PrIA:Chat-bot assisting Financial Advisors in Life Insurance Industry *

Barbara Ferreira
Digital Media Engg
Technical University of Denmark
Copenhagen, Denmark
s161526@student.dtu.dk

Shishupal Rathore
Digital Media Engg
Technical University of Denmark
Copenhagen, Denmark
s160962@student.dtu.dk

Sudhir Chaurasia
Computer Science Engg
Technical University of Denmark
Copenhagen, Denmark
s137239@student.dtu.dk

ABSTRACT

This study researches feasibility of intelligent chat-bots as trustworthy tools for critical business use cases like customer acquisition process in life insurance industry. The research was conducted by developing an intelligent chat bot integrated with Facebook messenger and an independent web page. This bot could converse with the users, save shared details, assist users on searching for relevant information and building awareness, knowing premium rates and applying for insurance. The research was to observe and understand user reactions, to the design of conversation, the response time, accuracy of results and information security. As part of the research, it was observed that acceptance to the tool on Facebook platform was fairly easy due to familiarity of platform. However concerns around information security and privacy persisted as personal details of users were being shared through the platform and the insurance company through Facebook would have access to private information. Experiments and user interviews demonstrated that familiarity to platform improves learning ability and makes it convenient for users to explore new tools. It also demonstrated that users expected organizations to take responsibility for safely guarding user's personal information. Users were open to accept marginal errors in accuracy of results but were unforgiving to security lapses and any compromise with their confidential information. The research was able to identify an optimum response time.

Keywords

Chat-bot, Conversational Agent, Information Extraction, Decision support systems, User experience, User-centric evaluation, Decision-making, Human-computer interaction, User testing, Guided conversations, Preference elicitation, Privacy, Life Insurance

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

1. INTRODUCTION

Traditionally, financial advisors have assisted customers in building knowledge about life insurance, evaluating different risks and buying life insurance. The customer acquisition process in Life Insurance industry is represented in Figure13. Most of the conversations happen in person with advisors, takes time and is perceived as inefficient and expensive. Customers also perceive insurance application process as slow, inefficient and lacking transparency.

As per Israel Gonzalez-Carrasco[7] an Intelligent chat-bot is a computer program that is designed to simulate an intelligent conversation with a human user via auditory or textual methods. These chat-bots are designed to engage in small guided conversations and the personas are designed and created by programmers.

As per Oracle [10], 6 out of 10 most used apps globally are messaging apps with 4.1 Billion users. It also predicts that 65% of the consumers prefer messaging app when contacting a business and over 50% would prefer a purchase through a messaging app. With 90% of the businesses using Social media to respond to customer requests, Oracle predicts that Chatbots could save $174 Billion across Insurance, Financial Services, Sales, and Customer Service. There are user concerns around trust, ethics and integrity but if they are well addressed, then AI personal assistants could result in a new and efficient channel of conversing directly with customers, helping them on building understanding, reviewing products and buying life insurance products. An approach combining AI personal assistant and Financial advisors could pave the way for huge cost savings for insurance companies and deliver a highly efficient and effective customer experience.

This paper describes an approach to design and develop intelligent chat-bots that can simulate conversations and helps customers in applying for insurance, searching for insurance related information and calculating premium. The purpose is to design a digital conversation solution that could efficiently replace part of personal conversation that a Financial advisor has with customers. A critical aspect will be to ensure that an appropriate balance is maintained between convenience and information security. Providing users with the control to talk to the Intelligent bot at his/her convenience, getting details and information and addressing concerns that a customer has, could create a strong relationship of trust between the customer and insurance company. It could better comprehend customers needs and help in providing more suitable solutions and services to customers accordingly.

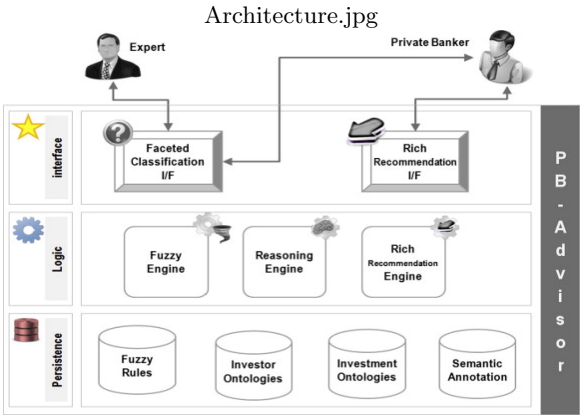

Figure 1: PB-Advisor Architecture [7]

2. RELATED WORK

Israel Gonzalez-Carrasco et al [7] designed the PB-Advisor, a multi-investment recommendation system, based on semantic technologies and fuzzy logic. By combining the two technologies with faceted searching, the system provides private bankers with tailor-made recommendations for their clients, based on their characteristics. PB-Advisor suggests private banker with the most suitable portfolio for each investor, considering account investors preferences and risk tolerance. The key features of the recommendation process are:

- Investor categorization based on facets
- Determining the investor profiles based on social and psychological characteristics, by using Semantic and Fuzzy Logic approaches
- Investment (portfolio) classification based on Semantic and Fuzzy Logic approaches
- Matching the portfolio characteristics with the investor categories
- Obtaining a rich recommendation based on XBRL (eXtensible Business Reporting Language)

Based on fuzzy logic and ontologies, PB-Advisor categorizes the investor and with the faceted classification process it automatically determines the investor profile with respect to risk tolerance. Figure 1 represents the architecture of PB-Advisor.

Another related work example is of Alexander Felfernig's [6] designed, knowledge-based recommender application FSAdvisor (Financial Services Advisor). It provides assistance to sales representatives and determines personalized financial service portfolios for the customers by simulating the behaviour of sales experts. It is licensed to a number of prominent financial service providers in Austria. It supports the conversation between a sales representative and a customer by assuring consistency, relevance and accuracy of proposed solutions, identifying additional selling opportunities and by providing intelligent explanations for solutions. FSAdvisor, as showed in Figure 2, integrates model-based diagnosis, constraint satisfaction and customization, and supports customer-oriented sales conversations.

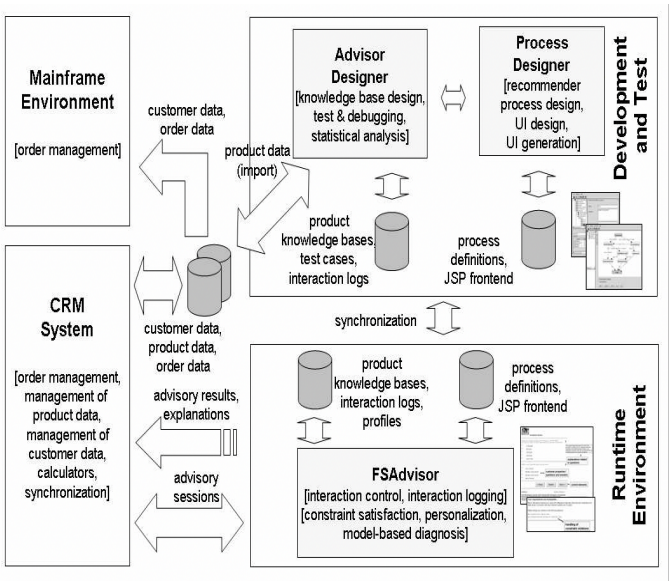

Figure 2: FSAdvisor Architecture [6]

As part of the solution it uses Selected Recommender Technology that exploits deep knowledge about the product domain in order to determine solutions matching customer's needs. FSAdvisor is also based on constraint satisfaction problem solving. The constraint solver identifies solution for a recommended task. If no solution is found, constraints with a priority > 0 (0 is the highest priority) are shared in order starting from lowest priority. Later, if only non-relaxable constraints (priority $= 0$) remain, a repair mechanism is activated.

The solution is currently installed for around 150 sales representatives of the Hypo-Alpe-Adria bank since July 2003 and for 1400 sales representatives of the Wustenrot building and loan association since June 2004. The motivation of developing a knowledge-based recommender technologies was to improve the efficiency and effectiveness of sales processes in terms of solution quality, accuracy and intelligent documentation of advisory sessions.

A very different application of an AI based chat conversation was demonstrated by Stavros Vassos et al.[1] with Artbots - deployed for providing engaging experiences in museums. The Art-bot interacts with visitors through Facebook messenger chat and conveys information about the museum artifacts in the form of short stories. The Art-bot impersonates Mario Praz and is able to converse with guests and visitors in the Mario Praz museum. Through direct conversation with an automated chat-bot persona on Facebook the Art-bot reveals the intricate and "hidden" stories behind the life of Mario Praz. The Art-bot prompts visitors to ask questions, challenges them to carefully observe artworks and think about them.

Mario Praz's personified chat-bot [1] is a first step towards a modern exhibition method that:

- is not invasive to the museum setting
- allows an engaging communication through the familiar chat interface
- conveys the hidden information about the exhibits in

an organic way inside casual conversations

- provides a user friendly way to receive feedback in a gamified manner

Other related works are from Chu et al [2] who proposed an intelligent trading advisor, Mohammed et al [13] proposed Financial Software Advisor: COFINA and Lee [8] proposed iJade stock advisor which are very similar examples of chat-bot based advisory solutions in financial domain. However none of the text-based conversation solutions have been designed and tested in Life Insurance domain, that this project could refer to.

3. PROPOSED SOLUTION

As the group researched related work and industry reports, the group built an understanding of the chat-bot solutions and the way it works. To better understand, user pain points, the group talked to users in the age group of 23-55 years and discussed with them the challenges they face while applying for or buying a Life insurance. The group also discussed with a couple of financial advisors to understand the process of insurance application and final approval. Current process of customer acquisition in life insurance is represented in Figure 13.

Key concerns as shared by the users were around the lengthy procedure of application, very little transparency especially around the charged premium rate, and dependency on information shared by financial advisor. Many users would build understanding of insurance and products during the interaction, and information provided by advisors would usually be considered trustworthy. Few of the users would later also search online to verify the information shared by advisors.

When asked for possible solutions that could make this process interesting, some of the suggestions were around providing an interactive technical solution that is flexible, helps user to gather trusted information before taking the decision and is transparent i.e. it could explain logic behind the calculation of the premium rate. Based on the user discussions three key requirements were identified for the group's solution - PrIA:

- Interactive technical solution that could gather the required details with an interesting conversation and provides user the flexibility of sharing details at any time
- Solution that could help user to gather information online within the conversation and help in taking final decision
- Sharing approximate premium rate that will be charged and the logic of calculation in an understandable way

For the first requirement, group researched on the documents that users usually fill - either online or during conversations - to provide all the details required for the application. The details in the application form [3], were the basis for design of the personal details to be requested during conversation.

For the second requirement, google search results were determined as the most trusted source of information to user's queries and to build user's knowledge

For the third requirement, group researched on additional documents and identified a paper [12] that explained the calculation of term insurance premium rates for a defined population. This paper formed the basis for the algorithm that was coded to share premium rate with users and to explain the calculation logic.

Before, initiating the design and development process, the group again met the earlier interviewed users to validate if the requirements identified would address the concerns as shared by users earlier. There was a positive go ahead from the users on the requirements.

3.1 Design

To ensure acceptance of solution, it was very important that the solution is designed such that is easy to use and learn. A lot of chat-bot based industry experiments have utilised independent web based solutions as well as social media e.g. Facebook. Each solution has its pros and cons in terms of perceived familiarity of platform, ease of use, convenience, information security and trust. Determining the right platform as solution was the first Hypothesis that the group decided to test with users. We asked users to chat with Dominos Pizza customer service representative [5] on Facebook. Everyone tried it and the experience was positive. However there were 2 common concerns - usage of Facebook platform for financial business like Life Insurance and sharing confidential details on Facebook i.e. information security.

The findings from this testing were very important for our group and hence the decision was to build a bot that could be integrated with Facebook platform as well an independent website and test both the interfaces with the users. Few technical considerations were to ensure that solution is easily accessible on tab and smartphone and the experience across platforms is consistent.

3.2 Technical Architecture

The conceptualized solution required considerations for integration with:

1. Facebook and any website where the bot may need to be integrated or hosted.
2. Google Mail or any mail framework for insurance application confirmation and query resolution.
3. Google sheet or any database for storing customer information.
4. Google Search or any other search application to search, retrieve and visualise information for user.
5. Premium calculation code

There are many technical tools available for building chat bots from scratch.A comparison of different platforms is demonstrated in Figure 3. Olga Davydova [4] compared the best and most popular 25 chatbot frameworks on multuple parameters.

Based on the group's evaluations criteria around flexibility, open framework and integration capabilities, a combination of API.AI and FlowXO were selected to design and develop the prototype. Figure 4 is a high level depiction of the technical architecture of the prototype solution.

Each component in the PrIA architecture was integrated to deliver a very specific functionality:

Bot Name	Features	Programming languages / Apps / Integration	Channels
IBM Watson Conversation Service	Built on a neural network (one billion Wikipedia words). Has three main components: Intents, Entities, Dialog	Node SDK; Java SDK ; Python SDK; iOS SDK; Unity SDK	speech;image;text
AgentBot	Understands natural language.	Memory to maintain coherence during long conversations.	Gathers customer information to deliver customized solutions.	Continuous evolution.	Clarifies intent.	Use our REST API to integrate with your CRM and other platforms.	voice or messenger channel
Msg.ai	Intent models and tone classifications	Deep learning	Interactive Smart Cards	A/B Testing	Integrates with popular customer support offerings.	messaging
wit.ai	Allows to use: Entities	Intents	Context	Actions	Natural Language Process (NLP)	Node.js client	Python client	Ruby client	On other platforms: HTTP API	voice	text
Api.ai	API.AI matches the query to the most suitable intent based on information contained in the intent (examples, entities used for annotations, contexts, parameters, events) and the agent's machine learning model. API.AI transforms the query text into actionable data and returns output data as a JSON response object. Leverage predefined knowledge packages collected over several years.	SDKs: Android	iOS, Cordova	HTML	JavaScript	Node.js	.NET	Unity	Xamarin	C++	Python	Ruby	PHP (community supported)	Epson Moverio	Botkit	Java	Agent Demo Page	Actions on Google	Facebook	Slack	Twilio IP	Messaging	Twilio	Skype	Tropo	Telegram	Kik	LINE	Spark	Alexa	Cortana	Twitter
Microsoft Bot Framework	Understands the user's intent. To give your bot more human-like senses, you can incorporate LUIS for natural language understanding, Cortana for voice, and the Bing APIs for search.	Bot Builder SDK (.NET SDK and Node.js SDK.)	Bot Connector	Developer Portal	Bot Directory	From your website or app to text/SMS, Skype, Slack, Facebook Messenger, Office 365 mail, Teams and other popular services.
Microsoft Language Understanding Intelligent Service (LUIS)	Uses intents and entities. All LUIS applications are centered around a domain-specific topic or content related. Active learning. You can use pre-existing,world-class, pre-built models from Bing and Cortana. Deploy models to an HTTP endpoint with one click. LUIS returns easy-to-use JSON.	C# SDK	Python SDK	Node JS SDK	Android SDK	Activate your language understanding models from your application on any device. You can incorporate LUIS for natural language understanding, Cortana for voice, and the Bing APIs for search.
ChatterBot	The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then chooses a response from the selection of known responses to that statement.	ChatterBot is a Python library Has direct support for integration with Django	Console	API	Speech recognition
Octane.ai	Real-time analytics	-	Facebook Messenger
FlowXO	https://flowxo.com/features	https://flowxo.com/services	For Messenger, Slack, SMS, Telegram & Web
Reply.ai	Visual Bot builder. Easily leverage Natural Language Processing (NLP) engines wit.ai and api.ai for your advanced use cases. Analytics	Integrate with your backend	We support Facebook Messenger, Kik, Telegram, LINE, SMS or your own chat screen. We use native custom UI elements in each channel. Web widget available too.

Figure 3: Bot Frameworks Comparison [4]

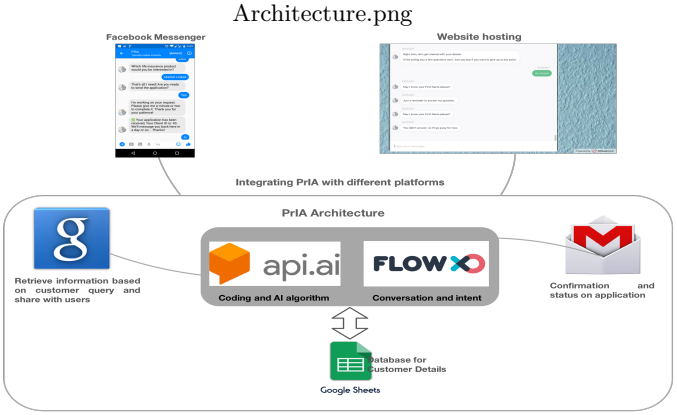

Figure 4: PrIA Architecture

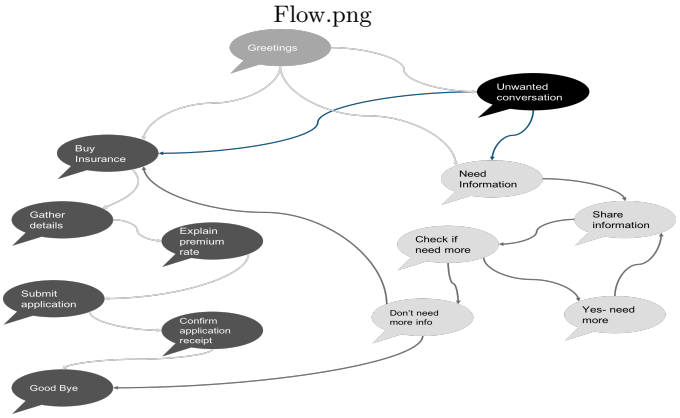

Figure 5: Conversation Flow Design

- FlowXO has very useful features to design conversations, categorise intents and drive rule based conversations.
- Api.ai (Dialogflow) allows development and integration of different executable codes e.g python or Java code which FlowXO doesn't allow. Combination of FlowXO and Api.ai became the foundational development platform for PriA.
- GoogleSheets, used as a database to store all information shared by users during the conversation, was integrated with this platform.
- Gmail integration helped in communicating with users outside the chatbot. This helped in sharing confirmation on the details with customers along with application reference number from a verifiable and genuine email address.
- To ensure that users could ask a query and a verifiable response could be shared, Google custom search was integrated. With this functionality, users could ask a question about life insurance, products, eligibility criteria etc. during the conversation and results were shared with users which could later be verified by users if they wanted to. This feature was incorporated to ensure that trust is enhanced in the overall conversation.
- The algorithm for premium calculation was coded in python and was then integrated as part of the overall chat-bot solution.

The bot developed was then integrated with Facebook messenger API and an external webpage. Important thing to ensure was that the conversation experience is consistent across the platforms. The different features of the bot conversation are represented in Figure 6.

3.3 Designing the Conversation

The group referred this document [3] to extract the details that are required for completing the insurance application. Overall conversation was designed considering three high level intents and 1 context:

- **Apply for Insurance:** Users are ready to share the details and apply for insurance
- **Know about Insurance:** Users have queries and they want response to their queries before buying the insurance
- **Premium Calc:** Users could provide details and know the approximate premium amount for a Term Life insurance product
- **Small Talk:**This flow was about Greetings, Farewell, Confirmation, guiding the conversation if it goes in different direction etc.

The engagement within chatbots [9] could be designed as Declarative (task-oriented) or Conversational (Data-driven and predictive). As part of the research, the prototype engagement was designed as a combination of declarative and conversations. Tentative flow of a guided conversation within different contexts is represented in Figure 5.

3.4 Premium amount calculation

During initial requirement conversations with customers, a key concern raised was transparency in the premium amount being charged. Customer perception is that the premium amount charged is very high and there is no transparency in the way the premium is calculated. This factor was the key concern in trusting the insurance company.

To address this concern, group decided to research on the premium calculation and this paper [12] from NYU - Stern

was referred to develop program for calculating the premium calculation and sharing the key values with customers.

The actuarial mortality table describes the life span of Americans and is an input parameter in the algorithm.

Expected Premium Stream $= P \sum_{x=t_0}^{t_n} e^{-r(x-t_0)}(l_x/l_0)$
where **P** = Current premium value
t_0 = Customer current age
t_n = Age till which cover is requested
r = Interest rate
l_x = Number of lives at a particular age from Mortality table
l_0 = Number of lives from Mortality table at the current age of customer

Expected Payout Stream$= M \sum_{x=t_0}^{t_n} e^{-r(x-t_0+1)}(d_x/l_0)$
where **M** = Expected Death Benefit
$d_x = l_x - l_{x+1}$ is the death probability from mortality table

If *Expected premium stream = Expected payout stream* then
P/M $= \sum_{x=t_0}^{t_n} e^{-r}[(l_x - l_{x+1})/l_0]$
Approximate annual premium $= P/M * policyfacevalue$

4. FINDINGS

As part of user testing for the project, 6 users volunteered to help the group in defining the requirements, chat with the bot on a regular basis, engage in timely user interviews and provide their feedback to develop and improve PrIA. The age of the users varied from 23-55 and were from different nationalities - Bulgaria, Brazil, Sweden, Nepal, India and Denmark. All the users were professionals, had bought life insurance in the past and were aware of the typical application process. Moreover, the group had access to a very young fellow student user group of 15-20 users who helped in testing the app and providing lot of valuable feedback for improvement.

The group kick started the project, to gather requirements, by conducting user interviews with the 6 users. Detailed questionnaire utilised during the whole user testing process can be referred in Figure 14.

Objective of the first experiment with users was to validate the hypothesis on whether Facebook messenger as a medium could be a potential solution or not? During the interview, a quick experiment of 'Dominos' pizza messenger bot was done. Without any help, users were able to quickly converse with the bot and order Pizzas. This experiment provided ideas to the group to design the conversations for Facebook bot.

The second important hypothesis was whether users would be open to Social Media messenger platform or the preference would be a secured and closed chat experience. Users were wary to the security concerns, in general, with Facebook or other Social media platforms but reported that the overall chat experience on Facebook is very user friendly and if the organization takes responsibility of ensuring security of the user information then the users would be open to Facebook messenger. Both the hypothesis were validated with 15 users. From the results as shared in Table1 indicates that 9 users prefered FB messenger platform as well as Web based bot. Interestingly, one user was not familiar with Facebook messenger before test but he also liked FB messenger bot while testing for Dominos bot. Also, 4 users were comfortable with any solution.

User	Familiarity-to-FB-Messenger	Prefer-FB-Messenger-bot	Prefer-Web-based-Bot
User1	Y	Y	N
User2	Y	Y	Y
User3	Y	N	Y
User4	Y	N	Y
User5	Y	N	Y
User6	Y	N	Y
User7	Y	Y	Y
User	Y	Y	N
User9	Y	Y	N
User10	N	Y	N
User11	Y	Y	N
User12	Y	N	Y
User13	Y	Y	Y
User14	Y	N	N
User15	Y	Y	Y

Table 1: User Interview Results

From interview results, it was clear that most of the users were comfortable with any platform. Hence, the group initiated design and development of the solution considering integration with Facebook and an independent website.

4.1 Testing the Conversation Flow

As part of the first iteration, the group had researched and collected words and phrases and this were added as part of the conversation module. During testing, goal was to observe the words and phrases that users use while initiating and conducting the conversation and collect bag of words and phrases. Interestingly a lot of similarities were observed when users initiate conversation online and common initiation happened with 'Hi', 'Hello', wishes etc.

4.2 Determining the Response Time

User experience research from Sabre Labs [11] mentions that till 8 seconds a user waits until he or she starts trying to reboot the 'unresponsive' chatbot. Moreover, in absence of hard feedback on the background processing, users are automatically turned off after 20 seconds. We tested our response time with 15 users. 8 users on FB messenger chatbot and 7 on independent website.

In the first experiment, general response time was set to half second and for heavy processing or calculation, it was set to 10 seconds. Users reported that the general response time was too fast for a human and for heavy background processing it seemed too slow.

In second experiment, general response time was set to 2 seconds and for heavy processing or calculation, it was set to 7 seconds. Out of 15 users, 9 reported that general response time was slow. For heavy background processing conversation 12 users reported that response time was appropriate.

In third experiment, general response time was set to 1 second and for heavy processing or calculation, it was set to 8 seconds. Out of 15 users, 13 users reported that general response time was appropriate. For heavy background processing, 14 users reported that response time was appropriate.

In the final experiment, general response time was set to

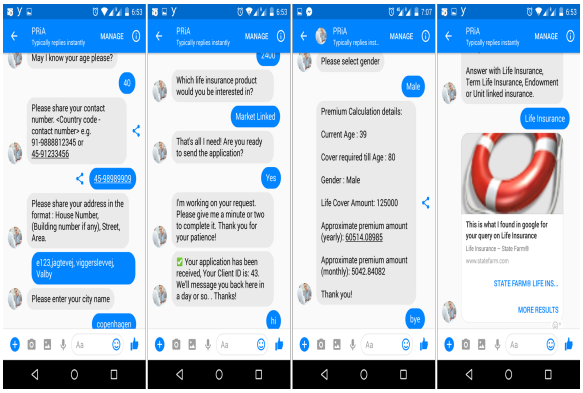

Figure 6: PrIA-Chatbot (a)Apply for Insurance (b) Confirmation on Application (c) Premium Calculation (d) Search for Information

1.5 seconds and for heavy processing or calculation, it was set to 7.5 seconds. Again, 12 users reported that general response time was appropriate. For heavy background processing, 13 users reported that response time was appropriate.

With these experiments, the group was able to define the response time thresholds for general conversation responses and for heavy processing conversations. Throughout the testing, the following thresholds were maintained:

- Response time for general conversation at 1-1.5 sec
- Response time for heavy processing conversations at 7-8 sec

4.3 Testing the Search Information flow

In the second iteration, search feature was incorporated in the bot along with the insurance application feature. With the search feature, user could seek information on insurance in general and on the products before applying for insurance. As part of this feature, google search was invoked on the query shared by user, and then the results were displayed along with a link to the website and a short summary of the search result. This feature was a very critical for the project as this would help in building trust and confidence with users.

There were multiple hypothesis to test on this feature:

- Would users expect 1 result on the question asked or they would expect multiple results?
- Would users expect pictures along with text result or only text result would suffice?
- Would users be interested in a short summary with a flexibility to click on the link and explore further details or the users would expect all the details in the chat?

As part of the first experiment, 10 users were provided with the top most google search result on the query and this was displayed along with the picture and link to the website. Also users were provided with a flexibility to ask for more results. User response was positive for the results displayed. They liked the picture along with a short summary and 8 users clicked on the link to visit website and gather more details. Overall, the feedback from users was to provide more results option so that users could look for more results in case required. Since, the app couldn't continue to provide multiple results, hence users suggested that maximum 5 results could be shared with users and after that they may have to again initiate the search.

5. DISCUSSION AND CONCLUSION

The goal of the project has been to test the feasibility and effectiveness of chat based applications in the insurance industry. It is a democratic way of letting users, at their convenience, gather and validate information and eliminate the dependency on financial advisors to provide domain related information. The essence of letting user validate the shared information builds a feeling of transparency, trust and confidence in the overall process. Building the features on a known platform like Facebook messenger, considerably reduces the time taken to learn and become familiar to the app and in turn reduces user acceptance curve. The three features of applying for insurance, search for information and premium calculation tries to address the general perceptions and concerns of insurance users around boring process, no transparency and dependency on financial advisors.

However, the developed application still has very basic features and for it to become a full fledged application will require considerable amount of additional efforts. As part of the prototype, the group incorporated basic Natural Language understanding to ensure a guided conversation is followed. But with a diverse set of users, the Natural Language processing will need to be robust to intelligently determine user intent for a large variety of words, phrases and sentences.

For the group to provide product related information, it needed to have access to Insurance companies APIs and the access is only available to registered advisors. Hence the search function currently developed, gathers the available information from google for very specific parameters related to insurance. In case of insurance companies, they could leverage the current feature with their product APIs and provide very specific insurance and product related information to the customers. In the current prototype, the results that are provided to the users have been restricted to 5, but based on business scenarios these could be expanded to any number.

While building the premium calculator, group has currently referenced the U.S. Social Security Life Table, based on year 2014. This table contains the mortality rates and life information for residents of the 50 States and the District of Columbia in US, civilian residents of Puerto Rico, the Virgin Islands, Guam, American Samoa, and the Northern Mariana Island. This was the only life table publicly available for the group to refer while developing the premium calculator and hence the calculation from PrIA may be relevant to the residents of United States only. In case the application has to be used for residents outside of United States then appropriate life table will need to be referenced. Moreover, the premium calculation currently demonstrated covers only Life term product. For other products, respective actuarial algorithms may need to be referenced.

Overall, the prototype was a demonstration of feasibility of chatbots for customer acquisition process. However, to make this application business ready, will require concen-

trated improvement efforts and future work.

6. REFERENCES

[1] S. V. B, E. Malliaraki, F. Falco, J. D. Maggio, M. Massimetti, M. G. Nocentini, and A. Testa. Interactive Storytelling. 10045:433–437, 2016.

[2] S. C. W. Chu, H. S. Ng, and K. P. Lam. Intelligent trading advisor. *Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology:*, 1:53–58, 2000.

[3] L. I. Corporation. No Title. 300(300), 1956.

[4] O. Davydova. 25 Chatbot Platforms: A Comparative Table.

[5] Dominos. Dominos Messenger Bot.

[6] A. Felfernig and A. Kiener. Knowledge-based interactive selling of financial services with FSAdvisor. *Proceedings of the National . . .*, 100:1475–1482, 2005.

[7] I. Gonzalez-Carrasco, R. Colomo-Palacios, J. L. Lopez-Cuadrado, Á. García-Crespo, and B. Ruiz-Mezcua. PB-ADVISOR: A private banking multi-investment portfolio advisor. *Information Sciences*, 206:63–82, 2012.

[8] R. S. Lee. iJADE stock advisor: An intelligent agent based stock prediction system using hybrid RBF recurrent network. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 34(3):421–427, 2004.

[9] Oracle. Chatbot Primer.

[10] Oracle. Oracle Chatbots Infographics.

[11] E. Seo. Best UX Practices for Building Chatbots.

[12] N. Stern. TERM INSURANCE CALCULATION ILLUSTRATED. 7224000(254 20):7224135, 2010.

[13] K. A. Touahria Mohamed and F. Louis. Cognitive Modeling of a Financial Advisor- COFINA.pdf, 2007.

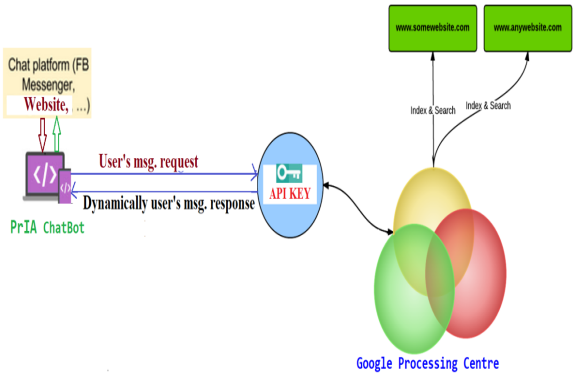

Figure 7: API Integration High Level Architecture

APPENDIX

A. GROUP CONTRIBUTION

As part of the report each of the Group members have contributed equally

B. LINK TO PRIA

Facebook page link is: https://www.facebook.com/insurintelligentl

Webpage link is: https://depreciative-suppre.000webhostapp.com/Priya.html

C. API INTEGRATION

As part of the PrIA application development, multiple interfaces e.g. Google Sheets, GMail and Google Search were integrated with the application. A high level integration flow of PrIA with Google applications in represented in Figure 7

C.1 Custom Search Credentials

First, create the project by using google cloud console link to create custom search credentials and after creating it enable API before using it, to make it search based on user's request.

Here is the link: https://console.developers.google.com/apis/

Process has been represented in Figure 8

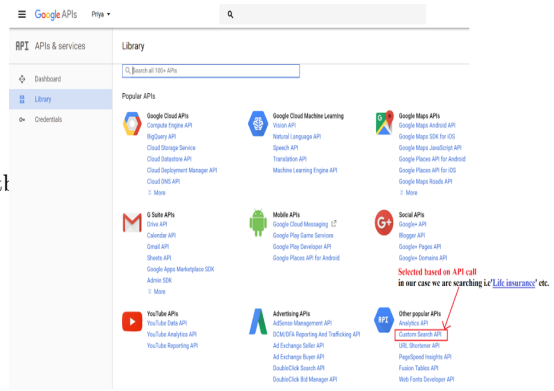

Figure 8: Create Custom Search Credentials

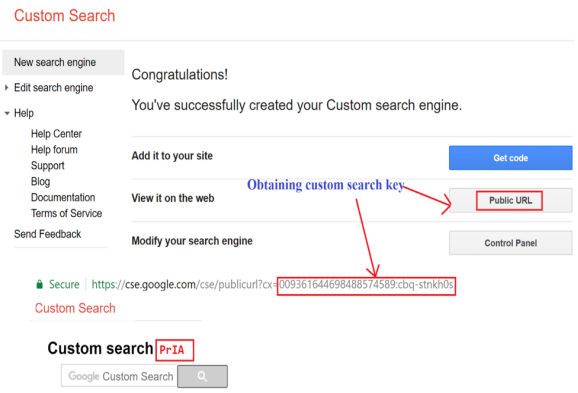

Figure 9: Get Custom Search Key

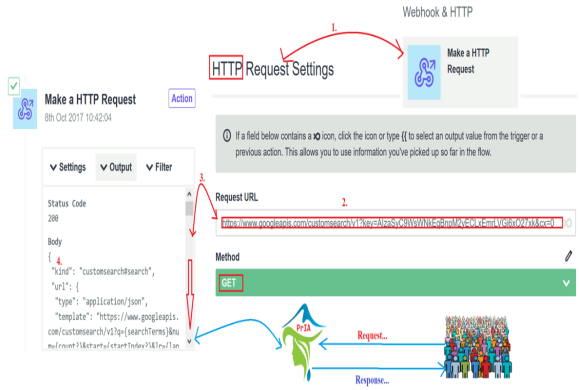

Figure 10: Configuration to call API

C.2 Get Custom Search Key

Using Google CSE URL, create custom search engine, to get CSE key from Public URL. Link: https://cse.google.com/cse/ Process has been represented in Figure 9

C.3 Configuration to call API

Configure API call in FlowXO using URL to call Google Search Engine dynamically based on user's queries. Process has been represented in Figure 10

D. USER TESTING

As part of the user testing, 6 users volunteered to help in testing the functionality. Few pictures captured during user testing can be referred in Figure 11

E. USER STORY MAP AND MVP

Group followed lean methodology while designing and developing the PrIA application. The user story map and MVP were defined as part of the process. It can be referred in Figure 12

F. CURRENT PROCESS FLOW

The current process flow of Customer Acquisition has been represented in Figure 13

G. USER TESTING QUESTIONNAIRE

The questionnaire referred during user testing could be referred in Figure 13

Figure 11: Users testing the application

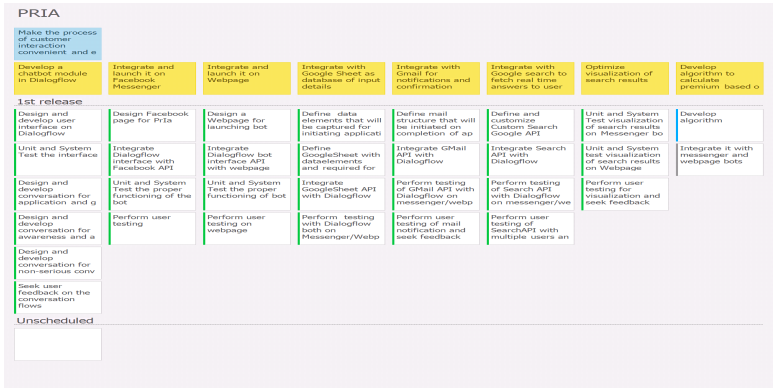

Figure 12: User Story Map and MVP

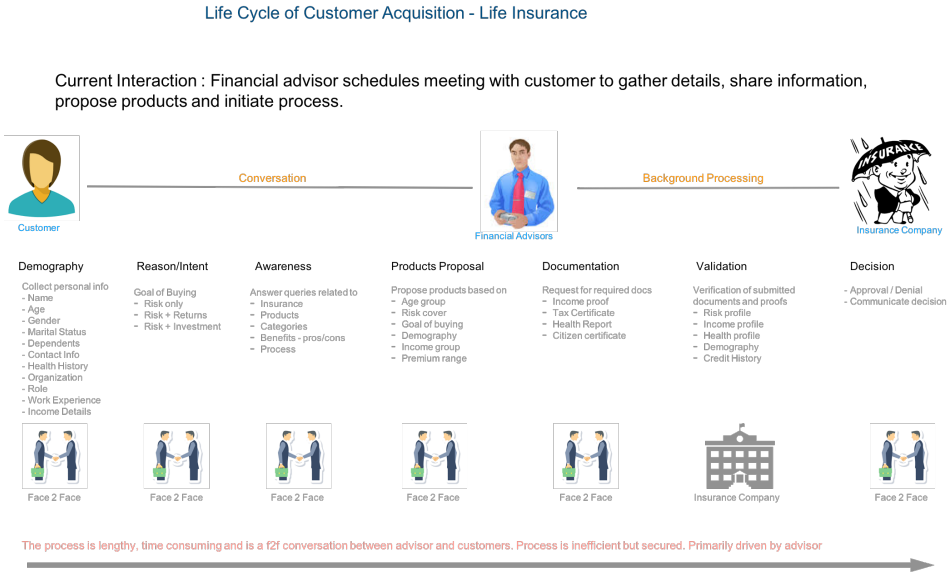

Figure 13: Current Process for Customer Acquisition

Interview Conversation Format.pdf

Project in Digital Media Engineering

PRiA - Insurance Assistant

Problem Statement
People want to buy insurance. But the process of buying an insurance is very tedious, time consuming and lengthy. PRiA is an experiment to develop a new user experience that is engaging, quick and interactive.

Questions for the users:

- User Information
  - Age
  - Gender
  - Country
  - Profession
  - Bought insurance before or not?
  - How well versed with buying insurance?
- Data to be logged:
  - Date of experiment/user interview
  - Time at which experiment/user interview was conducted
  - Place
  - Medium - was it in-person or virtual?
- This messenger is being developed to assist customers who have an intent to buy a life insurance or who wants to know more about life insurance before buying
- Questions that are important before testing the messenger app
  - Have customers bought an insurance before?
  - What was the process through which it was bought?
    - Was it through an interaction with an advisor and then bought?
    - Was it through an interaction with a Customer Service Rep, then an advisor and then bought?
    - Was it through an interaction with CSR, advisor and then submitted everything online?
    - Or it was completely digital - what was the interface and how was it done?
  - How was the overall experience of buying an insurance in a scale of 1-5?
  - What do you prefer to continue from that experience and what would you like to see a change?
  - How was your satisfaction level in a scale of 1-5?
- When customers have a need to buy insurance or want to know more about insurance before buying it, will customers be comfortable interacting on Facebook messenger?
  - Yes or No?
  - And why?
- What is the preference?
  - talking to an advisor in person
  - or talking to an advisor digitally
    - through a messenger on Facebook
    - website messenger
    - or through voice assistant
- If we develop a messenger service to assist customers, will they be willing to test it and provide their inputs?

Once these questions are answered, share the link with the users:

- With the users (who have said yes for messenger), let's share the messenger link and with others the web link
- Let them work with the app without us interfering. Our job will be to monitor user activity and register comments on user behaviour
- Once user is done with using the tool, then have an interview with the user
  - How easy is the interface to use?
  - What problems were faced starting with the interface?
  - Were the responses appropriate?
  - Was the time to response fine? Should it be increased or decreased?
  - The phrases./ the words being used are those appropriate?

Figure 14: User Testing Questionnaire