MyCampus Chatbot - A Conversational Agent that offers information to IST Students

Francisco Maria
francisco.maria@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal
October 2019

Abstract

In this thesis we are going to discuss the creation of a personal assistant for the students of Instituto Superior Técnico, in the form of a chatbot. The objective of the bot is to facilitate the lives of students and make existing information easily accessible. The bot will get all of its information from the FenixEdu API, this information includes degree information, student’s information, courses information, payments, dates to evaluations and even information about rooms. The bot is also able to give the user notifications about course announcements. We also used and tested Google’s Dialogflow as a tool to create chatbots.

Keywords: ChatBot; University personal assistant; DialogFlow; Firebase; FenixEdu API; IST;

1. Introduction

This thesis is going to discuss the creation and development of a voice and text based chatbot to help IST students to get specific IST information. The application is an attempt of making the information already available in the IST app or in Fenix easily accessible.

To go about their normal day, students need access to some information given by the school. From the shuttle schedule, to when and where their next class is, the school cafeteria menu and even the news about Técnico and all that happens in it. To facilitate their lives this kind of information should be given in the easiest and most comfortable way possible.

At this moment this information is all available in the university website, the Fenix website and, some part of it, in the IST mobile application.

The Fenix website is where personal information about the user is and its one of the most used platforms by IST students. However, opening the browser, inserting the Fenix credentials and then search for the information is neither the quickest nor the most comfortable way of getting information, especially if it is simple information. An application that can be easily accessible and that has all of the user personal information is needed.

The IST website is where information about the school is located. This includes contacts to various school infrastructures, information about courses and more basic information.

In the IST mobile application, you can find information like the cafeteria menu, shuttle schedule and evaluation dates. The app can also sync your school calendar with your phone calendar. It is a quicker and easier way to get information, but it does not have as much data as the Fenix website, it misses some information about the students.

So, as seen, there is a lot of information that the students need and it is scattered throughout multiple places. There is a need to link that information to make it easier to access. There is a need for something like a personal assistant, or a chatbot, that knows everything about students as well as IST. Something that would respond to user queries and know the user context to better help him. An app where all the information can be found, from the mail of the Academic Office to the link to the curriculum of a specific course.

2. Related Work

In this section we are going to talk about research that is of value to our project. We will first talk about research that introduces chatbots and their interaction with humans. Then we are going to show some university chatbots. And we will end by showing projects that use Dialogflow as we do.

2.1. Introduction to chatbots and chatbot-human interaction

2.1.1 Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations

J. Hill et al. [7] gave us an insight on how communication changes when people communicate with
a chatbot and when they do it with a human.

To analyse this new form of communication, human-chatbot communication, the authors collected unaltered instant messages, conversations from volunteers of a university, to be the human-human conversational component of the test. For the human-chatbot conversations they used real conversations of Cleverbot, an internet chatbot, given by the developer. In total they investigated seven variables in the messages, words per conversation, messages per conversation, average number of words per message, the percentage of unique words per conversation, abbreviations, emoticons and profanity.

People sent fewer words per messages when talking to bots, but sent more than twice as many messages that when talking to humans. The reason is that people were modelling their communication to match that of the chatbot, to test this hypotheses the authors analysed the data and discovered that humans chatting with Cleverbot average a similar (small) number of words per message as Cleverbot and people chatting with humans average a similar (higher) number of words per message as their messaging partner.

Conversations with the bot had more profanity with almost a 30-fold average increase in profanity and sexually explicit conversations, when talking to the bot.

With this paper we have learned that when communicating people try to recreate the style of communication of their partner, so in our project we have to give a good example of the type of communication that we want with our chatbot. The chatbot has to talk in a simple and correct way if that is what we want of the user.

2.1.2 The relationship between motives for using a Chatbot and satisfaction with Chatbot characteristics in the Portuguese Millennial population: an exploratory study

T. D. Rieke et al. [10] tried to explore a part of chatbot research that is lacking, the analysis of chatbot characteristics and motives for using this technology. To analyse that they made two questionnaires, one about satisfaction with chatbot characteristics and another about motives for using chatbots. The survey was done to Portuguese millennials of the University of Porto, this population and their interests about chatbots are very interesting to this paper because the students of IST are in the same age and conditions.

They start the results chapter by describing the participants of the questionnaire. The questionnaire was answered by 317 people but of them only 258 responses were found adequate. In these responses the number of female participants and male participants were practically the same and their ages were between 18 and 38, but the majority of the participants were under 26 years old.

The first question was about the familiarity with chatbots, as seen in other papers almost all of the students knew chatbots, but when asked if they used chatbots more than 30% said that they had rarely used chatbots or nothing at all. When asked about a chatbot characteristic, that they prefer, the participants seem to be interested in a chatbot that is fast and accessible and not a chatbot with emotions and feelings.

In the end, with this results, they concluded that the Portuguese millennials have some experience with a chatbot, but rarely use one. Contrary to other research that classifies emotion in a chatbot as an important feature, Portuguese millennials prefer a bot with fewer emotions and more speed and accessibility, features like sadness or empathy may cause a decrease in satisfaction with the bot. Efficiency is the most important characteristic for this group. The last finding was that convenience, exploring and social are three big motives, for this group, to use a chatbot.

Even though the population of this questionnaire was of Porto University students we think that IST students may have similar beliefs and that we can learn a lot with this paper. We have learned that Portuguese millennials care about efficiency and one of the big reasons for them to use a chatbot is the chatbot being convenient, which is a goal of our project. Our bot should then be as efficient as possible. It should be a bot that gives an answer in a quick and correct way.

2.2. Similar projects

2.2.1 Técnico Application

The Técnico application [1] is a mobile application that enables users to access some Técnico information easily.

The application shows news from IST and announcements from classes. It also shows the cafeteria menu for the day, the shuttle schedule, parking information, it enables students to take a ticket for the academic office, see courses information, evaluations, the student curriculum and the payments that the student needs to make. The application also allows students to sync their phones calendar with their class schedule.

The application does a really good job in showing some information, but it does not show all of the information that Fenix has available and sometimes students may have difficulty finding what they want.

With this application we have learned a little bit of
what information is available right now to Técnico students and what our application has to do better to help students and faculty.

2.2.2 Ambient Intelligence: The MyCampus Experience

N. Sadeh et al. [11] report about a five years project of the MyCampus group at Carnegie Mellon University. This group has been experimenting with Ambient Intelligence technologies aimed at enhancing everyday life. The project concentrates on developing an open Semantic Web infrastructure for context-aware services with an emphasis on privacy and usability. The project itself was implemented as an application that is aimed at enhancing everyday life of students and faculty, in the university campus, and it uses context-awareness to do that.

They start the paper by talking about the architecture of the project. The system is composed of a number of agents or applications that the user can download. These applications are context-aware and each of them has to access some information about the user and is context to function correctly. This information is stored in a e-Wallet.

The e-Wallet is a repository of static knowledge about the user. The knowledge in the e-Wallet is divided in several categories: Static knowledge, is context-independent, it includes the user name, email and some preferences; Dynamic knowledge is context-sensitive, often involving some preferences like “When in class I do not want to receive messages.”; Service invocation rules, these rules help on how to get external information to the e-Wallet, both personal and public; Privacy preferences, in the privacy preferences we have the information about what the user is willing to disclose to others under different conditions.

The MyCampus infrastructure was conceived to enhance campus life at Carnegie Mellon University and over the years the team has worked to create and refine applications that can help students and faculty in their daily activities. Some of those applications are: Context-aware recommender services, include recommender services for places where to eat, nearby movies and public transportation; Context-aware message filtering services, allow to specify when the user wants to see different types of messages based on the nature of the message; Context-aware reminder applications, applications that remind users about tasks they have to perform in relation to their location and other contextual attributes; Context-sensitive “crime” alert applications, warn users when they are entering an area more prone to incidents; Collaboration applications, Enables people to selectively share PowerPoint presentations; Community applications, these applications are extremely popular and can go from mundane calendar scheduling and people locators to more sophisticated virtual, context-aware poster applications.

In the end the authors concluded that in this type of context-aware applications the user privacy issues are very important and that capturing the preferences of the user is a major impediment to the broad acceptance of ambient intelligence technologies.

This paper is relevant for our project because it shows how context-aware apps work inside a University Campus and gives ideas of useful apps and features using context-awareness for our project. It also tries to make these context-aware applications to help and better the lives of students and faculty in campus, a key part of our project.

2.2.3 Chatbot in a Campus Environment: Design of LiSA, a Virtual Assistant to Help Students in Their University Life

M. Dibitonto et al. [5] did some research to then implement LiSA a chatbot that helps students in their campus life. With this research they want to understand which information and services are better accessed by students and how the chatbot personality influences the user experience and interaction. To do that they did a survey to identify the users’ needs and their inclination to use a chatbot.

A chatbot can do a number of services but in this paper they want to know what kind of services are most desired by students. If it should provide services like booking an exam or paying annual fees, if the messages should be delivered in pull and/or push mode. In pull mode the chatbot responds to questions by the user but in push mode it gives information to the user without any input. Another big question is what kind of personality should the chatbot have.

They then do a background introduction where they talk about what is a chatbot and why is it used, the conversational interface of a chatbot and the user experience. They have an example on this topic where they say that the use of the users first name during a conversation with the chatbot is perceived as positive. In the end of this section they conclude that a well-designed chatbot has to have a personality, has to establish a relationship with the user, understand the context and hold a conversation about general arguments. It should not try too much to imitate a human and should be able to overcome users’ misspellings, other mistakes or rude comments.

In the next section they presented how to design a chatbot directed to university communities,
They discovered that when asked a sensible per-
different behaviour during a conversation with a
when talking to the questionnaire chatbot. It has
authors give information about the user’s behaviour
toing to the university.
ferred tools are email and phone, this is because it
ary issue due to the possible changes, the pre-
human; the first one is a chatbot that
portant events coming soon. Pull information can
communication tool, not suitable for communica-
tion is preferred to ask information. Another high-
son requests which means that personal interac-
to give this questionnaire they used LiSA a low-level
dialog system chatbot built on the Facebook Mes-
When asked about what tools they use to com-
unicate with the university the highest satisfac-
tion rate comes from phone calls followed by in per-
son requests which means that personal interac-
tion is preferred to ask information. Another high-
light from the questionnaire is that students do not
use Facebook to get information, the main reason
for that is that Facebook is considered an informal
communication tool, not suitable for communica-
tion with the University.

When students were asked how they usually
look for several kinds of information they saw that
enrolment was the first reason for students to ask
for information and that they prefer to get it by
phone or in person. Class schedule is another pri-
mary issue due to the possible changes, the pre-
ferrred tools are email and phone, this is because it
allows students to get this information without go-
ing to the university.

In the last section of the questionnaire the au-
ths give information about the user’s behaviour
when talking to the questionnaire chatbot. It has
been shown, in other papers, that people have a
different behaviour during a conversation with a
chatbot. In this study they found the same thing.
They discovered that when added a sensible per-
sonality, hurt by aggressive behaviours, some stu-
dents apologised to the bot and others tried to ex-
cuse their behaviour with the curiosity of testing the
chatbot limits.

In the end they say that users would find useful
to have the chatbot push information to them, es-
pecially class schedule changes. They would also
like information about University’s events and ac-
tivities that could be personalised to student’s in-
terests. Exam booking and enrollment information
it is also going to be taken in to account.

This paper is relevant for our project because it
gives a really good perspective of what students
want of a University personal assistant. The ques-
tonnaire was done in a different University with a
different campus and different information, but we
can still learn with all the answers and conclusions
that they take. It helps us to know what to use and
not to use in our chatbot.

2.2.4 Evaluation, development & testing of an
educational support chatbot

B. I. Systems et al. [12] determined if there is a
use for chatbots in a university, specifically Cardiff
Metropolitan University, where all of the ques-
tionnaires and data came from. In this paper the author
shows all of the data collected in is questionnaire
to students and staff and then shows how he im-
plements the chatbot and all its functionalities.

The first survey done in this paper was to the
students, in total 100 students took the survey in a
week coming from a total of 27 different courses.

This survey had two main goals: know if students
are satisfied with university communications and if
they would adopt chatbot technology as a solution.

From the survey they found that: to communi-
cate with the university students prefer email and
then face to face communication. This shows that
this students only communicate when they are in
the university, most of them are only in there 3 or
2 times a week. This is a relative limited commu-
ication time for students who communicate face
to face; the most frequent question that students
need to ask is assignment queries and a lot of them
are simple ones, like word count, due date and
marking criteria; in terms of communication plat-
forms, the most used is WhatsApp but this kind of
platform does not have a chatbot technology. But
94% of students have Facebook Messenger in-
stalled which has an open chatbot platform; 60%
of students said that they would feel comfortable
communicating with a chatbot to receive the same
information they would get from a human; the fi-
nal question was about chatbot personality and if a
chatbot should have one. Students were split be-
tween a yes and a no/don’t care, so the author con-
cluded that students do not want to be fooled into thinking that they are talking to a human when they are not.

With this data the author concluded that the survey was favourable to the creation of a chatbot for the university. Students were comfortable with communicating with a chatbot, had a good platform to create a chatbot already installed on their phones and seemed responsive to the idea. They concluded that Facebook messenger is the best platform to have a chatbot. That the chatbot could help staff receive fewer emails a day and that the chatbot must have information from the university website to make it easier for staff that have difficulty using the site.

The system itself is all built using Oracles IBCS and Facebook Messenger to deploy the bot. The login is made in Moodle, the tool used by the university. After the initial greetings the boot asks the user login info and with that he establishes a session with Moodle through Moodle’s API. After that the user can make the questions he wants, the bot responses are in two sections, a Q&A (questions and Answers) section made up by all FAQs (Frequently Asked Questions) found on the university site and the data that can be taken from Moodle via the API. When the user inputs a question that the boot cannot be sure in which intent it sits it gives intent options for the user to choose, guaranteeing that the user gets is correct answer. When the user asks broad questions like “can I have assignment feedback” the boot also gives him a list of more specific topics, in this case it gives a list of the user assignments. One interesting feature that the authors added is the possibility to talk to a human if the bot does not understand the question. In this case the bot asks the user if he wants to talk to a human and if he wants then the question will be placed to a human. Lastly a user can only be logged in 5 minutes before it as to re-authenticate. This feature was implemented because the access to Moodle gives access to a lot of sensitive information.

In the end they talk about the Facebook platform, that gives a lot of tools to get detailed analytics and add-ons to, for example, access web payments.

In this paper we see another example of a chatbot for a University, but unlike other solutions this chatbot also helps staff and takes them in to account. This solution gave us different ideas about student/staff preferences and features to have in a university chabot, like different options if a question is vague. It also gave us an insight into the development of the bot, even though it was on a different platform that the one we are going to use, it was very helpful.

2.3. Architectural similar projects
In this section we show three projects that use DialogFlow to create their chatbots, the same tool that we used to create ours.

2.3.1 Conversational modelling for chatbots: current approaches and future directions
M. McTear [9] examined the conversation models of some chatbot frameworks. We are going to pay more attention to DialogFlow, the framework we are going to use. The author also assesses if it models important conversational phenomena like follow-up questions, changes of topic and out-of-scope utterances.

The current frameworks extract intents and entities from the user’s queries. An intent is something that the user wants to achieve and an entity is a parameter that is required to fulfill an intent.

There are two main types of conversational interaction, one-shot queries and slot-filling dialogues. One-shot queries are user-initiative queries. They are initiated by the user and are simple questions/commands with no context and a simple answer/action. Normally there are no follow up questions and no subsequent questions by the system; slot-filling dialogues are system-initiative, are initiated by the system and can have some context. For example, the system can ask if a person wants to travel and then if they say yes, asks were they want to travel. In this case the system maintains a context of the conversation. A more natural way of conducting a conversation is in a mixed-initiative way. After the system asks a question the users asks another question in the same context. But most of the systems do not allow this type of conversation. Follow up questions can be asked if the system does not understand the query or if does not have the intent needed.

After this explanation of conversation models the author compares some frameworks. We are only going to talk about DialogFlow. DialogFlow allows some of this behaviour, it allows follow-up intents to be added to any intent and provides some built-in follow-up intents. It sets up an output context for a parent intent so that the context can be maintained between follow-up intent and parent intent. Slot-filling is also guaranteed, DialogFlow uses a parameter table were the required parameters and their associated slots are specified.

This paper is a good introduction to conversational models and too chatbots in general, it gives us a basic understanding on conversational models and in the end it also has a topic on difficulties that developers have in chatbot creation. It also talks a bit about features in DialogFlow, the tech-
nology that we will use.

2.3.2 Developing an Intelligent Chat-bot Tool to assist high school students for learning general knowledge subjects

D. Dutta [6] describes a chatbot tool for high school students and how it was built. This tool will answer questions about general knowledge. At this moment to get an answer to their questions students have to google the information, read links and papers until they find the answer they want. With the chatbot the process will be simplified.

In this paper the authors analyse and test chatbot platforms to find the best one. The platforms analysed were Dialogflow (API.ai), Wit.ai, Luis.ai and Pandorabots. To test these platforms, they trained all of them with the same knowledge and then tested with the same input. From these tests they concluded that Pandorabots matched the least number of intents. In the case of false intent id matches, Wit.ai and Luis.ai scored a confidence score higher than Dialogflow and in this test a lower confidence score is preferred. They reached the conclusion that Dialogflow is the best tool based on the tests. Dialodflow also has a feature where the developer can add follow up intents.

After selecting the platform to build the chatbot, they explained how to build it. Dialogflow is a platform that has easy integration with a lot of platforms and devices. In this case the authors choose to make an android app and a web platform. For that they used node.js to build the server, BotUI JavaScript framework to create the interface and DialogFlow HTML and Android SDK to learn and know how to contact the DialogFlow API. They also used a feature of DialogFlow that makes the developer able to easily add pre-made small talk intents to the chatbot, this enables the chatbot to answer questions like how are you? making it a little more interesting for the learners.

With this paper we learned that a chatbot is a good tool to teach and get knowledge from, we also learned that the DialogFlow application is the best to make chatbots and learned more about developing chatbots in this app and how easy it is to do it.

2.3.3 TA-bot: An AI agent as a Teaching Assistant

TA-bot [8] is a chatbot developed to be a virtual teaching assistant with the objective to complement the human staff and enhance the learning experience. This bot is part of a family of bots where each bot has a different specialisation. TA-bot specialisation is to answer questions from the syllabus.

In this paper the authors describe the architecture and design of the chatbot.

The architecture of this chatbot is composed of three tiers, the first one is the front-end used to interact with the bot, the second is the processing tier that handles the natural language processing and the final tier is the back-end that contains the knowledge of the bot. The front-end is in a Facebook Messenger application. The second tier is DialogFlow. The questions are going to go from Facebook to DialogFlow that is going to use its built-in natural language processor to process the data. The back-end uses Google App Engine and Google Big Tables. They use the Google infrastructure to simplify onto a single stack vector. Google DialogFlow also supports different integrations like Facebook Messenger, Telegram, Slack and more.

DialogFlow works with intents and entities. An intent is an intention of a user and in DialogFlow the developer can map the intent of a user to an action. In this case some of the intents of the chatbot are going to be, learning about deadlines for an assignment, inquiring about where the course meets and many more. An entity is anything that can be referenced like a textbook, the course, the classroom and so on.

There are two types of knowledge in the chatbot, “stable” knowledge like who teaches the class, where it is and “dynamic” knowledge, knowledge that changes every semester. They decided to make this part of the project in a separate back-end because it overcomes limitations in the processing tier, more complex queries can be made and the App Engine provides an easier interface to maintain and modify entities. In the App Engine, entities have a simple model, they can have some text-based information, some numeric information and/or date information.

In the end, after developing and testing, the authors concluded that a virtual teaching assistance can be created quickly and easily with out-of-the-box technology.

This paper is of great use to ours because it gives as an insight on how the architecture of a chatbot should be. They use the same tools as we are going to use and gave us a better understanding on how to use them and what to use to create a chatbot using DialogFlow.

3. Proposed Solution

In this section we are going to talk about the bot we created and how we did it. We will start by naming the features of the bot. Then we will explain the architecture and how we made it.
3.1. Features
The objective of this project was to make a personal assistant that can help and improve IST students university experience. The project was set to use the student’s context to give them needed information. Although more context information was set to be used, due to time constrains and problems with the information, we had to restrict it to student’s information given by Fenix. Like their schedule, courses, payments, curriculum and rooms information. With this information and more, given by the Fenix API, we were able to make more that 20 features.

3.2. Architectural Overview
In this section we are going to talk about the technologies that we used to develop the chatbot. Most of the technologies will be services by Google, but we will also use the Fenix API to get information about the students. The figure below shows how the project communicates between the different used tools.

![Figure 1: Tecnico Assistant architecture.](image)

As seen in the figure the architecture of the project is composed of the Google Assistant app, Dialogflow, Firebase and some external APIs:

**The Google Assistant app:** Is our interface and is where the user inputs requests. It then communicates with Dialogflow by sending what the user typed.

**Dialogflow:** Will get the answers to the user. Dialogflow job is to interpret the user query and infer the right way to treat the query, when it does, it sends information about which code to run and the information that the code needs to Firebase. It receives the response that has to give to the user by Firebase.

**Firebase:** Where the code is located, with the information from the user query and information that is requested to different APIs, the code in Firebase generates an answer to give the user. Firebase also has the database.

**The external APIs:** FenixEdu API and Google Cloud Scheduler. These APIs have endpoints to be contacted in order to get information, the information is returned in a JSON object.

All of these technologies and how we used them in our project are going to be explained in more detail in this section.

3.2.1 Dialogflow
Dialogflow is a Google platform that enables users to build voice and text-based conversational interfaces. The platform has all the logic implemented to handle the human-computer interaction. A Dialogflow application first starts with an agent that comes with default fallback and default welcome intents. The agent helps in processing user input into structured data that can be used to return an appropriate response. The developer only has to fill it with intents. It was because of this lack of needing to worry with human-computer interaction logic, and the ease to create intents that we chose this platform.

We first started our app by creating the agent. We then filled it with intents, we made a total of 40 intents, each of them resolves a different query by the user or responds to an app event. For example, the getNextPayment intent is triggered when someone asks for their next payment, while the setupPush is triggered when a user wants to receive notifications. An intent defines how to map user input to a corresponding response. This is done with the help of the training phrases, a section of the intent where examples of user queries are put. With these examples the agent can see which intent training phrases is the closest match to the user input.

When handling queries by the user we may need some information from the query, like the shuttle station or the schedule of the shuttle. This type of information is called an entity. Entities are also specified in the training phrases. An entity can be system specific like @sys.date or can be created by us like @shuttleStation. For example, in the sentence ‘when is the shuttle in alameda tomorrow?’, alameda would be a value of the entity shuttleStation while tomorrow would be a sys.date entity.

We used system entities like date and any. The entity any matches everything, it does not have a specific value. We also created 6 entities.

Our six entities are: contactos, that has the information available in a IST service, like fax, phone, postalCode etc; degree that has all of the IST degrees; estacionamentos, that has the IST parking locations; IST, that has all of the different ways that a student can refer to IST, for example Técnico, IST, Instituto Superior Técnico, this was used to facilitate the writing of the training phrases; istPlaces has all the IST services that a student can access information of; and shutlleStations has all the shuttle stations.

In an intent there is also a field called event, in this field an event name can be put and the intent will trigger when that event is called. Events can be of the system or created by the programmer.
In our intents list we have intents that only exist to respond to system events, these intents have no training phrases. These events are UI specific events, Log in events, push notification events and the exit event. Some intents also have events created by us so we can call them. For example, when a user clicks an option in the available information list an event is triggered to call the intent the user wants.

To connect the intents to the code that resolves them Dialogflow has Fulfillments. With the Fulfillment the code is hosted elsewhere. To contact this code Dialogflow sends post requests to a webhook and the webhook responds in the same way, with the final response for the user.

### 3.2.2 Firebase

Firebase is a Google platform that offers several cloud services like cloud storage, authentication services, website hosting and more. This is where all the webhooks code and data is, we chose this platform because it is an easy integration with Dialogflow.

In Firebase we first used Cloud Functions, a service that allows us to host and execute code in the Google Cloud. With Functions and the Actions on Google Client Library is easy to treat the Dialogflow post requests, these requests come with a JSON object. The Actions on Google Client Library abstracts that part of the request and enables us to treat the requests easily.

The other service of Firebase that we used was Cloud Firestore a NoSQL database in the cloud. In the database we inserted information about the users. Information needed to send notifications to the users and IST spaces information. With this NoSQL database we found an easy to use database that is already integrated in the library we were using.

All of the code in Firebase in node.js

### 3.2.3 Fenix API

Fenix is where all of the student’s information is located so we use the FenixEdu API to get this information, the API itself is a list of endpoints. To get the information in an endpoint a get request must be made, in response we obtain a JSON object with the demanded information.

### 3.2.4 Google Cloud Scheduler

The Google Cloud Scheduler is used in our assistant to run a task at a specific time. We use this platform to once a day send a get request to a specific endpoint in our Firebase Functions. This endpoint is used to send notifications to all of the users that ask the assistant to receive them.

### 3.2.5 Google Assistant

Google Assistant is a personal assistant made by Google, it can research on google, talk to the user, see the weather and much more. The assistant is available in Android phones, iPhones, computers and many more Android and Google devices.

We used the Google Assistant as our user interface because it had an easy integration with Dialogflow and our intents, it saved us time developing an app to be our UI and it is available in a number of devices that our users use, the IST students.

### 3.2.6 Actions on Google

Actions on Google is a platform that enables developers to make apps in the Google Assistant. An Action is support for a user request, it is represented by an intent and his fulfilment [2]. When we integrate our Dialogflow project with Google Assistant our intents became an Action in the Actions console.

As mentioned before we also use the Actions on Google Client Library, this library enables us to use different features of the assistant like certain UI elements, Log in and interact with Dialogflow.

### 4. Work Evaluation

To evaluate our system we decided to first test its performance, comparing to other IST technologies, its usability, using the SUS, and get the user opinion on it. We made the test to 20 users, all IST students. To test the performance we gave the testers some tasks to complete, from there we collected metrics like the time to complete a task and number of errors the user made. In the end the user answered a questionnaire where we used the SUS and other questions to test the usability of the system and the user satisfaction.

#### 4.1. Tasks

We first started the test by giving eight tasks to the user to complete. First on the chatbot and then on another IST technology. We can make 25 different tasks with the features of our chatbot, but to make the test less lengthy we choose the eight that we thought were more important and represented the chatbot: find the next shuttle in Alameda; find the user next payment; find the announcements of a course; find the next evaluation of the user; find the schedule of a class room; find the final grade of a user course; find the user next class; and find the user IST email.

All tasks were given a start point and an end point, so that we can register the time they took. The tasks were first done on the chatbot. The other
IST technologies were the IST website, for the first task, and the IST Fenix website for the others. To evaluate the chatbot in these tasks we recorded two metrics: Time to complete the task, the time that the user took to complete the task; Number of errors, the times the task had to be started over.

With data from the times the testers took to complete the tasks we calculated their average and standard deviation. This information is in the table bellow.

<table>
<thead>
<tr>
<th>Task</th>
<th>Chatbot</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>11.69 ± 3.51</td>
<td>23.69 ± 7.82</td>
</tr>
<tr>
<td>T2</td>
<td>8.32 ± 2.03</td>
<td>13.82 ± 5.78</td>
</tr>
<tr>
<td>T3</td>
<td>16.18 ± 5.96</td>
<td>7.94 ± 2.76</td>
</tr>
<tr>
<td>T4</td>
<td>9.62 ± 3.89</td>
<td>10.17 ± 2.39</td>
</tr>
<tr>
<td>T5</td>
<td>11.57 ± 3.79</td>
<td>22.98 ± 13.50</td>
</tr>
<tr>
<td>T6</td>
<td>13.19 ± 3.47</td>
<td>16.44 ± 6.96</td>
</tr>
<tr>
<td>T7</td>
<td>8.05 ± 2.90</td>
<td>8.66 ± 3.11</td>
</tr>
<tr>
<td>T8</td>
<td>8.27 ± 2.56</td>
<td>11.10 ± 5.18</td>
</tr>
</tbody>
</table>

As seen in the table, the chatbot has better times that the alternatives, except in task three, we can conclude that the chatbot is quicker than the Alternatives.

Regarding the alternatives times, we can see that the quickest tasks, task three, four and seven are tasks that the student is more used to do. These tasks are automatic for the students and are easily done. Task one, was the only one done in the IST website and the one that was more hidden and harder to get to, some students said that they don’t even use the IST website to find this information but Google. We can also see that task one and five are the ones with higher standard deviation, this is because some students knew were this information was and others not.

Regarding the chatbot times, the one that was the slowest and the only one slowest that the alternative was task three. This can be because if a student does not specify the course that he wants to know the announcement options appear for him to chose the course.

About the errors, almost all of the chatbot tasks had at least one error, this was because some times the chatbot did not understand what the user wanted from his question. As we said before the chatbot is trained with example phrases, and if the users question was too different from the example phrases, the chatbot said that he did not understand the question. This type of error only appeared in the begging of the tests because as soon as they appeared we trained the chatbot. Another type of error was when students used speech to talk to the bot and it would understand them badly.

The errors in the alternatives to the chatbot were all because the students had difficult reaching the information and had to take several paths before reaching it. As we can see the number of errors in the alternatives are greater that in the chatbot, and the errors are in tasks that the students don’t user very often or are hard to find the information.

With the data from the times and the errors we can conclude that our chatbot preforms better in tasks that the user uses less often, as it gives an easy way to find the information, while in the alternatives the information can be hidden and hard to find.

4.2. Questionnaire

After completing the tasks, the students were asked to respond to a questionnaire. The questionnaire was divided in three part.

The first three questions were about there sex, age and the number of years they are enrolled in IST. All of the students that made the test were between 21 and 27 years of age, they were all doing their masters but their knowledge and experience of IST varied, as some students only had been there for one year and others over 5.

The next section of our questionnaire was aimed at testing the usability of the system, for that we used the SUS [4]. The SUS is a 10 questions questionnaire with five responses each, that goes from Strongly agree to Strongly disagree. This system as references in over 1300 articles and publications and gives a reliable result in small sample sizes, as ours. To interpret results the system has a way of transforming the results into a grade, going from F to A [3]. After doing the calculations our system ended up with a B grade, this means that our system does not have usability problems but still has room to improve. We can also take from the first question of the SUS, I think that I would like to use this system frequently, that the students would like to use the system as they all responded with a 4 or 5, being 5 the closest to strongly agree.

The final part of our questionnaire was done to assess the chatbot and what the students liked about it. We started by asking them if they prefer to use the chatbot or the alternatives, all testers answered that they preferred the bot. This tells us that, for the students, the chatbot is a better and more comfortable way of getting information that the alternatives.

The second question gave a list of all of the chatbot features and asked them to choose their favourites. The three favourite, with 80% of the testers selecting them were: give the time of the next shuttle; give the students next class; give the students next evaluation. These three features are the ones that impact the students academic life the most. This tells us that the information the students
like the most are the ones that they need every day and is important to their academic life.

The third question was optional and asked the tester if he thought that was something missing in the system. One tester said that a FAQ section was interesting, another asked for access to more information and a last one said that it would be interesting if the chatbot was, in the already available, IST mobile application.

The last question asked the testers if the chatbot functionality was something important or if they would prefer an application with the same information but without the chatbot. Only 2 students said that they would prefer something without the chatbot and 18 said they like the chatbot.

With the questionnaire we can conclude that the students liked the chatbot and they would use it. We learned that information that affects the student daily life is more important and the students actually care about the chatbot technology and not only the information available.

5. Conclusions
A student needs a lot of information to carry his normal academic life, in this project we tried to create an application that gives needed information to students in a more comfortable and quick way.

We used Dialogflow and other Google technologies to develop the chatbot. We had more difficulties with Dialogflow at the start while we were still learning how it worked, it can be a tricky technology to understand when we need to navigate through countless documentations and there are many ways of doing the same thing. But after learning how it works, doing intents and new functionalities on the chatbot can be easy. Integrating with Fenix was also something difficult, especially trying to integrate the Fenix log in in the Google Assistant and the Google log in system.

Dialogflow also understood the testers really well, matching the questions of the users to the right intents, even if they were slightly different from the training phrases. It only had problems when two intents were similar and when the any argument was used, something that should be avoided. In the end Dialogflow was a good an easy way of developing a chatbot, with it we only needed to do the connection between it and the information.

After concluding the development of the system we tested its usability, performance and user opinion. The tests were very positive, our system is great in terms of usability and is performance better that the alternatives that exist. By analysing the results we saw that even with the users favourite features, of the system, being the ones closest to the alternatives in performance, the users still preferred our system. This tells us that we achieved our goal of having a more comfortable way of giving information to students, and that they prefer a chatbot to give them information they use daily and not only information that they don’t know where it is.

In the end we were able to develop our system and deliver a product that the students of IST liked. We could also conclude that we reached our goal of developing a system that gives information to students in a more quick and comfortable way.

References