Artificial Intelligence
- ML : Machine Learning
- FNOL : First Notice of Loss
- BLOB: Binary Large Object

1.3 Requirements

The user is required to have a smart device with internet access. The device should contain at least one of the channels, currently Telegram, to which the bot is configured in order to converse with it. The user should also have a functional E-mail address which has to be registered with the insurance company at the time of buying a policy.

1.4 Constraints

VAMInsuranceBot is mainly designed to address the concern of notifying a claim to the insurance claims department. Hence, most of its functionality would appeal to existing policy holders who desire to file a claim. It provides minimal assistance to new users who wish to acquire a policy by providing them with the contact details of the insurance company. Besides this, the bot requires the user to authenticate himself via an OTP before filing a claim on any of his policies.

1.5 Assumptions

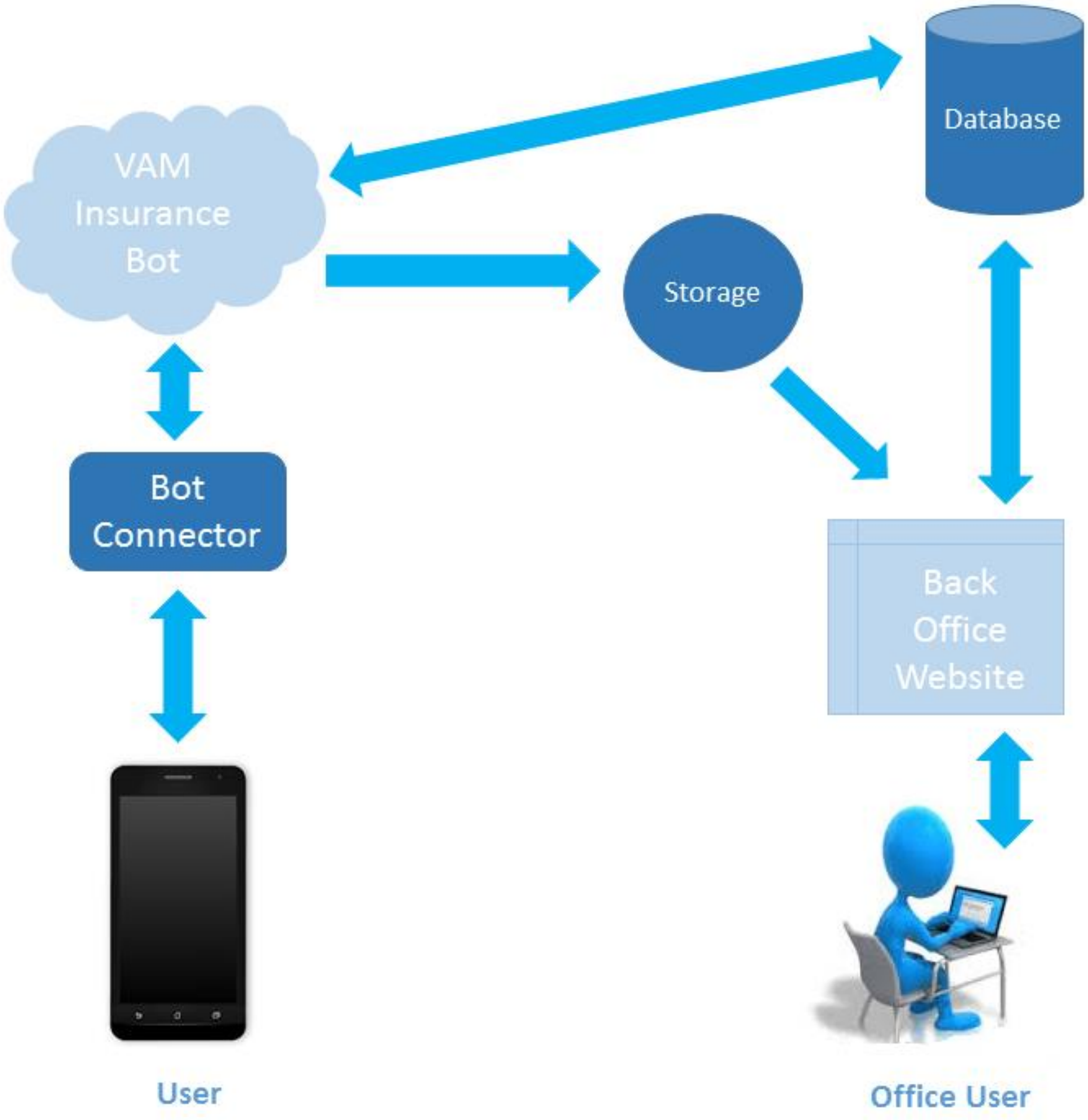
It is assumed that the user is proficient with the use of a smart device and has a basic understanding of the English Language to be able to comprehend and reply appropriately to the questions asked and the statements made by the bot. It is also assumed that the user is equipped with necessary documents like the license of a driver in case of an accident or the FIR in case of a theft, etc. at the time of notifying his loss.

1.6 Risks

A bad network connection may be a possible risk as it may interfere with the continuous dialog flow between the user and the bot and could make the process a bit difficult. Also, a signal loss at any time during the claim procedure may result in the claim not being filed due to expiry of session.

2. Technology Architecture

2.1 Application Architecture



- **USER**: In order to connect with the bot, the user must install the Telegram app on his smart device and add the bot to his conversations.
- **BOT CONNECTOR**: The Microsoft Bot Connector is a middleware that helps the bot connect to different channels. Hence, this bot coupled with Microsoft Bot Framework compatible API, will allow the Connector to forward messages from the Bot to a user, and will send user messages back to the Bot.
- **VAM INSURANCE BOT:** The bot is hosted as a Web App on Microsoft Azure. Cloud hosting ensures enough resources for scalability, performance and uptime.
- **DATABASE:** The database may be located anywhere, either on the cloud or locally. It stores details about the policy holders belonging to the company,

information about their policies and also a history of renewals and claims filed by customers.

- **STORAGE:** The storage is primarily a blob storage and is currently hosted on the cloud. It segregates user data based on PAN number and policy number. It holds the logs of all conversations with the bot and also stored images sent to the bot.
- **BACK OFFICE WEBSITE:** The back office website grants officers in the claims department the access to view, approve or reject claims and renewals made on the various policies the customers of the company own.
- **OFFICE USER:** This category of users would pertain to all persons in the claims department in-charge of approving or rejecting claims made on policies.

2.2 Data Access Layer

The data is accessible to the all back office users (having appropriate permissions), the bot and the administrator. Users are permitted to view data related to themselves and their policies. The bot can access personal and policy details of all users conversing with it. The administrator has absolute access to the database and can also access the logs of all conversations that take place with the bot.

2.3 Error Handling

Should an error occur, an explanation will be provided by the bot describing the problem. Instead of ending the conversation, the bot will prompt the user to reply as expected. An error, in this application, can be defined as data entered by a user that does not correspond to his details, data entered in a format apart from that specified or any sentence uttered by the user that falls outside the range of the bot's comprehension capabilities.

2.4 Current Operating Model

The current model is equipped with the basic features required to file a claim. It is partially menu driven and has the ability to understand few natural language sentences.

2.4.1 AT THE ONSET

The conversation always starts from the user's end. He is first required to identify himself as an existing policy holder or a new customer and the bot chooses its reply accordingly. Existing customers are asked for the policy number they would want to inquire about or claim, while new users are simply given the contact details of the company which may include the URL of its website and/or a few telephone numbers.

2.4.2 USER VERIFICATION

The verification is a two-step procedure. The bot asks the user for his PAN number and checks if it matches with any of the PAN numbers of the existing customers of the company. If it does, the user is asked for his date of birth as in the company records, else, the user is intimated that the entered PAN number is invalid. The date of birth given by the user is matched against the entered PAN number. Only if both the entries match with the records of the database, it verifies the user by sending an OTP to his registered Email-address. This is an essential security step as only the genuine clients of the company are given access to their policies to query or claim them.

2.4.3 TASK MENU

Successfully authenticated users are displayed a list of the policies they own. The customer enters the required policy number and the bot checks whether the entered policy number is correct and prompts the user to enter the right number in case of a discrepancy in the characters. The user is then given a list of tasks that he can ask the bot to perform. The list includes filing a claim on the policy, filing a renewal for the policy, viewing the details of his policy, navigate to another policy owned by the user and also an option to exit the conversation. The task chosen by the user directs the conversation further. At this point, the bot also intimates users, whose policy expiry date is less than three months from the current date, to file a renewal.

2.4.4 VIEWING POLICY DETAILS

A request to view the policy requires the bot to query the database and retrieve details of the policy and the policy holder. The details of the policy shall include its identification number, its start date and its expiry date, a list of claims that have been filed for that policy along with the claim number and dates and a similar list for the renewal history. The details of the policy holder may include personal details of the user like his name, age, address, contact details, email-address etc. These details are subject to change depending on the database of the insurance company.

2.4.5 REQUESTING FOR A POLICY RENEWAL

A request to renew the policy allows the bot to mention the date of expiry to the user and the bot sets the renew field in the database against the policy number. Further, a record of the renewal request along with the date and policy identification are stored in the database.

2.4.6 CLAIMING THE INSURANCE BENEFITS

2.4.6.1 Ensuring valid policy and non-duplicate claim:

A request to file a claim on the policy is a more complex process. The bot first checks the validity of the policy by comparing the current date and its date of expiry and rejects the request if the policy is invalid. Further, it checks if the initiated claim is a duplicate claim and indicates to the user if the claim has already been filed along with the date of filing and is still in an open state. The bot proceeds with the actual enquiry only if the policy is valid and the claim is not duplicate.

2.4.6.2 Identifying claim category:

The bot first identifies the type of the claim by asking the user if it is a case of theft or damage. The dialog flow further is directed by the category chosen by the user.

2.4.6.3 Damage:

In case of a damage, the bot questions the user about the date of the accident, its location, involvement of a third party and also gives the user the liberty to describe the accident. It further captures details about the vehicle involved in the accident along with the details of the person driving the car at the time of the accident. In case a third party is involved in the accident, the user is required to specify certain necessary details of the third party. The bot also asks the user to post pictures of the damaged vehicle, the license of the driver, RC of the vehicle and an image of the FIR in case of third party involvement. The bot prompts the user to send pictures when the message from the user does not contain an image.

2.4.6.4 Theft:

Similarly, in case of a theft, the bot acquires from the user a description of the vehicle and the date, location and nature of the theft through a series of questions. Relevant pictures are required to be sent to the bot in this case as well as and when the bot asks for them.

2.4.6.5 Editing details, viewing claim and submitting:

Once all the details the bot needs are captured, it displays them to the user and inquires if any field requires changes. Fields that need to be changed are mentioned by the user and the bot registers them again. Finally, when assured by the user that all details have been correctly entered, the bot saves the details to the database, generates a claim number for the claim and displays it to the user so that he can take note of it for future references.

2.4.7 OTHER FEATURES

The bot, after every procedure, gives the user an option to dictate it to perform any other task it is capable of. The user can also navigate to other policies that he owns in a single conversation with the bot.

A salient feature of this bot is its ability to remember the date and time of the last message it sent to the user in a particular conversation. This is useful for validating the session of the user as he is barred from continuing a particular conversation if he replies to the bot’s message after the delay of a certain period of time. This ensures that a user authenticates himself every time he enters a new conversation with the bot.

Another important feature of the bot is the chat log. Records of all conversations with every user are stored in the cloud. They are segregated based on the policy numbers and date and time of querying the respective policy. Hence, a single conversation may require more than one file, depending on the number of policies the user navigates to. This makes tracking renewals and claims filed on a policy convenient. Images posted by the user when claiming a policy are also named in a similar fashion and incorporate the topic of the image in their name. Thus, accessing the details of a claim made on a policy is a breeze.

2.5 Target Operating Model

The target operating will resemble the current model in its basic functionality but additionally may incorporate the following features:

- To enable it to garner more requests, it may be connected to more channels like Skype, SMS etc. if required.

- The current verification is a two-step process that authenticates via an OTP sent to the registered E-mail address. It would be changed to authentication through an OTP sent to the registered mobile number for distribution purposes.
- The database may not necessarily be hosted on the cloud. Its location will depend on the location of the company's database.
- Presently, the bot is capable of understanding only certain natural language sentences that contain keywords that it understands. Efforts shall be made to enable the bot to comprehend more sentences and hence, improve its intelligence through application of advanced techniques like AI or ML.
- The current dialog flow of the bot can be changed to include more dialogs depending on the company's needs.

2.6 Availability

The application is available for use to any individual with an internet connection on his smart device. He must have access to his E-mail account while using the application and can converse with the bot through Telegram.

2.7 Security

The use of SSL establishes an encrypted link between the client and the server. It allows secure transmission of sensitive information like the user's PAN number which is used by the bot for identification. The two-step verification gives users authority only to tamper with the policies they own. Moreover, claims filed on a given policy are recorded and hence duplicate claims are rejected by the bot. The chat logs per policy also allow a detailed study of the conversation in case of any suspicion.

2.8 Scalability

Since the application is hosted on Microsoft Azure, its scalability is directly related to Azure web app scalability. Scaling features in Azure depend on the app service opted for. The suitable app service can be selected and Azure offers its users the flexibility to change the app service at any time, if the need arises. Thus, configuring scaling is easier in Azure as compared to traditional hosting. Azure SQL and Azure Storage are Geo Redundant with approximately 6 copies made over 2 servers in different locations providing almost zero downtime.