Students

Al-Hubaishi Waleed iCompetence

Kaminsky Jens Informatik

Coaches

Kalunder Madlaina  
Agotai Doris

How to create a cool App <todo>

**Swiss Engineering  
Event App (SEEA)**

IP5 Documentation 2019-XX-XX

# Summary

The goal of this project is to create an app for the Swiss Engineering Association that will attract new members. To achieve this goal, a voice recognition/generation feature was suggested. This Document will summarize how such a feature could be implemented, what is currently possible and even what might be possible in the future by looking at cutting edge research that is currently being done in this section.

# Abstract

This project is focused on how to transfer the Swiss Engineering Web platform booking events process into a mobile app. Where the database used for both the web platform and the app must be the same.

The objective of this project is facilitate booking events process using the apps, so that the Swiss Engineering members can totally rely on it to manage their booked events and explore what is available on the platform. The app has been implemented using XAMARIN Cross Platform app developing tool to ensure that no matter which device the members have in their pockets; whether iOS or Android may enjoy the intuitive experience of booking, managing and exploring the events.

The mixture of working with Agile methods while focusing on the usability experience (UX) resulted in an intuitive to use graphical user interface, where all important elements related to the core features of the app are well presented in the home page screen, so that the end-users, no matter what age group they belong, or what technical affinities they possess can enjoy the app and use it without any need for tutorials or help on how to navigate through the different app screens.

# Management Summary

At the beginning I focused on understanding what elements does XAMARIN offer us the developers to deal with and what limitations do exist. Then the decision were made, considering the Personas, to offer the simplest type of designs that the Persona can deal with and have no problems understanding how to interact with the elements and navigating through the available screens. Therefor the main screen includes all the core features buttons, distributed into two groups, while the main feature of the app, which is the built in assistant feature, is represented at the top in full width.

Another important part was to find a suitable assistant service to accomplish the tasks requested, and as the options were limited from the start to choose from three candidates, which are the major player nowadays on the market, namely Google Assistant, Apple Siri and Amazon Alexa, we have decided to go with the Google Assistant service for a number of reasons, the most important may be the ease of use, and the huge infrastructure, resources and the tutorials available and invested from Google to develop their service.

Using both XAMARIN and Google Assistant, simple user interface and logic behind it, this Swiss Engineering Event App was created. The app offers an intuitive workflow to accomplish the end-users goal, and contains simple user interface elements, that the user must know already its meaning and what is its purpose, based on end-users previous experience dealing with the other apps.

The workflow starts from the main screen and user is asked to navigate back to the main screen in order to start working with another feature, while this gave the simplicity element, but I would personally rather have mixed structure instead of a linear workflow. The end-user needs three clicks at max on the back button to go back to the main screen, or as an element of easiness, a click on the logo of Swiss Engineering should bring the end-user back to the main screen as well, although this is not really a mobile application approach and more web-sites, but it was a necessity to implement it this way, in order to really use every single element displayed on the screen.

The requirements gathering process didn’t include the end-user from the Swiss Engineering Organization, but a sample of potential end-users who are indeed engineers from different fields and backgrounds were consulted in order to know what do they really need in an event app, and what information do they count on to make a booking decision. Nevertheless for a future update, the Swiss Engineering end-users should be consulted and be involved to refine the design and data offered.

An extended list of features should be also added to the built in assistant to expand the services it offers, and not only to save tags and view answer simple questions, but to book apps directly for instance.

A major limitation that was faced during the implementation and designing phase, is the lack of an appropriate data base structure to include all the new attributes that must be saved for each end-user, for example the liked tags and a well-defined attributes information, as the content of the location attribute was not only the actual address, but an inconsistent address information such as only the name of the location, and sometimes the street name and number without postal code. So a re-built database should make the output and displayed information more reliable on the app.

Another major flaw, is that the process of reserving events is too inconsistent at the moment and varies from an event to another, sometimes the user should contact the organizers to book and sometimes the end-user may be able to book the event directly, therefore a consistent process of booking will definitely benefit the app in the future, so that the end-user might book an event directly without having to navigate to the Swiss Engineering website to see how to proceed with the booking process.

**Revisions list**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Comment** | **Author** |
| 0.5 | 26.04.2019 | Initial Version with the template elements | Jens Kaminsky |
| 1.0 | 28.04.2019 | Technical Research & a part of the introduction | Jens Kaminsky |
| 2.0 | 16.5.2019 | Abstract, Management summary, state of the art, methodologies, topic in depth and completed the introduction. | Waleed Al-Hubaishi |
| 3.0 | 16.05.2019 | The Concept part has been added | Waleed Al-Hubaishi |

Table of Contents

[1. Introduction 6](#_Toc8941572)

[2. Research (FACTS) 7](#_Toc8941573)

[2.1 state of the art 7](#_Toc8941574)

[2.1.1 ASR 7](#_Toc8941575)

[2.1.2 Visual-only recognition of normal, whispered and silent speech 8](#_Toc8941576)

[2.1.3 Multilingual speech recognition with a single end-to-end model 9](#_Toc8941577)

[2.2 Methodologies 10](#_Toc8941578)

[2.2.1 Feature extraction 10](#_Toc8941579)

[2.2.2 Voice Classifier/Identifier 11](#_Toc8941580)

[2.3 Topic in depth 13](#_Toc8941581)

[2.3.1 Evolving search engines 13](#_Toc8941582)

[2.3.2 Communication in service providers 13](#_Toc8941583)

[2.3.3 Voice biometrics as authentication 13](#_Toc8941584)

[2.3.4 Smart Assistant 13](#_Toc8941585)

[2.3.5 Forensic Department 14](#_Toc8941586)

[2.4 Technical research 15](#_Toc8941587)

[2.4.1 Google assistant 15](#_Toc8941588)

[2.4.2 Siri Assistant 17](#_Toc8941589)

[2.4.3 Alexa Voice Service (AVS) 18](#_Toc8941590)

[3 Concept 19](#_Toc8941591)

[3.1 Identifying the Persona 19](#_Toc8941592)

[3.2 Requirements gathering 20](#_Toc8941593)

[3.3 Features formulization 21](#_Toc8941594)

[3.3.1 Brain storming session 21](#_Toc8941595)

[3.3.2 Feasible/Filtered ideas 22](#_Toc8941596)

[3.3.3 SEEA final features list 27](#_Toc8941597)

[4 Prototype 28](#_Toc8941598)

[Implementation 28](#_Toc8941599)

[Analysis & future thoughts 28](#_Toc8941600)

[attachments (sources, literature) 28](#_Toc8941601)

[literature references 28](#_Toc8941602)

[honesty policy 28](#_Toc8941603)

# Introduction

Swiss Engineering is an association that spans over multiple professions and has a total of around 13’000 members in Switzerland. They are holding over 100 events annually over multiple of their Sections. These events are not only for current members also have the additional benefit of attracting new future members.

The Swiss Engineering association is currently in the process of updating their event booking website. To reach a greater audience they intend to also have a native mobile app to make booking on the go easier. This application would also be showcased on events with the idea to improve the image of the Swiss Engineering by showing that it has a modern app.



The IP5-Swiss engineering event app team aims to build an app with a simple UI design that helps the Swiss Engineering members to accomplish their goals at a glance and reach the needed piece of information within a sequential and logically ordered number of screens, so that no tutorials are need to know how to use the app and also all Personas can use the app intuitively no matter which technological affinities they possess.

The idea is not overload the app with a ton of features, but to keep the app simple and direct, and present only the features that the Persona wishes to see and have in an event app, so that the Persona keep focus and concentrated and not easily distracted by the UI elements.

In order to achieve that, the IP5 team used the UX knowledge that were gathered during the semesters to design a clean and neat user interface that accommodates to the user experience with the other apps that the end-user used during the years.

Another important element was considered while designing the app which is number of clicks needed to accomplish a task, so that is why the app has the advantage enabling the end-users to reach their goals in a less number of steps compared to the Swiss Engineering website.

All of this was done by offering the features requested by the client and also a sample of potential end-users who are indeed engineers and contributed to the success of the app design by giving their list of features they wish to have and also the type of information that could made up their decision on whether or not to book an event.

Also in order to improve the app functionality, a number of features were added to help the users get the most out of the app, such as the tagging system, which helps to recommend events to the members, by receiving the input of the built in assistant and convert it into suggestions and recommendations that might interest the members.

Last but not least, the social aspect of the Swiss Engineering organization was also addressed by offering a feature inside the suggestions system which helps the members to meet familiar faces and make new friends who shares a similar taste and interests.

# Research (FACTS)

After presenting what is the team of SEEA aiming for, now is the time to start exploring what is available on the market, and what are the latest technologies used on the field to accomplish one of the most important features in the project, which is the built in assistant.

In the following chapters the latest developments to the Voice Recognition field will be discussed, and how could those benefit the app greatly.

## state of the art

This chapter includes the state of the art algorithms that are used nowadays in the Speech Recognition field.

Some of the algorithm that were discussed and explored are not related to the goal that the voice assistant in this project must accomplish, rather an interesting algorithms that might be added in the feature to extend the algorithm used in the project and add extra value and functionality.

One of the major algorithms that were interesting to read about and search for was the Lips Movement speech recognition as it will provide the opportunity to implement the search event feature for instance via speech recognition, where the end-user must not say the words out loud when the situation is not suitable for that.

### ASR

the idea of Automatic Speech Recognition ASR is to translate the input voice into a written text. And because of the recent development, such algorithm exists nowadays on the majority of smart phones via the built in voice assistant apps such as Google assistant for Android devices and Siri on iOS devices, not only that but on every single smart speaker that contains the voice recognition such as amazon Alexa.

Those ASR model has benefited greatly from the development of deep neural networks, yet there is still a major challenge ahead, such as dropping the out a big part of the training data, and had therefor hard time sometimes recognizing a word when the training data dropped out contains the set of parameters containing that word.

Data augmentation has helped a lot regarding this matter, the same way it helped before in the process of classifying images. The way it is used in the process of ASR is to extend the number of data used to train, not by adding more learning data by adding new files that size might be large, but by manipulating the existing data to offer another version with but with different parameters, for instance the voice can be used as is to provide one learning data, then speeding up or lower it down to provide two different versions of the same data used.

This trick may be applied the same way it is applied in the image processing and classifying, by not manipulating the image itself, or the voice in our case, but manipulating its corresponding spectrogram to create a different version of the same uploaded file that will require no more extra space. The results were surprisingly good, and effective.

A screenshot of a cell phone

Description automatically generated

Figure 1 training data with two other versions created from it

### Visual-only recognition of normal, whispered and silent speech

Another interesting state of the art algorithm that might be helpful to use in the noisy or in contradiction the loud places is the visual speech recognition.

The Idea behind the visual speech recognition is to not use the voice to recognize the words but rather to use the lip movement, that would absolutely not help to accomplish the goal of this project but it was a really interesting idea to discover.

As you have expected this model will not use the microphone or audio files to get its learning data but it will use the video instead to detect the lips movement and link it to the desired word, but as the header indicates, this model is not about only about silent speech, but also whispering and talking out loud as well as talking normal.

This algorithm requires that the 4 different models corresponding to each of the 4 possible context should be included in the same algorithm, and therefore the AI should learn each word 4 times. The question raised is whether or not could the AI return the correct interpretation of the movement when the mode(context) is not specified while testing.

The biggest challenge to implement this model was that the lip movement changes according to the context in which it is said, for instance the word “success” would have four different lips movement model, depends on whether it has been said silently, whispered, normal speech or said out loud. There is still a big chance that a word might be confused with another one in a different context if the mode is not set correctly while testing, or if the training data of a single context were not provided.

The results out of testing this algorithm showed a big success and state of the art results when the context has been set and the training data has been provided while testing. On the other hand the performance dropped by a big margin when the mode(context) has not been set, or when the training data of the set context has not been provided but another context training data.



Figure 2 capturing the lips movement

### Multilingual speech recognition with a single end-to-end model

This model is pretty interesting, as it provides the possibility to combine multiple languages under the same model, so that at the end the algorithm may decide to which language does this sentence belong.

The algorithm idea is not new, rather it is an improvement over an already existed model (Listen-Attend-Spell attention based sequence-to-sequence ASR by William Chan) so that it can be adapted to help the purpose of differentiating the languages.

The idea of this algorithm is to include a set of languages, let us say L1 … Ln in a set L, and also include their corresponding characters in a set for each, in this case we have C1 … Cn, at the end we have also the language specific training data set which consists of (X1,Y1) for the first language till (Xn,Yn) for the language Ln.

The trick here is that we should be able to retrieve all the training data sets in one big training set, and all characters Set (C) as well.

Afterwards the big training data set is used to train the model, with no indication given which language does this training data set belongs to. But where is the trick ?

The trick is that this test was done over 9 Indian languages, which have a rare percentage of intersection of words and characters, so a small number of words may be used in more than one language, in this case we made sure that word X for instance will have most of the time only one language where it belongs to as the number of intersection is very limited, so the total number of words recognized will help us to identify the language used as we still have the set of all training data separated according to the language it represents.

So at the end when the model finishes recognizing the words, a probability shall be calculated to which language does a sentence belong to, based on how many words came from each language separately, thus the language with the highest probability will most likely be the one which the sentence belongs to.

## Methodologies

The voice recognition is considered to be another input alternative rather than the textual classical way of entering input to the machine via keyboard. In this sub-chapter the various types of voice detection and recognition shall be introduced.

All different types of voice recognition requires two modes in order for it to work appropriately, which are:

1. Training mode

In training mode a huge amount of samples must be collected – the more the better – in order to train the system no matter if it was Speaker dependent or independent.

The samples must be words or even sentences captured by a microphone which is the input device in this case.

1. Testing mode

Acoustic/audible characteristics must be analyzed out of the input sample, and then the important features shall be extracted out of it.

The feature vectors are then used to generate an input pattern which will be saved in a form of a matrix, then the unknown pattern when entered must be compared with all the input values in the matrix, and the best match found should be considered to be the correct interpretation and perform the action that it leads to.

### Feature extraction

So after this small introduction, let us explore now the different type of feature extraction techniques wish will be used afterwards to train the system.

#### Linear Predictive Coding (LPC)

This tool is considered to be the most trendy and powerful among all. In LPC we examine the input speech in a frame-based manner to produce vectors.

The LPC requires a pre-emphasize, which surpass the input speech, then the output of the pre-emphasizer must be blocked into frames, afterwards shall each frame be windowed, so that we can reduce the amount of signal disruption at the starting and the end of each frame.

Finally each windowed frame is auto correlated, and the maximum value of the correlation is considered to be the order of the LPC analysis and used afterwards to return the LPC coefficient which is the result we seek.

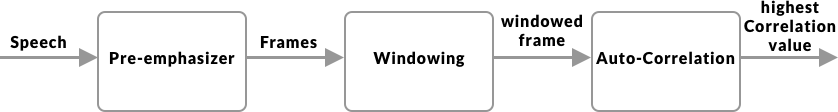


Figure 3 LPC

#### Perceptual Linear Prediction(PLP)

It is similar to the LPC, but the captured speech must be adapted to the psychophysics of human hearing. Its spectral features differs from the normal LPC in way that matches the human auditory system, so therefor it rejects a lot inappropriate information (filtering) which might be caused by the speaker, but non-hearable by human frequency captured.

#### Relative Spectral Filtering(RASTA)

This technique serves the same purpose as the PLP, but it helps to drop out/filter the voices (frequencies) caused by the background noises that are not related to the speech itself, so if this technique was combined with the PLP, then the results should be brilliant, then the Rasta passes each feature coefficient only.

It also includes a linear filtering of trajectory of power spectrum in the case of the noisy speech.

#### Mel Frequency Cepstral Coefficient(MFCC)

The word Mel would face any researcher about the topic of speech recognition, and that is for a good reason.

MFCC has huge achievements in the field of speech recognition, because it doesn’t offer only the possibility of recognizing the speech, but also the speaker.

The MFCC processing method is based on short-term analysis, which means that for each frame an MFCC vector must be computed, thus it is considered to be the best among all when it comes to estimate the human system response.

In order to obtain the coefficient (the result we seek in all types of speech recognition techniques) we apply a hamming window to the input (speech sample) in order to reduce the disruption of a signal.

At the end a Discrete Fourier Transform (DFT) must be used to produce the Mel Filter Bank.

To calculate the MFCC the following formula can be used.

Mel (f) = 2595\*log10 (1+f/700).

### Voice Classifier/Identifier

So after that features have been extracted from the signals, those features should be used to train the system to classify the words spoken. Here is a list of most commonly used classifiers.

#### Hidden Markov Model(HMM)

This model is trendy and an easy approach when it comes to classify words. It is based on huge vocabulary speech recognition systems, characterized by a finite state Markov model and a set of output distributions, and automatically trained on large speech data for many hours, thus comes its major advantage which decreasing the time and the complexity required for training the huge vocabulary in the system. Unfortunately this advantage led also its major limitation, which is the complexity to find the error of its scheme in order to enhance the performance.

#### Neural Network(NN)

Neural network are used mainly to solve complex identifications tasks, and they have also many number of advantages over the others, such as its ability to function independently of the speaker (unknown speaker) and its ability to work with noisy data. When compared to the HMM, the neural network provides much more better accuracy, especially when the amount of training data is large as HMM is used mainly when the number of training data is limited.

The neural networks is being used also in phoneme recognition, and therefore we have also a combination called NN-HHM, where HHM is used for the language modeling, and the neural network for the phoneme identification part.

#### Dynamic Time Warping(DTW)

Dynamic Time Warping uses dynamic programming to perform the optimization process of identifying the similarities between two samples, the first is the original, and the second is the manipulated version of it. That is why it has been used to identify the manipulated versions of voice, video or even images. But it is not used as often compared to the other techniques due to its continuity issues.

#### Vector Quantization(VQ)

The VQ is considered to be among the best when it comes to save time, space and the computation effort. VQ is basically a function mapping process, where it maps a word or input from a large space, into a smaller space called cluster, this cluster is identified by a code word, those collection of code words construct a code book.

Using the VQ method, a new codebook is constructed for each speaker, thus it may recognize the speaker even. This constructed codebook acts as a pre-recorded words for the user, then used when the speaker is being tested to be identified or to recognize what does the speaker say in the system.

When it comes to Voice Recognition, the VQ is used to retain the high speaker recognition rate as a parameter to identify elements like number of speakers for instance as well as the size of training database.

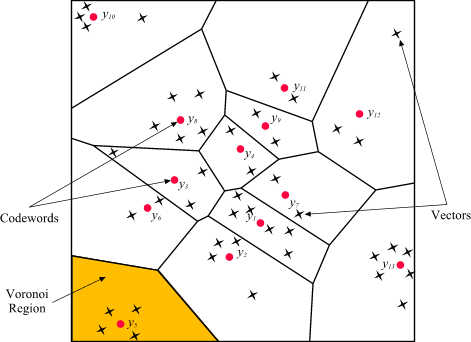


Figure 4 Codewords in 2-dimensional space. Input vectors are marked with an x, codewords are marked with red circles, and the Voronoi regions are separated with boundary lines (Source: http://www.mqasem.net/vectorquantization/vq.html)

## Topic in depth

This sub-chapter will go through the fields where the voice recognition is mostly used and in which context does it help to accomplish a specific purpose. At the first glance, one might think that voice recognition is used only for the smart assistant or voice messages, but it really did improve many other fields, that no one might consider the impact of voice recognition to those, and here are some.

### Evolving search engines

The search engines nowadays have become a necessity in our lives, but we find sometimes difficulties expressing what do really mean in verbalized words, and would prefer to say our query more naturally, here exactly came the benefit and the impact of voice recognition.

It is not meant that we search our queries via voice, although that is also an option on the major search engine such as Google for instance, yet it is meant that the search engine use the voice recognition training data entered daily and the intent behind it to give back better search results with high percentages of accuracy.

### Communication in service providers

With the higher number of services available nowadays, we can only imagine how many customers does each of those have, therefor the high number of technical support calls and guidance request, unfortunately that might lead to a long waiting periods on call till the customer been redirected to the technical assistant.

That has arisen the need of an innovative solution to help the customers get the guidance they needed in a short amount of time, thus is the voice recognition integrated.

In this situation the customer doesn’t have to enter any numbers using the keypad, yet the customer should be able to speak more naturally and let the smart assistant guide him/her through the problem solving process.

### Voice biometrics as authentication

This model has been used even in Switzerland by some companies, Sunrise telecommunication for instance, as they would ask the customer to use a sample of the voice to get its biometrics and construct a model so that the customer doesn’t have to give a number of personal sensitive data which might take a while before illustrating the customer’s request and intent of calling.

The voice biometrics are not only used in the services company, but also to authenticate the unlocking process of the smartphone and also to give commands to the virtual smart assistant.

### Smart Assistant

Thanks to Amazon, Google and Siri, the smart voice assistant industry is growing rapidly nowadays and companies compete to provide a better experiences to their clients to acquire a bigger market share.

Voice assistant helps to accomplish mostly all types of task that we can think of, for instance setting up an alarm, playing music, calling contact, writing messages …etc.

Although a huge amount of people still consider the breakage of their own privacy, nevertheless the smart assistant is the considered to be the next big thing, especially when the companies start to expand the capabilities of the voice assistant as Google did for instant with the Dublex feature presented last year.

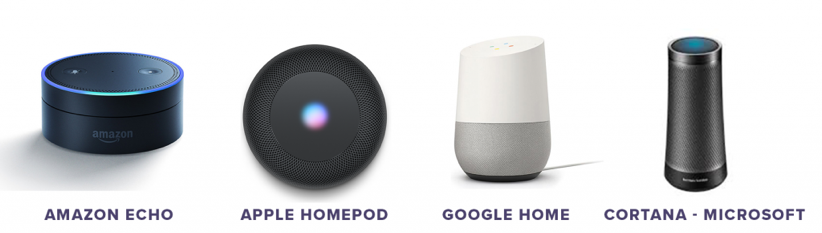


Figure 5 avaliable voice assistants on the merket (source: https://idfive.com/ideas/preparing-website-voice-recognition-usability/)

### Forensic Department

Although it may seem weird, but voice recognition has helped a lot so far in identifying criminals using their voice, in this case if a sample of the criminals voice was captured as the crime was committed, this sample can use later on to identify the criminal among a list of suspects by comparing their voices to the captured sample.

## Technical research

This section will go into depth about which existing implementations for natural language processing are available and how they could be integrated in the SEEA solution

### Google assistant

Google assistant is a platform independent natural language processor. It supports both Android and iOS devices.

It is recommended to use api.ai for the natural language processing in the background. Api.ai tries to extract intents and entities from the input.

#### Entity

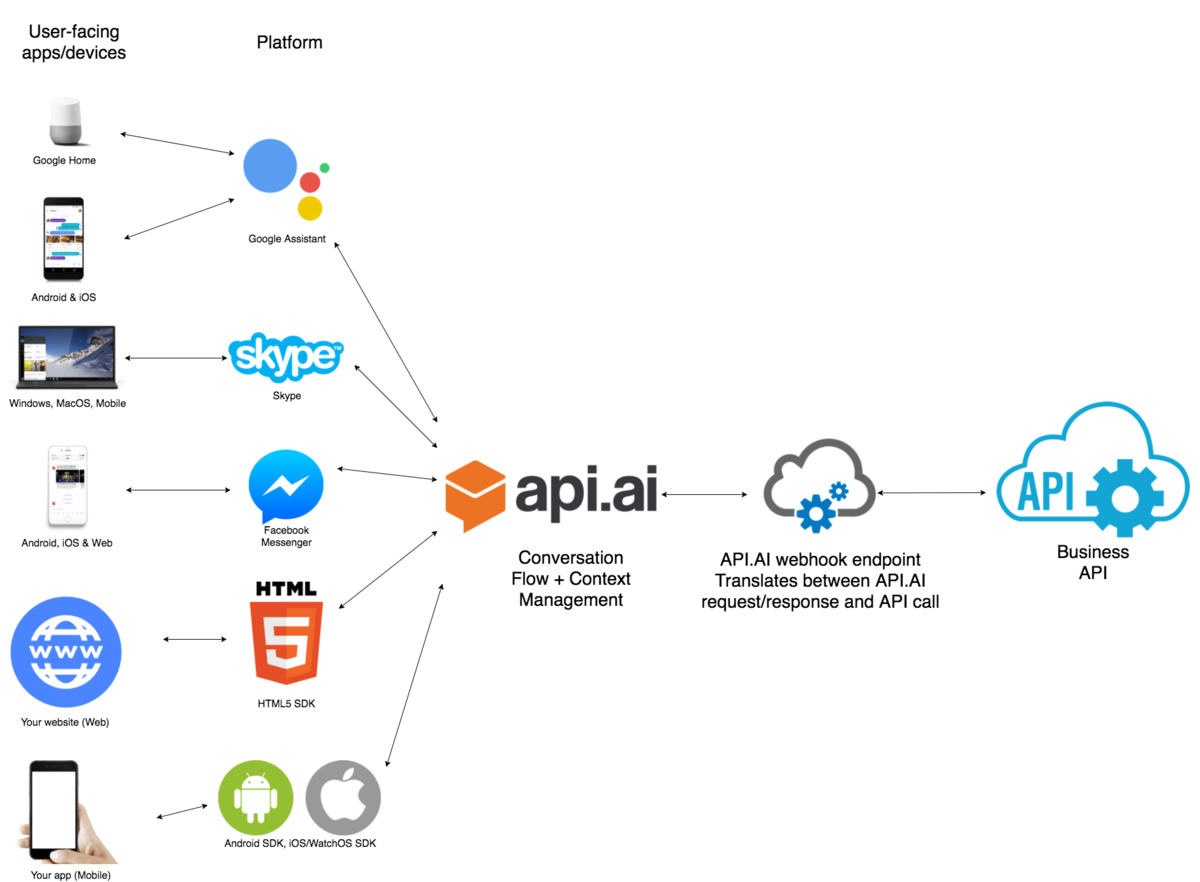
Entities could be for example Animal.

The Entity Animal contains multiple different animals, e.g. dog, cat, bird, etc.

After creating a list of things that are part of the “Animal” Entity you can also help the A.I. by providing synonyms. E.g. A “Puppy” is also a Dog, “Dogs” can also be interpreted as Dog, etc.

#### Intent

Intents are actions that can be derived from phrases the user tells the program. E.g. “Tell Me A Joke” would be an intent, the program will be prompted to execute the task (intent). Like for entities, the developer needs to provide the algorithm with example sentences that should trigger the intent. The more examples are given, the more non predefined sentences the A.I. can use to trigger the intent.



#### Pricing:

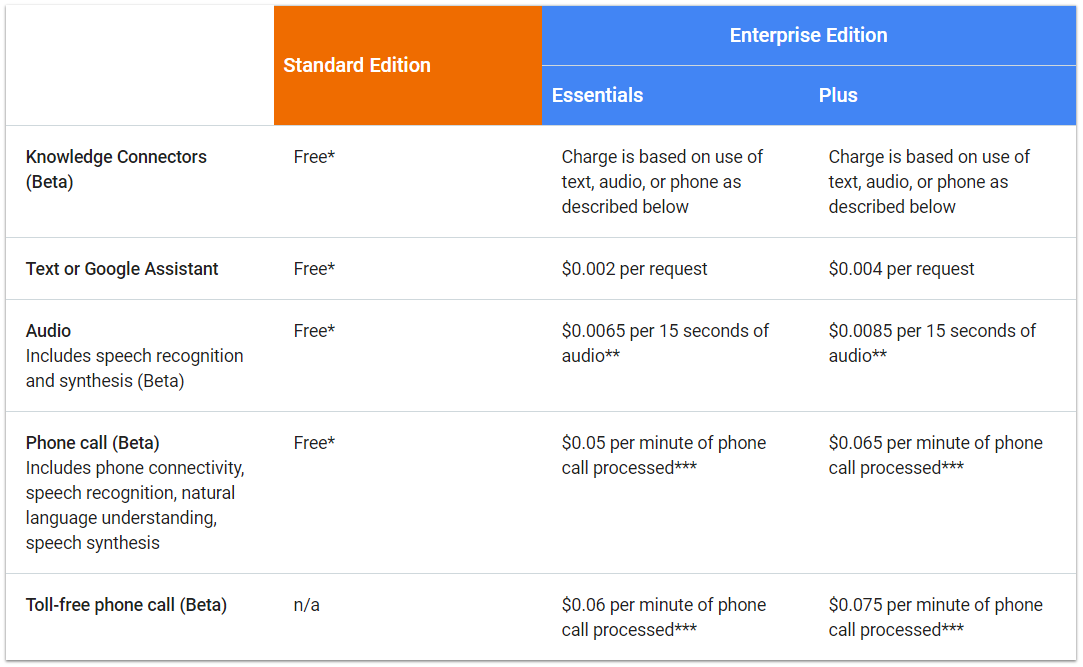
**Dialogflow Standard Edition** provides all of the core features of Dialogflow, but interactions are limited by usage quotas, and support is provided by the community and e-mail. It is ideal for small to medium businesses that want to build conversational interfaces or those who want to experiment with Dialogflow.

**Dialogflow Enterprise Edition** provides higher usage quotas and support from Google Cloud support. Dialogflow Enterprise Edition is a premium offering, available as a pay-as-you-go service. It is ideal for businesses that need an enterprise-grade service that can easily scale to support changes in user demand.

Available in two pricing plans:

**Essentials**: This plan contains all features offered by Dialogflow Standard Edition, plus enterprise-ready quotas for speech recognition, speech synthesis, and telephony gateway.

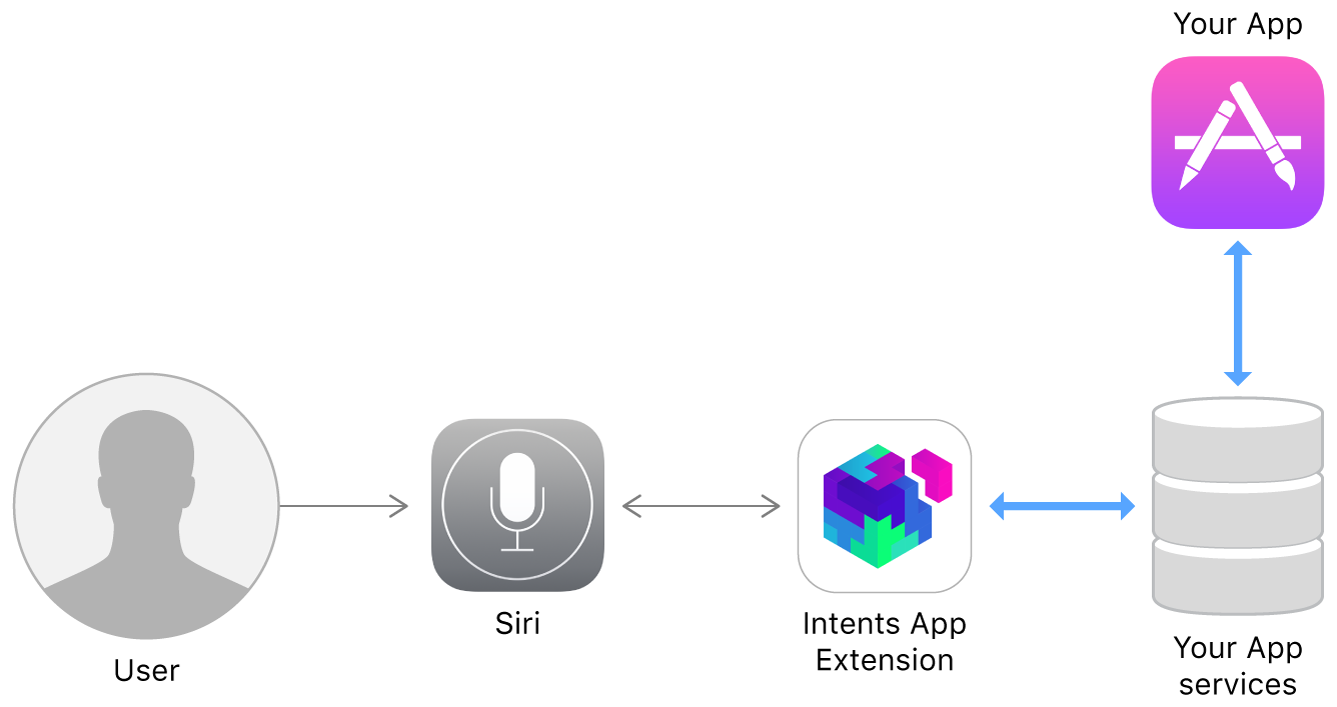
**Plus**: This plan contains all features offered by Essentials, plus enterprise-ready quotas for knowledge connectors. Each request from an Enterprise Plus agent performs the regular intent recognition and entity extraction, as well as a knowledge connector search.



### Siri Assistant

The development environment of Siri is called SiriKit and it offers the possibility to translate calls to the assistant into calls for the app you want to use. For this to work an internet connection is mandatory as the voice data is sent to apple servers for the natural language processing.

Apple has a strict List of so-called Domains and Intents which the application call has to be assigned to for it to work.



Domains:

* Lists: Simple list creation and editing
* Visual Codes: Display QR codes
* Ride Booking: Requesting a ride, this is intended for services like Uber
* Messaging: Simple text messaging
* Photo Search: Making a search request for photos and videos and display them.
* Payments: Creating a payment request or sending money
* VoIP Calling: Make a call (also supports video)
* Workouts: Starting, pausing and finishing a routine, explicitly specified as workout
* Climate and radio:

Pros:

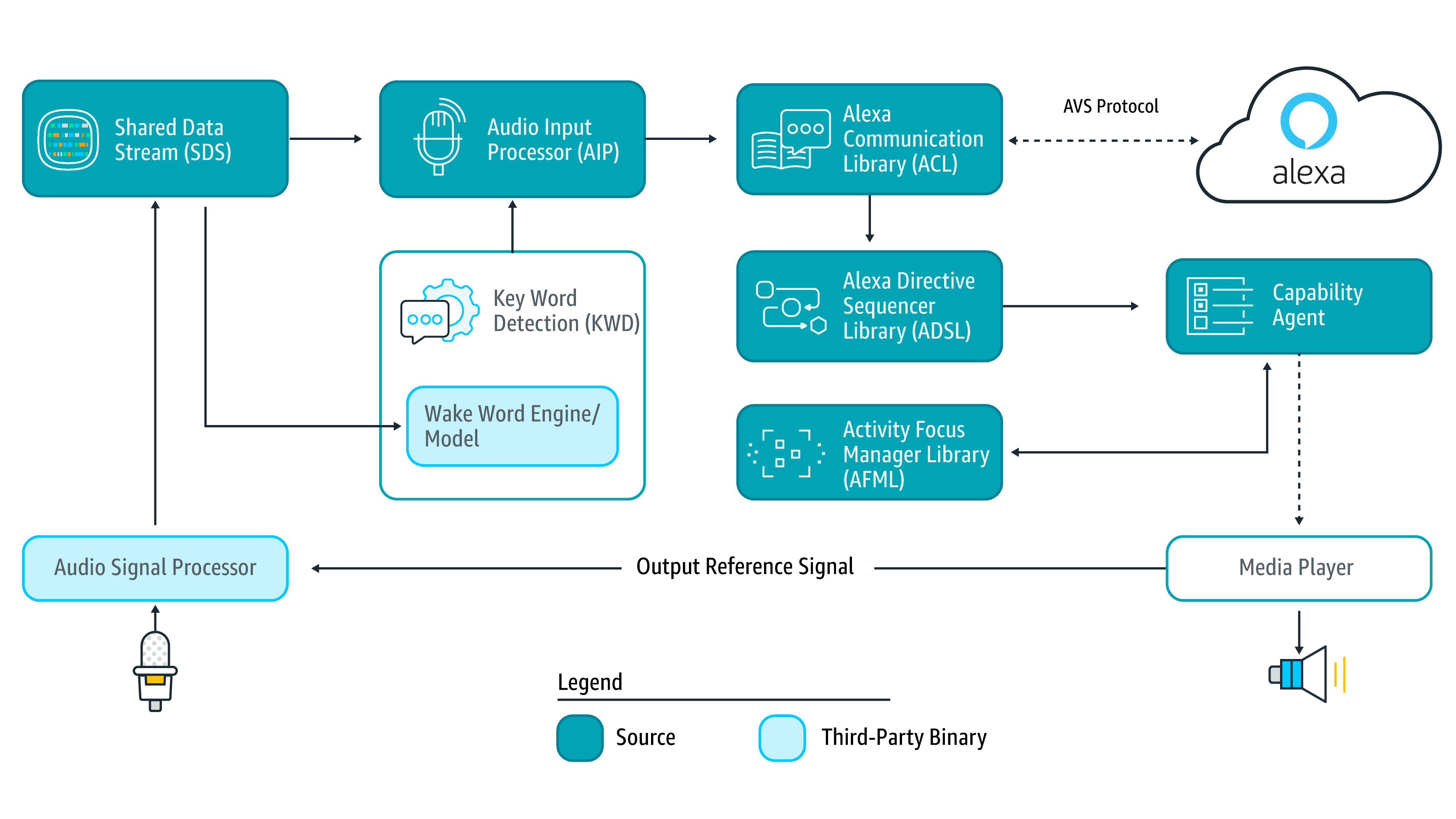
+ Apple would take over the NLP part

Cons:

* Only works with an internet connection
* Domains limit the usability for the app
* Always requires specific keywords for it to work

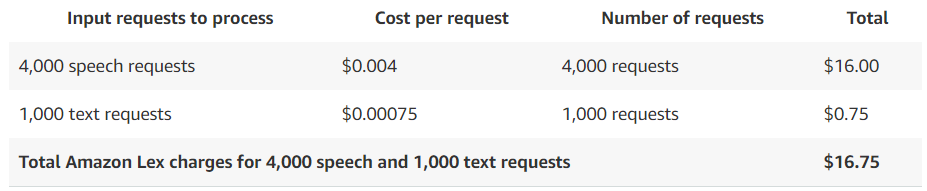
### Alexa Voice Service (AVS)

AVS is a platform independent service that is used over an HTTP/2 api.

AVS is built like Siri in the way that it is task oriented. The tasks of AVS are called directives and events, the following are supported:  


* Alerts: timer and stopwatch features
* AudioPlayer: control music playback, etc.
* Bluetooth: manage Bluetooth connections
* DoNotDisturb: enable DoNotDisturb mode on device
* EqualizerController: control equalizer settings and equalizer modes
* InteractionModel: Allow the client to support complex interactions and Alexa routines
* Notifications: API for notifications
* PlaybackController: navigate playback queue via GUI or buttons
* Speaker: volume control, mute and unmute
* SpeechRecognizer: API for speech capture
* SpeechSynthesizer: Text to Speech API
* System: System state
* TemplateRuntime: visualize metadata of requests

Pricing:



# Concept

After doing the research part, the decision were made to go for Google Assistant, for a number of reasons.

1. Google Assistant is available on both Android (built-in) and iOS (with an installed app), and that fulfils the clients requirement of having an assistant that works on multiple Operating Systems platforms.
2. Google Assistant is partially – yet not officially – supported in Switzerland, unlike Amazon Alexa for instance, which is totally not supported. This caused a lot of problems when the team tried to get support from the Google customer support service, as the service is not supported officially, thus not the full capabilities of the Assistant might be used, yet Google Assistant was the best regarding the functionalities offered among the others.
3. Google Assistant support multi-languages, which the others do as well, yet Google Assistant algorithm kept improving within the last couple of years, and it shows brilliant results when it came to Voice Recognition, unlike the other Voice Recognition services. That multi-lingual support fulfils the clients requirement of having an app that can operate on at least two languages (German and English) for start.
4. Google Assistant tutorials are available online, and the initiation of a new action is done using powerful APIs, namely Google Actions and Dialogflow Consoles, which their tutorials are available online as well.

Beside the clients request, that the voice assistant must be included in the app, it was totally up to the team to determine what features should be included as well, and come up with ideas for the UI.

In this chapter the concept idea will be discussed in details.

## Identifying the Persona

Before starting listing all the ideas and features that were brainstormed, a simple question must be answered due to its great impact on what features to include, “What group of end-users is this application targeting?”.

In order to answer this question, the client were asked to describe a typical member of Swiss Engineering in order to start establishing a persona that matches those characteristics.

The app is targeting the Swiss Engineering members, who are mainly engineers starting mid 30’s, looking for events which might give them a quick brief into what is new and available in the market, so that the members can choose then to go deep into the presented topic in the events or pass it and look for another topic that may be interesting. Swiss Engineering offers also another type of events where it is not about new topics, rather a meeting where members can get to know each other and have small quick discussions with each other.

After this input a Persona has been constructed, which you will find in the appendix.

But as the app will be published on the Google play and App stores, there is also a chance that an end-user, who might not be a member, will download the app and start exploring it. Therefor another Persona were constructed as well to address this type of end user, although only two features will be enabled for them.

## Requirements gathering

This app shall be used to book events and is targeted to engineers who are members of Swiss Engineering organization, and as the client didn’t hand in any other requirements other than the voice assistant, it was the task of the team to gather the requirement in order to design the first prototype.

As the team didn’t have access to the members of the Swiss Engineering, a sample of potential members were asked about what would tasks do they expect to accomplish within an event app and what would they like to be able to do with such an app that it is not yet available, that sample contained students who are about to graduate, architectures, electrical engineers, Elektromonteur(electric technician), IT technicians, designers and finally biochemical engineer.

The result of the requirements gathering process were gathered afterwards and clustered, each set of requirements that are greatly correlated belongs to one cluster, and that cluster was called then a feature, except of some of the end-user input, which was a feature on its own, therefor a set of requirements were constructed to fulfil the purpose of this feature, Search Event feature for instance.

## Features formulization

As mentioned in the previous sub-chapter, each set of requirements constructed a features. Thus the list of features is as follows.

1. Talk To SEEA

At this point neither the end-user sample nor the developing team was convinced that having such a feature would improve the app functionality, or make it intuitive to use, yet this feature was kept because it was the clients request to have it.

1. Search for event

The sample of the potential end-user requested this feature in the requirement gathering phase, and the team knew from the start that such a function should be implemented by all means, as this is the core feature of any event app, and it is not an exaggeration if said, all apps.

1. Book Events

The booking event function should be enabled within the app, and that is an important functionality, as it is not really helpful to offer the end-user a search function then blocking the end-user from booking an event.

1. View Booked Events

This was considered at the beginning to be a feature on its own, its purpose is to view the list of booked events by the member of Swiss Engineering.

But the team knew that those list of features represent the core essentials features to enable, but lacks the element of end-user retention. Thus a brain-storming session was help to identify a list of features that might attract the member to use the app and having a good user experience, so that the app shall be used over and over to book events rather than using the web-site to do this task on a smartphone.

### Brain storming session

The brainstorming session was really helpful to identify the list of the features that the end-user didn’t think about while the interviews took place, yet the end-user would really like to have and be intuitive to use at the same time.

Facebook Marketplace, YouTube and even Ricardo for instance do all have a section where content is being suggested to the user, that might be found interested. So we thought that such a feature is important to have, as it would help give the user an exceptional personalized experience, and also help the client to get more events booked, that the end-user didn’t think before to search for.

At the beginning all ideas were allowed, no matter how complicated they were to implement and their feasibility to have. So therefor we ended up having about 20 ideas on the board, then filtered into what is feasible to have, and what would help to make the app usage intuitive, and also offer an important extra value to the end-user and the client at the same time. To test the intuition of the features a sample of the end-users were asked to give their opinions, so we ended up afterwards we a list of ideas that are feasible to implement, that were then presented to the coach and the client as well to get their opinion on what would they rather have.

A list of the full ideas that were elaborated during the brain storming session can be found in the appendix chapter.

### Feasible/Filtered ideas

As mentioned in the previous sub-chapter, the ideas gathered during the brainstorming session were then filtered, and formulated into conceptional design.

We have found out during the ideas conceptional phase, that many do have similar prerequisites in order for them to function properly. Thus a new parent feature were elaborated to classify the ideas filtered into groups, that parent feature is the Tagging System.

#### Tagging system

The Tagging System idea is simple, efficient and helps to deliver business value as well as a functional value. For each event added into the Swiss Engineering Database of events, tags should be inserted (or automatically extracted by an algorithm from the event name or description) into each events, so that the search function is not only dependent on the exact written event name or description, but also the tags that were marked under each event.

The importance of the Tagging system is that it offers the ability to classify the events into more useful and meaningful clusters, and that will help to retrieve even better results when it comes to search for events, as well as find a sufficient way to suggest events to book for the end-user. This system business value is that it offers also the possibility to know which types of events are the end-user mostly interested in, so that the client might have an insight and can estimate how many members will book potentially an event.

But in order to gather those tags, an intuitive, yet “cool” techniques should be implemented so that the end-user will have fun giving the tags(topics) that are found interesting.

An example of the Tagging system ideas are the following:

1. **EventTinder**

The Idea of EventTinder is to swipe right for the tags that the end-user is interested in, or to swipe left in case the end-user has chosen to not hear anything about this tags(topic) anymore.



Figure 6 EventTinder

1. **TagsSquare**

The idea of TagsSquare is to drag and drop the visible card into one of the four corners, where each corner represent the end-users level of interest about the name of tag written on the card.



Figure 7 TagsSquare

But beside the Events Suggestion feature, we wanted to include also some ideas that fulfils the reason of existence of the Voice Assistant in the app. Those ideas were also gathered in the brain storming session, and represented in the following subchapter.

#### Voice Assistant ideas

Voice Assistant is an important feature of the app, because it was the first feature requested by the client to implement and have, but with specification on how should it work or how is the end-user expected to interact with.

So after the brain storming session the following ideas were suggested to the client:

1. **Search Event by voice**

That is the obvious feature to include in an app that contains events, and must have a voice assistant enabled feature. But it was removed afterwards as the intuition of this feature was questioned, and after the feedback of the end-user which stated that they will most probably not use it, and rather search for events using the classical textual input approach.

1. **Add to my Calendar**

The idea is to simply ask the enabled voice assistant to save the event shown on the screen to the calendar app, or an integrated calendar feature inside the app itself. Because the feasibility of implementing an integrated calendar into the app was questioned, therefor we settled on adding the event to the default calendar app installed on the smartphone.

1. **Ask simple questions**

The idea is to enable the user to ask the built in assistant simple questions such as “when is my next event?”.



Figure 8 ask simple questions

1. **Follow up questions**

It is related to the previous idea of asking the assistant simple question, but this feature require that an app should be visible on the screen, and that the assistant is aware of that previewed event. The follow up questions might be something like “how far away is this event?” and the assistant should answer with the distance per Km rather than hours, as the time to reach the location my differ depending on the traffic jam and the way of commute, whether it was a private car, walk or using the public transportation. Another important aspect that was questioned, is that how would the end-user realize that such feature existed and is enabled?.

A list of the full ideas description of both Tagging System and Voice Assistant can be found in the appendix chapter.

But after listing all of those ideas side by side, a really cool intuitive feature emerged from both aspects, Tagging System and the Voice Assistant, which is Talk To SEEA.

#### Talk To SEEA(Tags System & Voice Assistant)

Talk To SEEA enabled us to combine both of worlds, Tagging System and the Voice Assistant into one intuitive feature. Talk To SEEA enabled the end-user to have a conversation with the built-in Assistant, so that the end-user may enter the Tags that the end-user find interesting, so that those Tags are used then to re-direct events to the Get Inspired page.

Talk To SEEA is able to recognize the name of the topic and extract it from the text, so that the end-user list of liked tags and the event been suggested match the end-users own preference and interest.

Talk to SEEA enables the end-user also to ask simple questions such as “when is my next event?” so that the end-user get the requested answer without navigating through the app screens.

If the end-user entered Tags were found, SEEA will show also a small action button in the suggestions bar that opens the Get Inspired page.

Talk To SEEA has two modes, the first is the text mode, in which the end-user will choose to enter the input textually, and SEEA shall respond also in text. The end-user might choose also to give the input via voice, and in that case SEEA will also respond via voice, but both end-users and SEEAs input will still be shown as text, so that the end-user might still check if SEEA got what was said correctly, and that the end user may refer to the textual output in case the end-user didn’t get what SEEA said correctly.



Figure 9 Talk To SEEA

#### Social Factor/Aspect

Because Swiss Engineering is in core an organization that gathers a lot of engineers. The idea came up as the team were searching for an idea, that delivers the “wow Factor”.

According to resent Harvard Business Review, engineers had the second loneliest profession, where lawyers take the lead.

The findings may not come as a big surprise. Engineers spend a lot of time working on their designs in front of their computers. Technology allows them to collaborate and coordinate with colleagues instead of meeting with them face-to-face. Their work environment doesn’t always allow them to form friendships with colleagues.

So the idea was formalized to take a step, and act against the loneliness. The idea of the Social Factor is to recommend events to an end-user, Lorenzo for instance, based on the new bookings of members who attended the same event with him previously, so if Lorenzo attended an event with Wing earlier, the new booking of Wing should be suggested to Lorenzo as well but anonymously, so that Lorenzo doesn’t know based on whom this event is being suggested, rather just labeled and listed under the Social Factor findings.

This idea will help the Swiss Engineering members to see familiar faces in each events, so that the member can then approach those people and get to know them and make friendships inside the Swiss Engineering Society, under the slogan “From Virtual to Reality”.

Of course the members are not forced to get to know those familiar people, but it is rather just a push from the Swiss Engineering towards this step, while the final decision is totally up to the end-user.

A screenshot of a cell phone

Description automatically generated

Figure 10 Social Aspect/Factor

### SEEA final features list

After the brainstorming phase and filtering the feasible ideas, the candidate features list to include has been set and prioritized, and after discussing this lists with the coach and the clients, the final list of features to include has been set as follows.

#### Talk To SEEA

Where the end-user will be able to interact with the built in voice Assistant to enter the preferred Tags, so that the suggestion the Get Suggestion Page improves.

#### Get Inspired

Where a list of suggested events based on the tags entered to SEEA been shown. And also the events that has been suggested based on the Social Factor that has been discussed previously.

#### Search events

Search event is the core feature, no event app can be accomplished and no good level of user experience can be offered without including this functionality.

#### Profile

The profile page will include all the personal data of the user, beside a tab to navigate to the booked events and a Dark theme toggle button to enable the Dark mode, to address the sample of end-users who liked this theme.

#### Saves Locker

Saves Locker is the feature that will enable the end-user the possibility to save events and searches, so that it is all kept under one place, but elegantly separated to enable the ease of access to the required data fast.

This list contains the features that were selected to be included in the first release of the Swiss Engineering Event App, where in the future, many extra features might be added from the list of the filtered features, such as EvenTinder.

All of those features should be enabled to the member users, but only the Saves Locker and Search events (with no booking) shall be offered to the non-member end-user, because it is required from the Swiss Engineering that only members can book events, and in order to offer suggested events, an attribute shall be reserved in the end-user record to save the liked tags and previously visited events on the Swiss Engineering database, which the non-member end-user basically doesn’t have.

# Prototype

# Implementation

<TODO>

# Analysis & future thoughts

what's out of scope, what could be done in future

# attachments

**Vision**

|  |  |  |  |
| --- | --- | --- | --- |
| VISION  *What is your purpose for creating the product?*  *Which positive change should it bring about?*  Book your event at a glance!   * The “Swiss Engineering Event App” is here to book events on user’s smartphone instead of using the website, as we realized that the website’s mobile usage is not as smooth and intuitive as it is intended to be. * We want to build an app with a simple modern design that is not overloaded with too many features that the user does not use on a regular basis or even at all. * “Make It Simple To Use” is the core objective that we aim for while building this app, using the UX guidelines while also keeping in mind that there will be no tutorial are the fundamentals and the base that “Swiss Engineering Event App” will be built on. * The end-user does not have time to spare; that is why the “Swiss Engineering Event App” will require less steps than the website to book events and edit event’s reservations. * The core functions of the “Swiss Engineering Event App” should be visible at the top and easier to access than the extra features, which the user can still access using a logical sequence based on the user’s previous experiences of dealing with smartphones apps. * The “Swiss Engineering Event App” will have the Voice Assistant feature included in a way that facilitate exploring the events landscape, details as well as booking events. | | | |
| Target Group  *Which market or market segment does the product address? who are the target customers and users?* | **Needs**  *What problem does the product solve?*  *which benefit does it provide?* | **Product** *What product is it?*  *what makes it stand out?*  *is it feasible to develop the product?* | **Business goals** *How is the product going to benefit the company? what are the business goals?* |
| * The Market Segment that the app targets, are the “Swiss Engineering” members. * The Target users are the members (mostly engineers) who want to book their events. * The registered students are also considered to be an important targeted users as they are future engineers and potential members who will be paying the membership fees in the near future; so the app is designed to retain them. | * Using the website on a relatively smaller screen size raise the risk of unintentionally messing up the layout, especially with a huge number of tabs and navigation menus, such as the number contained in the website; thus the app is needed. * Less elements per screen on the app, compared to a laptop, will keep the user fully focused and never distracted. * The app will always be easier and more intuitive to use than a website, especially on a smartphone. | * The “Swiss Engineering Event App” is a smartphone cross-platform app that runs on both iOS and Android natively. * The “Swiss Engineering Event App” is the only app developed to work with the “Swiss Engineering” platform. As there are no competitors, we believe that the success of the app will depend   solely on the member’s willingness to make the switch from using their laptops to reserve their events to using their smartphones instead. We are confident that they will never look back to use their laptops for reservations ever again, due to the easiness of use and the better experience that the app provides. | * Because of the impressive app, new members will join the association which would potentially increase the amount of reservations, and therefore the revenue from the hosted events. * The reputation of the association will improve as the number of event visitors increases. |

**Persona**

# 

Figure 11 Member user Persona

# 

Figure 12 non member user persona

# 

Figure 13 Onion model

**Brainstorming boards:**

A close up of text on a whiteboard

Description automatically generated

A close up of text on a whiteboard

Description automatically generated

A close up of text on a whiteboard

Description automatically generated

**Filtered Features**

**When is my next event**

This feature helps the end-user to get information about the next booked event. If the end-user has booked **“Does culture matter on international business ?”** event on 4th April, The end-user would say phrases like “Hey Seea, when is my next event ?” and the voice assistant should respond “your next event **“Does culture matter on international business ?”** is tomorrow at 5 PM”.

This requires that the voice assistant in use is aware of the end-user booked events.

A screen shot of a social media post

Description automatically generated

Figure 1 Original Photo credits to google

**Suggest events based on users of the same taste**

When a user books an event, the app would start creating a matching process in the background to suggest new apps to attend, based on what did the co-attendees who visited the same event booked later.

The matching process would have to completely anonymous so that the privacy of the end-user doesn’t get violated.

The developer team really liked this idea because it helps to create a social relationships between the end-users who have the same taste and interests. This idea’s human value is really high to be just dropped, because it would help to create a real social community around the Swiss Engineering platform.

A screenshot of a cell phone

Description automatically generated

**Let Seea get to know you**

Seea is the name of the voice assistant inside the app, the name is the initials of the project “**S**wiss **E**ngineering **E**vent **A**pp ”.

Seea would start asking the end-user several questions to get to know him/her better, and start filling out the gaps to construct a model and start looking what events should it suggest to the end-user.

Those suggestion can be shown whether in the same page after answering all the question, or later in the suggestions screen.

This idea requires that Seea is aware of the events Database and can convert the textual recognized speech into a query to retrieve the results.



**Use Voice Assistant for follow-up questions**

We as a developing team understands the different between the human and machine, for instance the human would remember the name of the streets that was raised in, visited as a teenager or had business in as an adult.

One of the first questions that the end-user would after showing an interest in an event, is where is it? even if the end-user saw the name of the street, the user would most probably search for that location to know how far away is it, especially if that event is taking place in another canton.

So this feature would save the time opening the maps and entering the name of the street and choosing the end-user location, as the end-user would ask simple direct questions, such as “How far away is it?” and Seea would reply “10 km” for instance so that the end-user can decide whether or not to book this event.

This feature requires that Seea is aware of the users current location – or the address listed as the address in the user’s profile – and the location of the event.

**Event search**

The obvious go to feature when voice assistant is to implement in an event app. The end-user would trigger the voice assistant “Seea” and ask her to search for events that match the entered vis speech criteria given.

Seea would then respond with showing the events that match those criteria as a list in the events search results screen.

Entering the criteria could be done in the following two different ways:

* **Input as one sentence**

The end-user would have to say full sentences, for instance “Show me the events about **economy** on **4th April** in **Zurich**”, then Seea will retrieve the result of searching for those three inputs: Topic, date and location.

* **Input as a conversation**

Seea would first ask the user about which type of search criteria input to enter via voice, for instance Seea would say “What topics are you interested in?” the end-user would say “Economy”, then Seea would ask about the date by saying “which date should I search for” then end-user would then say “4th April” and Seea would finally ask about the location by saying “in which canton should I search” and the end-user would say “Zurich”.

Of course the feasibility of implementing such a feature is still discussable, because of how intuitive is it to replace the normal text input which most of the potential end-users prefer with a speech input.

Seea has to be connected to the Database of the available events and their corresponding attributes needed to complete this search query.

**Add events to calendar**

This is one of the most intuitive to use features on this list. Normally when the end-user books an event, the next step is to copy this event information manually in the calendar, or wait for an event invitation via mail to import the event object to the calendar or if the user was lucky enough, the app would offer the end-user a button called “import to calendar” to import the event into the user’s private calendar service.

Seea has a potential usage at this point to import the booked event to the end-user’s calendar by just saying “import this event to my calendar” and that is it, the event is now saved into the user’s default calendar in use.

In this case, Seea has to be aware which calendar is the end-user using and also to know how to construct an event calendar object to import it into the calendar.

**TagSquare**

This is a gamic feature that might make the process of entering the end-user’s preferences more fun than usual. In the screen a square will be shown with a label in the middle containing a name of one of the tags entered as an event was created, the users task is to swipe the tag label into one of the four corners of the square, namely “**book me**, **interesting**, **meh** and **hell no**” each time the user swipes one tag into one of those four corners and release a new tag appear in the middle to repeat the operation again. The end goal is to teach the AI what topics are the end-user interested in to send the end-user better suggestions in the future.

All what is needed is an access to the list of tags used for events, and then a way to save the end-user’s choices and use them in the future to help provide even better suggestions.

A close up of text on a white background

Description automatically generated

**EvenTinder**

Dating is back on the table for all involved in a relationship ☺

This idea is basically about using the event’s tags to create a list of cards stacked on each other, all what the end-user needs to do is to swipe the card right or left, smash or pass.

The “smashed” cards tags will be used in the future to provide better recommendations in the future for the end-user.

The AI needs to be aware of the tags entered in the database to provide them as cards.

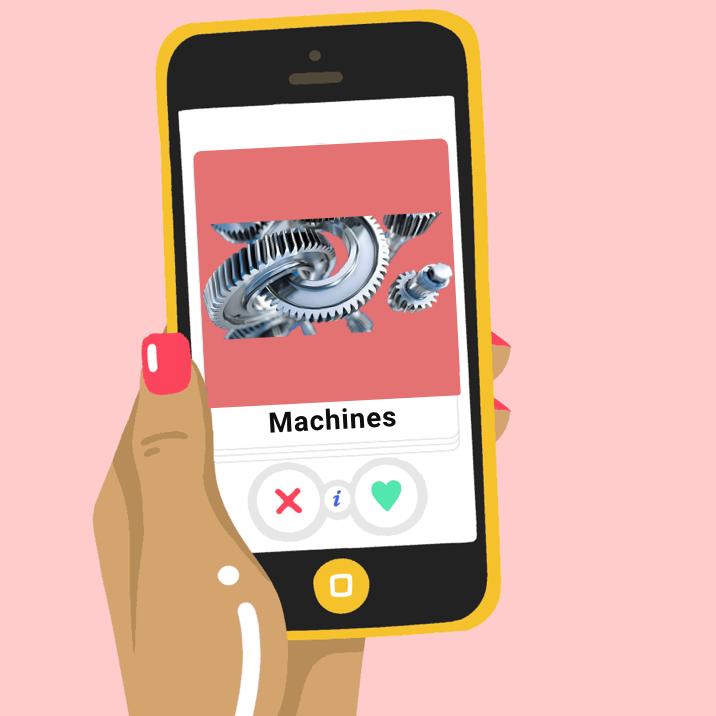


Figure 2 Original Photo https://www.bustle.com

**Seea Mascot**

This is another attempt of gamification. The Idea would be to create a visual personal assistant (e.g. Microsoft clippy) that is always present on the screen and would double as an integrated help function. For example, the mascot would notify the user if there is no internet connection present or would be clickable on pages that might require help. The mascot should be located on one of the edges or corners of the screen to be easily accessible without obscuring the view on the actual application and its functions.

The Mascot would need to be aware about the content of each screen so that it give helpful instructions, the team needs to design the Mascot as well.

**First Version of use cases:**

|  |  |
| --- | --- |
| ID | 1.1 |
| Name | Login |
| Actors | Member End User |
| Goal | Login into the platform |
| Pre-Conditions | The User has a membership |
| Steps | 1. The user opens the app. 2. The user taps on the profile icon top left. 3. The user enters the username and password. 4. The user taps on login button. |
| Alternative | --- |
| Post-Conditions | The user is now logged in. |

|  |  |
| --- | --- |
| ID | 1.2 |
| Name | Register |
| Actors | Non Member end-user |
| Goal | Create an account to be able to reserve events. |
| Pre-Conditions | The user doesn’t have an account. |
| Steps | 1. The user opens the app. 2. The user taps on the profile icon top left. 3. The user taps on register now label. |
| Alternative | --- |
| Post-Conditions | The user will be redirected to the Swiss Engineering website to complete the process of creating an account. |

|  |  |
| --- | --- |
| ID | 1.3 |
| Name | Reset Member’s Password |
| Actors | Member end-user |
| Goal | Reset the Member’s forgotten password to be able to login into the member’s account. |
| Pre-Conditions | The user has a valid account. |
| Steps | 1. The user opens the app. 2. The user taps on the profile icon top left. 3. The user taps on “Forgot My Password” label at the bottom. |
| Alternative | --- |
| Post-Conditions | The user will be redirected to the Swiss Engineering website to complete the process of resetting the account’s password. |

|  |  |
| --- | --- |
| ID | 2.1 |
| Name | Search Events. |
| Actors | End-User (Whoever has the app installed on a smartphone). |
| Goal | To explore a list of available events matching the search filters applied. |
| Pre-Conditions | The app is installed on the smartphone. |
| Steps | 1. The user opens the app. 2. The user taps on events card. 3. The user enters his preferred search criteria to apply as filters. 4. The user taps on search button. |
| Alternative | --- |
| Post-Conditions | A list of events that matches the entered search criteria will show up in a table. |

|  |  |
| --- | --- |
| ID | 2.2 |
| Name | Save search filters. |
| Actors | Member End User |
| Goal | Save search criteria to be able to use it in the future easily without having to re-enter them again. |
| Pre-Conditions | The Member has logged in successfully. |
| Steps | 1. The user opens the app. 2. The user navigates to the events card. 3. The user enters the search criteria to be applied while searching. 4. The user taps on the star icon. 5. The user taps on search button. |
| Alternative | --- |
| Post-Conditions | A list of events that matches the entered search criteria will show up in a table, and the entered criteria will be saved to re-use in the future easily. |

|  |  |
| --- | --- |
| ID | 2.3 |
| Name | Search using saved search criteria recipe. |
| Actors | Member End User |
| Goal | Search quickly using a saved search criteria. |
| Pre-Conditions | 1. The Member has logged in successfully. 2. The Member has a saved search criteria. |
| Steps | 1. The user opens the app. 2. The user taps on the favorite search card. 3. The user taps on one of the search criteria recipe shown. |
| Alternative | --- |
| Post-Conditions | A list of events that matches the entered search criteria will show up in a table. |

|  |  |
| --- | --- |
| ID | 2.4 |
| Name | Book event. |
| Actors | Member End User |
| Goal | Book a seat in an event. |
| Pre-Conditions | The Member has logged in successfully. |
| Steps | 1. The user opens the app. 2. The user navigates to the events card. 3. The user enters the search criteria to be applied while searching. 4. The user taps on the star icon. 5. The user taps on search button. 6. The user taps on one of the listed events. 7. The user taps on the reserve button. |
| Alternative | --- |
| Post-Conditions | The user will be redirected to the Swiss Engineering website to continue the reservation process. |

|  |  |
| --- | --- |
| ID | 2.5 |
| Name | Delete a search criteria recipe. |
| Actors | Member End User |
| Goal | Remove a search criteria saved recipe. |
| Pre-Conditions | 1. The Member has logged in successfully. 2. The Member has a saved search criteria. |
| Steps | 1. The user opens the app. 2. The user taps on the favorite search card. 3. The user swipes the card of the recipe to delete to the left. |
| Alternative | 3b. The user taps on the edit button top left.   1. The user taps on the minus (Negative) symbol beside the recipe to delete. |
| Post-Conditions | The recipe will not be listed anymore among the saved recipes. |

|  |  |
| --- | --- |
| ID | 2.6 |
| Name | Read Events Description. |
| Actors | End User |
| Goal | View the description of an event with all available information about location, date, topic … etc. |
| Pre-Conditions | The app must be installed on the smartphone. |
| Steps | 1. The user opens the app. 2. The user taps on events card. 3. The user enters his preferred search criteria to apply as filters. 4. The user taps on search button. 5. The user taps on one of the listed events. |
| Alternative |  |
| Post-Conditions | The description of the tapped event will be shown to the user with all included information. |

|  |  |
| --- | --- |
| ID | 3.1 |
| Name | Cancel reservation. |
| Actors | Member End User |
| Goal | Cancel a reservation of an event. |
| Pre-Conditions | 1. The Member has logged in successfully. 2. The Member has a reservation. |
| Steps | 1. The user opens the app. 2. The user taps on “My reservations” card. 3. The user taps on the reservation wished to be canceled. 4. The user taps on cancel button at the bottom of the reservation’s description. |
| Alternative | --- |
| Post-Conditions | The user will be redirected to the Swiss Engineering website to continue the cancelation process. |

|  |  |
| --- | --- |
| ID | 3.2 |
| Name | See my reservations. |
| Actors | Member End User |
| Goal | Open the list of the user’s reserved events. |
| Pre-Conditions | 1. The Member has logged in successfully. 2. The Member has a reservation(s). |
| Steps | 1. The user opens the app. 2. The user taps on “My reservations” card. |
| Alternative | --- |
| Post-Conditions | A list of the reserved events will be shown in a table. |

# literature references

[https://github.com/alexa/avs-device-sdk 2019-04-26](https://github.com/alexa/avs-device-sdk%202019-04-26)

<https://aws.amazon.com/lex/pricing/> 2019-04-26

<https://developer.apple.com/sirikit/> 2019-04-26  
[https://cloud.google.com/dialogflow-enterprise/pricing 2019-04-27](https://cloud.google.com/dialogflow-enterprise/pricing%202019-04-27)  
<https://developers.google.com/actions/distribute/> 2019-04-28

<https://www.swissengineering.ch/index.php?option=com_content&view=article&id=4&Itemid=6&lang=de> 2019-04-28

<https://xamarinhelp.com/interacting-google-assist-xamarin/> 2019-04-28

# honesty policy