



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	(Ελληνικά) Δημιουργία Προσωπικού Ψηφιακού Βοηθού για κινητές συσκευές. (Αγγλικά) Creation of a Personal Digital Assistant for mobile devices.
Όνοματεπώνυμο Φοιτητή	Χρήστος Γιαννούσης
Πατρώνυμο	Βασίλης Γιαννούσης
Αριθμός Μητρώου	ΜΠΣΠ/ 17013
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Ημερομηνία
Παράδοσης

04 / 02 / 2019

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Πατσάκης Κωνσταντίνος
Επίκουρος Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Ευχαριστίες

Η παρούσα διπλωματική πραγματοποιήθηκε στα πλαίσια του Μεταπτυχιακού προγράμματος Προηγμένα Συστήματα Πληροφορικής. Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή Ευθύμιο Αλέπη για την πολύτιμη βοήθεια και άριστη καθοδήγηση του, καθ' όλη τη διάρκεια εκπόνησης της εργασίας. Θα ήθελα επίσης να απευθύνω τις ευχαριστίες μου στον καθηγητή Χαράλαμπο Πατρικάκη, ο οποίος στάθηκε σημαντικός αρωγός στην προσπάθεια μου και μέντορας σε διάφορα στάδια της πορείας μου.

Περίληψη

Με την εξέλιξη της τεχνητής νοημοσύνης και του IoT έχουν έρθει στο προσκήνιο καινοτομίες και νέες εφαρμογές σε όλους τους τομείς. Εφαρμογές όπως τα έξυπνα σπίτια (Smart Homes), έξυπνες πόλεις (Smart Cities) και οι προσωπικοί ψηφιακοί βοηθοί έχουν αρχίσει σταδιακά να μπαίνουν και να αλλάζουν την καθημερινότητά μας. Μεγάλες ανταγωνίστριες εταιρίες, έχουν δημιουργήσει ήδη τους πρώτους και πολύ γνωστούς βοηθούς όπως είναι η Alexa της Amazon, ο Google Assistant της Google και η Siri από την Apple.

Ο στόχος αυτής της διπλωματικής είναι να μελετήσει τους προσωπικούς ψηφιακούς βοηθούς, την εξέλιξη τους καθώς και τις διάφορες εφαρμογές τους. Από την εκπαίδευση μέχρι τον πολιτισμό οι εφαρμογές των ψηφιακών βοηθών είναι πολλές και θα συνεχίσουν να αυξάνονται όσο βελτιώνεται και η τεχνολογία. Στη διαχείριση του σπιτιού όπου όλες οι συνδεδεμένες συσκευές θα ελέγχονται από το ψηφιακό βοηθό. Στο σχολείο όπου νέοι τρόποι εκπαίδευσης θα περιλαμβάνουν την εκμάθηση από βοηθούς chatbot. Στο μουσείο όπου οι επισκέπτες θα συνομιλούν με τα εκθέματα μέσω του ψηφιακού τους βοηθού.

Τέλος θα δούμε πως μπορούμε να κατασκευάσουμε μόνοι μας έναν δικό μας προσωπικό ψηφιακό βοηθό για κινητές συσκευές Android για την αναζήτηση και αναπαραγωγή μουσικής και βίντεο με τη χρήση του Spotify και YouTube, καθώς και ποιες τεχνολογίες και εργαλεία υπάρχουν διαθέσιμα αυτή τη στιγμή για τη δημιουργία ψηφιακών βοηθών σε μορφή Chatbot.

Abstract

With the evolution of Artificial Intelligence and IoT, many new technologies and applications have presented themselves. Applications such as Smart Homes, Smart Cities and personal digital assistants are steadily paving the way for the future. Their goal is to better our everyday lives and change the way we do things. Big companies have already invested and created several, very well-known assistants such as Alexa of Amazon, Google's Google Assistant or Apple's Siri. All of them have entered our lives, and one can find them in almost every house.

The goal of this Thesis is to study the personal digital assistants, their evolution and their capabilities. From Education all the way to Culture, the applicability of digital assistants is seemingly endless, due to the ever-growing technology and artificial intelligence.

In Smart homes, the personal digital assistant controls every device connected to it via the internet or Bluetooth: devices such as Smart TVs, the house's air-conditioning system and heating, or even the lighting of the house. In Education, new ways of teaching use digital assistants to change the learning process, making it more fun and easier for students to learn. In Museums, digital assistants can be used as means for "communicating" with the exhibits, making it possible to learn an exhibit's history through the exhibit's own voice and perspective.

Finally, we will explain how someone can create their own digital personal assistant for their Android devices, based on technologies and tools that are readily available. The assistant, Chatbot can search for and play music from Spotify and videos from YouTube.

Περιεχόμενα

1	Περίληψη.....	4
2	Abstract.....	5
3	Περιεχόμενα.....	6
4	Εισαγωγή.....	7
5	Ανασκόπηση πεδίου	9
	Ψηφιακός βοηθός Chatbot	9
	Alexa.....	12
	Google Assistant.....	12
	Siri.....	13
	Jarvis	13
6	Παρουσίαση και χρήση εφαρμογής	14
7	Αρχιτεκτονική συστήματος	16
	Watson.....	16
	Dialogflow	16
	Απάντηση Dialogflow, JSON.....	18
	Spotify SDK και API	20
	YouTube SDK και API	21
	Διαγράμματα UML	22
	Android.....	24
	Κώδικας του Manifest.xml	26
	Ασύγχρονες κλήσης της εφαρμογής, Spotify API.....	27
	YouTube API.....	28
	Κώδικας διαχείρισης Video Fragment.....	29
	User Interface	30
	Κώδικας διαχωρισμού των Views στον Adapter	31
	Ο xml κώδικας του Video Fragment.	32
	Κώδικας για την πλήρη οθόνη.....	34
8	Συμπεράσματα και μελλοντικές επεκτάσεις	35
9	Βιβλιογραφία	38

Εισαγωγή

Στην εποχή μας η τεχνολογία έχει κάνει τεράστια άλματα. Κάθε μερικούς μόλις μήνες εμφανίζεται και κάτι καινούργιο που σκοπό έχει να καλυτερέψει και να διευκολύνει την καθημερινότητά μας. Μεγάλες εταιρίες έχουν πέσει στον αστείρευτο αγώνα για καινοτομία και νέες τεχνολογίες. Τα τελευταία χρόνια το επίκεντρο της προσοχής των καταναλωτών προϊόντων τεχνολογίας έχει πέσει πάνω στις έξυπνες κινητές συσκευές (**smartphones**) με δύο βασικούς ανταγωνιστές τη Google και την Apple να ξεχωρίζουν.

Το [1] λειτουργικό σύστημα Android ιδρύθηκε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Στην αρχή το λογισμικό προοριζόταν για φωτογραφικές μηχανές κάτι που άλλαξε γρήγορα λόγω των δυνατοτήτων που είχε. Το 2005 η Google αγοράζει το Android και ξεκινά την κατασκευή του λειτουργικού συστήματος Android με πυρήνα Linux. Έτσι μέσα σε λίγα χρόνια η Google μαζί με μία κοινοπραξία μερικών άλλων μεγάλων εταιριών (Open Handset Alliance) όπως η Samsung, HTC, Motorola, φέρνει στο προσκήνιο της αγοράς τις έξυπνες κινητές συσκευές με λειτουργικό σύστημα το Android.

Η πρώτη έκδοση βγήκε στις 23 Σεπτεμβρίου 2008 και από τότε έχει δεχθεί πολλές αναβαθμίσεις διορθώνοντας αρκετά προβλήματα τόσο στην ομαλή λειτουργία του καθώς και στην απόδοση και ταχύτητά του. Όλες οι εκδόσεις παίρνουν το όνομά τους από κάποιο φημισμένο γλυκό με τις πρώτες εκδόσεις να φέρουν ονόματα όπως "Cupcake", "Donut", "Eclair" και "Froyo". Πλέον εν έτη 2019 το Android βρίσκεται στην έκδοση 9.0 με όνομα Pie, και δε θυμίζει σε τίποτα τις αρχικές του εκδόσεις έχοντας πλέον ξεπεράσει πολλά χρόνια προβλήματα που αντιμετώπιζε και με συνεχή υποστήριξη και καινούργιες ενημερώσεις να έρχονται τακτικά κάνοντας το ακόμα πιο ανταγωνιστικό και σταθερό.

Μία από τις πιο πρόσφατες καινοτομίες στο χώρο των κινητών συσκευών που κερδίζει συνεχώς έδαφος και ενδιαφέρον προς την εξέλιξη της είναι η προσθήκη και η χρήση των Ψηφιακών βοηθών. Η Google και το Android με την Google Assistant, η Apple και το iOS με την Siri, η Microsoft με την Cortana ανοίγουν ολοένα τον δρόμο για τη δημιουργία πιο έξυπνων, πιο γρήγορων και πιο σωστών ψηφιακών βοηθών.

Σημαντικό ρόλο σε όλη αυτή την ανάπτυξη των ψηφιακών βοηθών έχει παίξει η ραγδαία εξέλιξη και ανάπτυξη των συστημάτων τεχνητής νοημοσύνης (Artificial Intelligence, AI). Τα τελευταία χρόνια πολλές εταιρίες έχουν στρέψει την προσοχή τους στον ραγδαία αναπτυσσόμενο αυτό κλάδο της τεχνολογίας, μιας και που μπορεί να μας βοηθήσει να εξελιχουμε πολύ γρήγορα πολλούς άλλους τομείς και εκ συνήπια να διευκολύνουν ακόμα πιο πολύ την καθημερινότητά μας. Οι εφαρμογές αυτού του κλάδου είναι απεριόριστες. Η Google προσπαθεί τα τελευταία χρόνια να αναπτύξει αρκετά τον συγκεκριμένο κλάδο γιατί μπορεί να τη βοηθήσει στο έργο της για τη δημιουργία των έξυπνων πόλεων (Smart Cities), πόλεις δηλαδή που θα είναι τα πάντα συνδεδεμένα. Ήδη έχουν κάνει την εμφάνισή τους τα έξυπνα σπίτια (Smart Homes). Τα σπίτια αυτά παρέχουν πολλές δυνατότητες στους ιδιοκτήτες τους όπως είναι το άνοιγμα και το κλείσιμο μιας λάμπας, ο απομακρυσμένος έλεγχος μιας συσκευής, των παραθύρων, του θερμοστάτη της τηλεόρασης και πολλών άλλων. Όλα ελεγχόμενα από μια έξυπνη κινητή συσκευή (smartphone).

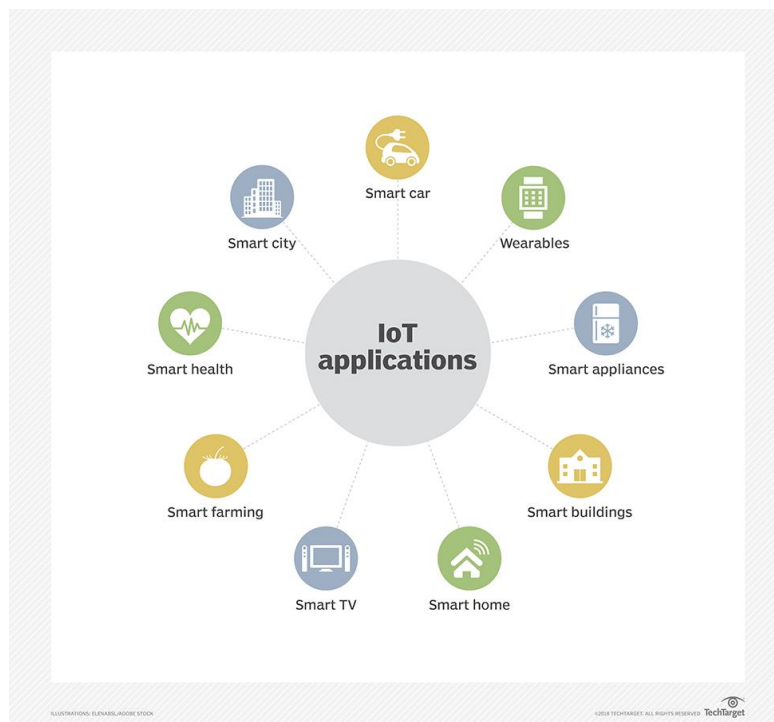
Τα τελευταία χρόνια τα Smart Homes έχουν εξαπλωθεί με μεγάλους ρυθμούς. Σύμφωνα με τελευταίες έρευνες [20] περίπου το 90% των Αμερικανών καταναλωτών έχει μια smart home συσκευή. Το 70% έχει αγοράσει ένα σύστημα όπως είναι η Alexa της Amazon ή το Google Home. Αυτό μας δείχνει πως η ιδέα του IoT έχει περάσει στους καταναλωτές και έχει αρχίσει να γίνεται κομμάτι της καθημερινότητας μας.

Το IoT ή αλλιώς το δίκτυο των πραγμάτων, είναι μια ιδέα για τη σύνδεση όλων των συσκευών έξυπνων και μη στο διαδίκτυο δημιουργώντας ένα οικοσύστημα. Οι συσκευές αυτές μπορεί να είναι απλές καθημερινές συσκευές όπως οι οικιακές συσκευές, τα αυτοκίνητα ακόμα και τα ηλεκτρολογικά ενός σπιτιού. Το IoT έχει γίνει ένας από τους πιο σημαντικούς τομείς της έρευνα και της καινοτομίας, λόγω των πολλών δυνατοτήτων που έχει. Η άνοδος αυτή έχει πυροδοτήσει την εξέλιξη των έξυπνων συσκευών και σπιτιών και φυσικά των έξυπνων βοηθών οι οποίοι βοηθούν στον έλεγχο των διάφορων συσκευών.

Οι εφαρμογές του IoT είναι πολλές [21]. Ξεκινάμε με το χώρο των καταναλωτών όπου βοηθούν στη μετατροπή μεγάλων εμπορικών καταστημάτων σε έξυπνα καταστήματα που διαφημίζουν τα προϊόντα στέλνοντας ειδοποιήσεις στις κινητές συσκευές των καταναλωτών που βρίσκονται μέσα σε ένα κατάστημα. Έπειτα στο χώρο των σπιτιών το συναντάμε στις έξυπνες συσκευές όπως τηλεοράσεις, ο φωτισμός, η θέρμανση του σπιτιού. Ουσιαστικά ολόκληρη η ιδέα του έξυπνου σπιτιού πηγάζει από την ιδέα του IoT. Μια επέκταση αυτού είναι η φροντίδα των ηλικιωμένων. Δηλαδή υπάρχουν συσκευές για την παρακολούθηση της υγείας και της κατάστασης των ηλικιωμένων ατόμων, οι οποίες μεταδίδουν μια ροή από δεδομένα η οποία είναι χρήσιμη για τους γιατρούς που παρακολουθούν την υγεία τους.

Μια ακόμα εφαρμογή του IoT είναι στα Μέσα Μαζικής Μεταφοράς. Τα τελευταία χρόνια έχουν κάνει την εμφάνισή τους οι έξυπνες στάσεις. Οι στάσεις αυτές μπορούν να γνωρίζουν την ακριβή θέση του λεωφορείου και του τρένου και να υπολογίζουν τον χρόνο άφιξής τους. Αυτό είναι εφικτό διότι η στάσεις είναι συνδεδεμένες στο διαδίκτυο και μπορούν να λαμβάνουν το στίγμα από τα GPS των μέσων μεταφοράς. Με αυτό τον τρόπο οι χρήστες των μέσων μπορούν να γνωρίζουν σε πόση ώρα θα περάσει το μέσο που περιμένουν ώστε να μειώνετε σημαντικά ο χρόνος αναμονής τους.

Επίσης τομείς όπως η γεωργία και η βιομηχανία το IoT έχει βοηθήσει πολύ στην περαιτέρω εξέλιξη τους. Στην γεωργία το IoT βοηθάει στην παρακολούθηση των καλλιεργειών μετρώντας στοιχεία όπως η υγρασία, η θερμοκρασία. Υπάρχουν εταιρίες στο χώρο οι οποίες παρέχουν τέτοιες υπηρεσίες σε αγρότες οι οποίοι με μια εφαρμογή στο κινητό τους παρακολουθούν τις σοδιές τους. Στην βιομηχανία το IoT βοηθά την παρακολούθηση της παραγωγής.



Εικόνα 1. IoT εφαρμογές [21].

Το θέμα με το οποίο θα ασχοληθεί η συγκεκριμένη πτυχιακή είναι η ανάπτυξη και η δημιουργία ενός Ψηφιακού Βοηθού. Οι ψηφιακοί βοηθοί έχουν αρχίσει να εξελίσσονται με γρήγορους ρυθμούς και τα επόμενα χρόνια θα βρίσκονται παντού. Θα δούμε πώς λειτουργούν οι ψηφιακοί βοηθοί και τα Chatbots και θα αναλύσουμε τη δημιουργία μιας εφαρμογής chatbot. Ο χρήστης της εφαρμογής μας θα μπορεί να ψάχνει με τη βοήθειά του για τραγούδια ή βίντεο που θέλει να βρει. Ο ψηφιακός βοηθός θα υλοποιηθεί με τη μορφή ενός Chatbot app για Android smartphones. Ο χρήστης θα συνομιλεί με το chatbot παρέχοντας του πληροφορίες πάνω στο τραγούδι ή στο βίντεο που θέλει να ακούσει ή να δει αντίστοιχα.

Ανασκόπηση πεδίου

Ψηφιακός βοηθός Chatbot

Τα Chatbots [2] είναι εφαρμογές που χρησιμοποιούν συστήματα τεχνητής νοημοσύνης για να επικοινωνήσουν και να επεξεργαστούν τα δεδομένα που τους παρέχει ο χρήστης. Οι χρήστες μιλούν με την εφαρμογή και αυτή με τη σειρά της προσπαθεί να ανιχνεύσει λέξεις κλειδιά ώστε να καταλάβει τί θέλει ο χρήστης να κάνει. Με τη βοήθεια του AI οι εφαρμογές των chatbot έχουν γίνει πιο έξυπνες δίνοντάς τους την ικανότητα να αντιλαμβάνονται τα ερωτήματα από τους χρήστες όπως και αν αυτοί τα διατυπώσουν.

Τα μέσα επικοινωνίας των Chatbots είναι οι εφαρμογές μηνυμάτων όπως τα SMS, οι εφαρμογές μηνυμάτων των Μέσων Κοινωνικής Δικτύωσης (Social Media Messaging Apps). Σήμερα το Facebook έχει ξεκινήσει την ενσωμάτωση των chatbots στη λειτουργία του. Δίνει τη δυνατότητα σε σελίδες μέσα στο Facebook να μπορούν να επικοινωνούν με τους επισκέπτες τους χωρίς την επίβλεψη των διαχειριστών. Με τη σειρά τους οι επισκέπτες λαμβάνουν από αυτό αρκετές πληροφορίες που αφορούν τη σελίδα. Ένα άλλο παράδειγμα χρήσης Chatbot είναι και το κλείσιμο εισιτηρίων για κάποια παράσταση ακόμα και για αεροπορικά ή ακτοπλοϊκά εισιτήρια.

Στο τομέα της εκπαίδευσης, τα Chatbots έχουν αρχίσει και ανεβαίνουν με μεγάλη ταχύτητα. Εφαρμογές που σκοπό έχουν να βοηθήσουν τον εκπαιδευόμενο/μαθητή να κατανοήσει διάφορα γνωστικά αντικείμενα σε διάφορους κλάδους όπως είναι η φυσική, τα μαθηματικά, η γλώσσες και η χημεία. Αυτό συνήθως γίνεται με εφαρμογές παιχνιδοποίησης (gamification) οι οποίες μετατρέπουν τη μελέτη σε ένα παιχνίδι γνώσεων μέσα από διάφορες δραστηριότητες – παιχνίδια γνώσεων στα οποία επιβλέπον αρχή είναι ένα Chatbot. Με αυτό το τρόπο οι μαθητές μαθαίνουν παίζοντας κάνοντας την εμπειρία της μάθησης πιο ελκυστική και απολαυστική. Το Chatbot μέσα από ερωτήσεις που κάνει στο χρήστη καταλαβαίνει το επίπεδό του και προσαρμόζει το βαθμό δυσκολίας του κάθε παιχνιδιού. Αυτό γίνεται με τη χρήση τεχνολογιών όπως είναι το Machine Learning (ML), Natural Language Processing (NLP).

Τον Ιούνιο του 2014 [6] εκδόθηκε ένα επιστημονικό άρθρο (paper) για το πόσο ελκυστικό θα ήταν για τους μαθητές η χρήση ενός Chatbot στη διαδικασία της εκπαίδευσης. Η ομάδα κατασκεύασε ένα Chatbot για την διδασκαλία του μαθήματος για την επιστήμη των υπολογιστών, για να δει πόσο θα αυξηθεί το ενδιαφέρον για την επιλογή του συγκεκριμένου μαθήματος. Το Chatbot χρησιμοποιούσε μια μέθοδο gamification ώστε να κάνει πιο ενδιαφέρουσα την εμπειρία του προγραμματισμού.

Σύγκριναν λοιπόν το Chatbot με το πρόγραμμα Alice, το οποίο χρησιμοποιούσαν για την εκμάθηση του μαθήματος, μέσα από έναν διαδικτυακό διαγωνισμό προγραμματισμού και βρήκαν πως τα ποσοστά των ατόμων που χρησιμοποίησε το Chatbot για την επίλυση του προβλήματος και τελικός παρέδωσε κάποια λύση στην διαδικτυακή πλατφόρμα ήταν μεγαλύτερο από το ποσοστό που χρησιμοποίησε το πρόγραμμα Alice. Επίσης, τα άτομα βρήκαν πιο ελκυστική τη χρήση του Chatbot από την Alice, δείχνοντας μεγαλύτερα ποσοστά ενθουσιασμού, διασκέδασης και συμμετοχής στην επίλυση του προβλήματος. Αυτή η έρευνα δείχνει πόσο πολύ βοηθάει στην εκπαιδευτική διαδικασία ένα Chatbot με gamification.

Το 2018 [5] οι εταιρίες Drift, SurveyMonkey Audience, Salesforce και myClever πραγματοποίησαν μια έρευνα για το πώς τα Chatbots αλλάζουν για τους καταναλωτές το τομέα των πωλήσεων και της επικοινωνίας με τις επιχειρήσεις. Το δείγμα αποτελούνταν από 1051 ενήλικες με ηλικίες από 18 μέχρι 64 και η έρευνα διήρκεσε μία εβδομάδα.

Τα αποτελέσματα της έρευνας έδειξαν πως ολοένα και περισσότερες επιχειρήσεις χρησιμοποιούν τα Chatbots για να συνομιλήσουν με τους πελάτες τους. Επίσης, βρήκαν πως το 15% των επισκεπτών στις ιστοσελίδες τους, προτίμησε τη χρήση Chatbot από το τηλέφωνο το οποίο έχει και την πλειοψηφία με 60%. Αυτό το ποσοστό όμως αναμένεται να αλλάξει και το ποσοστό της χρήσης των Chatbots να ανέβει τα επόμενα χρόνια πολύ περισσότερο.

Business Communication Channels

How have you communicated with businesses in the past 12 months?

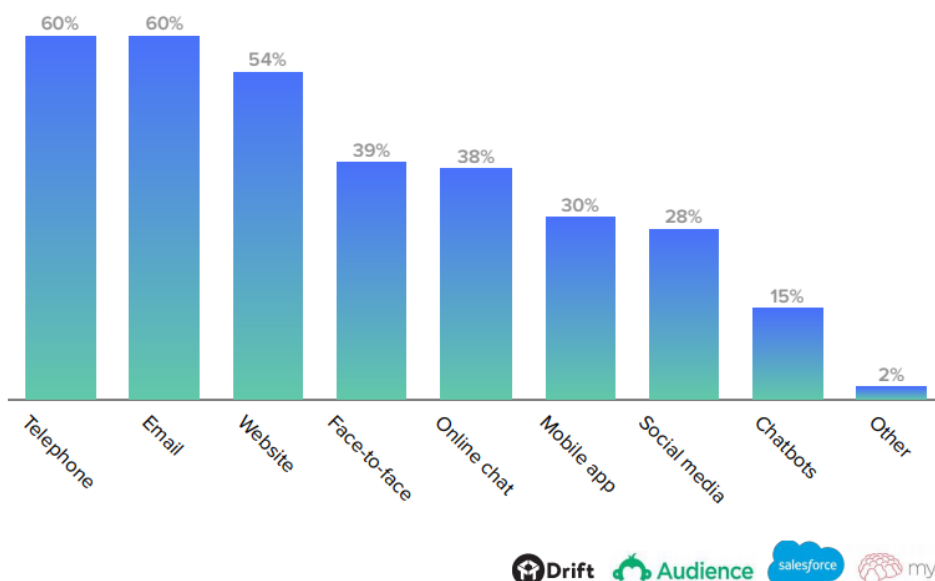


Figure 1. Ποσοστά της έρευνας για τη χρήση chatbot ως μέσο επικοινωνίας με επιχειρήσεις του έτους 2018 [5].

Επίσης αναφέρουν και ορισμένα από τα προβλήματα που αντιμετωπίζουν οι χρήστες των διαδικτυακών υπηρεσιών, με βασικότερο αυτή της δύσκολης πλοήγησης σε μια ιστοσελίδα. Οι ιστοσελίδες γίνονται όλο και πιο περίπλοκες, με δύσκολη πλοήγηση λόγω των πολλών συνδέσμων και των κρυμμένων λειτουργιών. Αυτό αναγκάζει τους χρήστες να σπαταλούν πολύ χρόνο αναζητώντας μέσα στις ιστοσελίδες.

Problems With Traditional Online Experiences

What frustrations have you experienced in the past month?

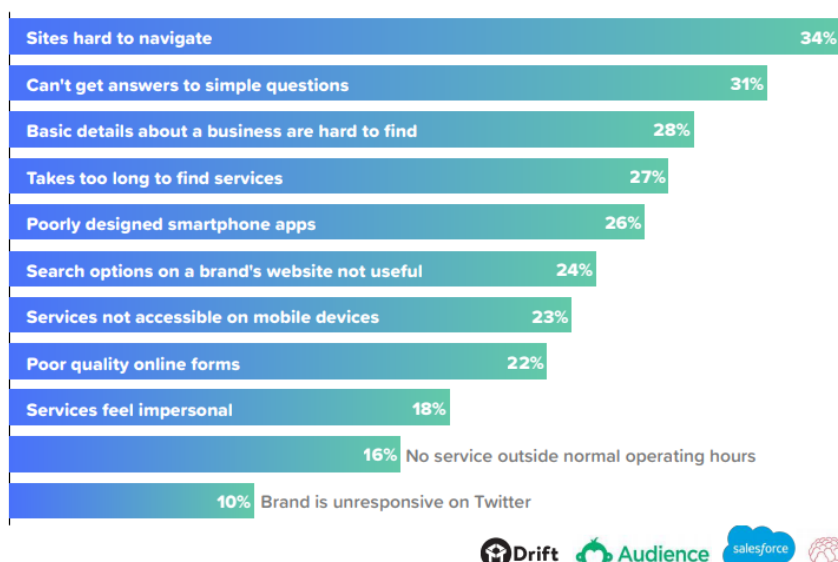


Figure 2. Προβλήματα στην παραδοσιακή πλοήγηση σε μια ιστοσελίδα [5].

Παίρνοντας τα παραπάνω προβλήματα η ομάδα προσπάθησε να βρει τα πλεονεκτήματα που έχουν τα chatbots και λύνουν τα παραπάνω προβλήματα αλλά και για πιο

λόγο θα τα χρησιμοποιούσαν οι χρήστες. Με τα chatbot η πλοήγηση σε μια ιστοσελίδα μπορεί να γίνει πολύ πιο εύκολη. Οι χρήστες πολύ απλά μπορούν να ζητούν από το chatbot τί ακριβώς ψάχνουν και αυτό να τους βοηθάει παραπέμποντάς τους στο σωστό σημείο της ιστοσελίδας. Επίσης μπορούν να συντομεύσουν την προσπάθεια των επισκεπτών από το να πάρουν τις πληροφορίες που ζητάνε, χωρίς οι χρήστες να χρειάζεται να στείλουν κάποιο email για να πάρουν πληροφορίες από τον διαχειριστή ή τον υπεύθυνο της ιστοσελίδας μπορούν απλά να μιλήσουν με το chatbot και να το ρωτήσουν και να μάθουν από αυτό για την ιστοσελίδα και τα προϊόντα της. Ένα άλλο θετικό είναι πως με τα chatbot έχουμε συνεχόμενη παροχή υπηρεσιών με 24ωρη κάλυψη. Έτσι έχουμε εύκολη και γρήγορη επικοινωνία με άμεσες απαντήσεις.

Predicted Use Cases for Chatbots

What do you predict you would use a chatbot for?



Figure 3. Πώς θα αξιοποιούσαν ένα chatbot οι χρήστες [5].

Εκτός όμως από την έρευνα αυτή, η ίδια η Google έχει συλλέξει τα δικά της δεδομένα πάνω στα chatbots και συγκεκριμένα στη δημοτικότητά τους τα τελευταία χρόνια, με βάση τις αναζητήσεις των χρηστών της.

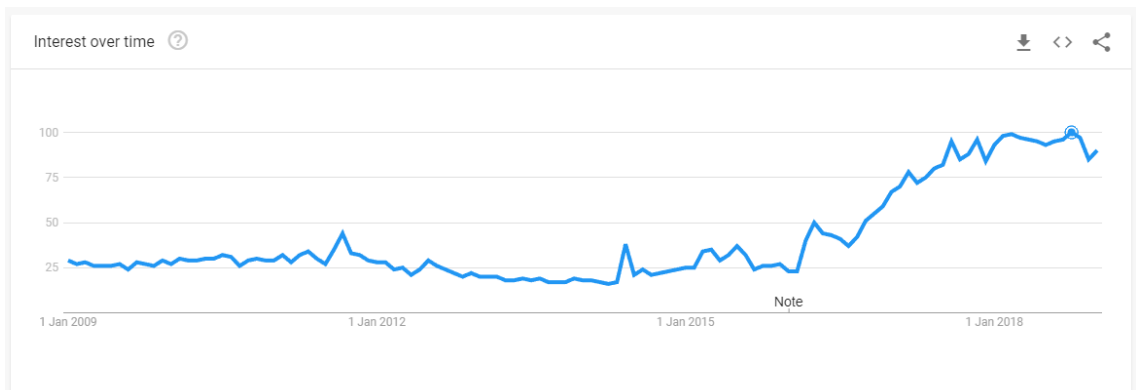


Figure 4. Δημοτικότητα των chatbot με βάση τα στοιχεία της Google

Πολλές είναι οι εταιρίες που παρέχουν ψηφιακούς βοηθούς και όλες κάνουν χρήση των chatbot. Η Amazon έχει την Alexa, η Google τον Google Assistant, η Apple με τη Siri και άλλες. Οι συγκεκριμένοι ψηφιακοί βοηθοί είναι και οι πιο γνωστοί. Όλες οι συσκευές android έρχονται

με προ εγκατεστημένο τον Google Assistant, όλες οι Apple συσκευές έχουν μέσα τους την Siri και οι συσκευές Windows περιέχουν την Cortana.

Το Νοέμβριο του 2018 [7] βγήκε ακόμα μία έρευνα πάνω στις συσκευές που χρησιμοποιούν κάποιο ψηφιακό βοηθό και την ραγδαία εξάπλωση τους και πώς αυτές χρησιμοποιούνται από τους χρήστες τους. Καταγράφει το μερίδιο αγοράς ανάμεσα σε εταιρίες παροχής τέτοιων συσκευών με ψηφιακούς βοηθούς και βλέπει τα στατιστικά της χρήσης τους. Το σημαντικό στοιχείο σύμφωνα με τους χρήστες και το μεγαλύτερο σε αξία είναι η αναπαραγωγή μουσικής όπως φαίνεται παρακάτω figure 6.

Top/most valued features on smart speakers (UK)

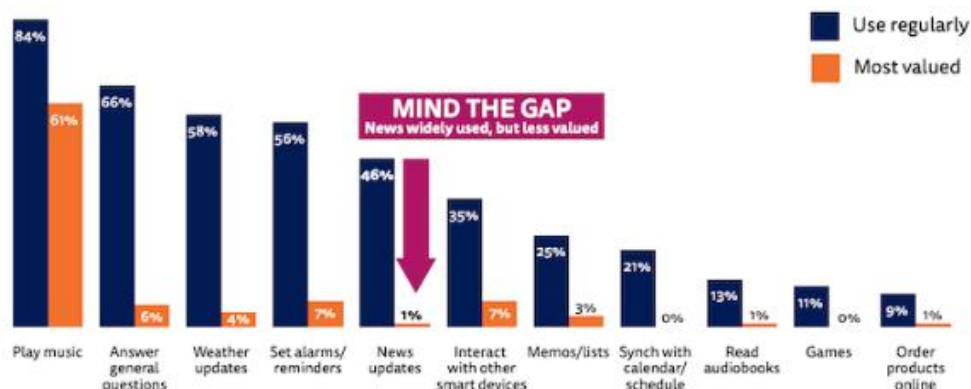


Figure 5. Στατιστικά χρήσης των συσκευών με έξυπνους βοηθούς [7].

Alexa

Η Alexa της Amazon [4] είναι μια ψηφιακή βοηθός που δημιουργήθηκε το Νοέμβριο του 2014 από την Amazon. Η ίδια η εταιρία έχει πει πως εμπνεύστηκε από τον κεντρικό υπολογιστή του διαστημόπλοιου Enterprise της διάσημης σειράς επιστημονικής φαντασίας Star Trek. Έκτοτε έχει προωθηθεί ως βοηθός σε επιχειρήσεις και σπίτια καθημερινών χρηστών.

Οι δυνατότητες της είναι πολλές. Από την απάντηση σε ερωτήματα μέχρι στην υλοποίηση εργασιών. Η Alexa έφερε την έννοια του έξυπνου σπιτιού πιο κοντά στον απλό πολίτη. Πλέον οποιοσδήποτε μπορεί να βάλει την στο σπίτι του και με μικρή προσπάθεια να το μετατρέψει σε ένα έξυπνο. Μπορεί να ελέγχει τη θερμοκρασία του σπιτιού έχοντας πρόσβαση στα κλιματιστικά ή τον κεντρικό θερμοστάτη του σπιτιού, να ελέγχει τα συστήματα συναγερμού, να ανοίγει την τηλεόραση, να αναπαράγει μουσική και όλα αυτά από απόσταση. Μπορεί ακόμα και να ελέγχει ηλεκτρικές συσκευές όπως είναι η μηχανή του καφέ και ο φούρνος. Επίσης οι χρήστες μπορούν να ελέγχουν το καθημερινό πρόγραμμά τους, να ρυθμίζουν ραντεβού και πληρωμές, να κάνουν τηλεφωνικές ή και διαδικτυακές κλήσεις και να στέλνουν emails και μηνύματα. Όλα μέσα από την συνομιλία τους με την Alexa.

Το μέλλον της Alexa είναι ακόμα μεγάλο καθώς η Amazon έχει δώσει τη δυνατότητα στους χρήστες και στους προγραμματιστές να δημιουργούν καινούργια ταλέντα – ικανότητες για την Alexa (Alexa Skills). Με αυτό το τρόπο οι δυνατότητές της επεκτείνονται με γρήγορους ρυθμούς με νέες ικανότητες να κάνουν την εμφάνισή τους, όπως η παραγγελία φαγητού αλλά και η πρόσβαση σε πολλών ειδών αισθητήρων. Οι χρήστες μπορούν να επικοινωνήσουν μαζί της είτε μέσω εφαρμογής για κινητά είτε από την δική της ηλεκτρονική συσκευή που ονομάζεται Echo dot.

Google Assistant

Ο Google Assistant [8] παρουσιάστηκε τον Μάιο του 2016 ως κομμάτι του Google Home, μια συσκευή παρόμοια με το Echo dot για την Alexa της Amazon, για έξυπνα σπίτια (smart home). Για κινητές συσκευές στην αρχή εκδόθηκε μόνο για τις συσκευές Pixel και Pixel

XL της Google. Μετά από 1 χρόνο ο Google Assistant πέρασε και στις υπόλοιπες android συσκευές με έκδοση λογισμικού Marshmallow και πάνω.

Ο βοηθός της Google έχει αρκετές δυνατότητες. Μπορεί να παίζει μουσική, βίντεο ακόμα και ταινίες μέσω του Google Movies. Ο χρήστης συνομιλεί μαζί του μέσω φωνητικών εντολών. Έχει την ικανότητα να αναζητάει και να αποσπά πληροφορίες για τα πάντα από το διαδίκτυο, να πει το καιρό και τις τελευταίες ειδήσεις. Έχει επίσης τη δυνατότητα να ελέγχει κομμάτια υλικού της ίδιας της συσκευής όπως είναι το άνοιγμα και το κλείσιμο του Wi-fi, του Bluetooth, να ανοίγει την κάμερα της συσκευής. Μπορεί να επικοινωνεί με τρίτες εφαρμογές όπως την εφαρμογή για Email του κινητού για την αποστολή μηνυμάτων και γενικά να ανοίγει και να κλείνει εφαρμογές.

Ο Google Assistant είναι αυτή τη στιγμή ένας από τους καλύτερους, δεύτερος μόνο στη Alexa της Amazon, για smartphone συσκευές προσφέροντας πολλές δυνατότητες στους χρήστες του.

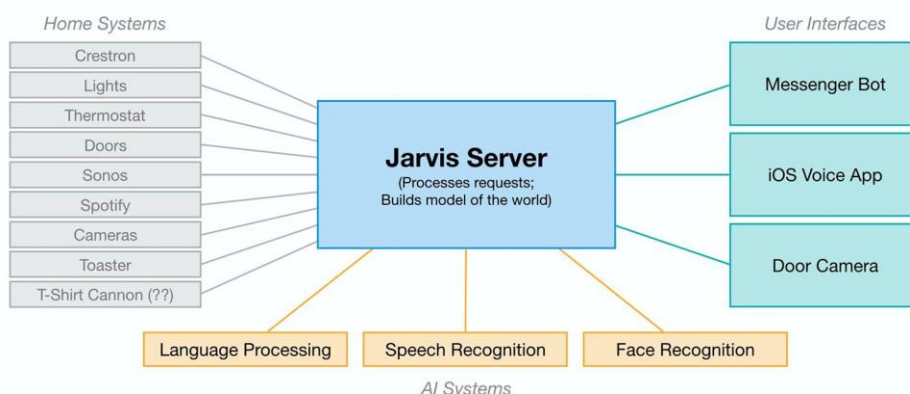
Siri

Η Siri [12] είναι το αποτέλεσμα της συνεργασίας αρκετών εταιριών και του Υπουργείου Άμυνας των Ηνωμένων Πολιτειών, που είχε ως σκοπό τη δημιουργία ενός προσωπικού βοηθού με μηχανική μάθηση. Το 2010 η Apple αγόρασε την Siri, την τελειοποίησε και την ενσωμάτωσε στο λειτουργικό της iOS για όλες τις συσκευές της, ως έναν έξυπνο προσωπικό βοηθό.

Όπως και ο Google Assistant έτσι και η Siri [11] έχει ικανότητες όπως είναι η αποστολή μηνυμάτων, η πραγματοποίηση κλήσεων, η ενημέρωση για το καιρό, να ψάχνει πληροφορίες στον παγκόσμιο ιστό και να αλληλοεπιδρά με άλλες εφαρμογές στη συσκευή. Μπορεί επίσης να ελέγχει και αυτή το υλικό των συσκευών όπως είναι το Wi Fi, το Bluetooth και την κάμερα. Επίσης η Siri υποστηρίζει και λειτουργίες έξυπνου σπιτιού όπως είναι ο έλεγχος των καμερών ασφαλείας του σπιτιού, ο έλεγχος της τηλεόρασης και του φωτισμού του σπιτιού.

Jarvis

Το 2016 ο Marc Zuckerberg [10] έφτιαξε έναν βοηθό με τεχνητή νοημοσύνη και τον ονόμασε Jarvis, εμπνευσμένο από τον Jarvis τον προσωπικό βοηθό του Iron Man. Ο βοηθός αυτός αν και σε πρώιμο στάδιο έχει τη δυνατότητα να ελέγχει το σπίτι όπως τα φώτα, η θερμοκρασία, την ασφάλεια και να μαθαίνει τις συνήθειες του χρήστη.



Εικόνα 2 Σχεδιάγραμμα λειτουργίας του Jarvis [10].

Ο Jarvis χρησιμοποιεί Natural Language Processing, αναγνώριση φωνής και προσώπου. Ο κώδικάς του είναι γραμμένος σε Python, PHP και Objective C. Ο έλεγχος του γίνεται είτε μέσω φωνητικών εντολών είτε μέσω γραπτών μηνυμάτων. Με τη χρήση της αναγνώρισης της φωνής και του προσώπου είναι σε θέση να γνωρίζει ποιος του μιλάει κάθε φορά και να προσαρμόζει αναλόγως τις απαντήσεις του με βάση τα χαρακτηριστικά και τις

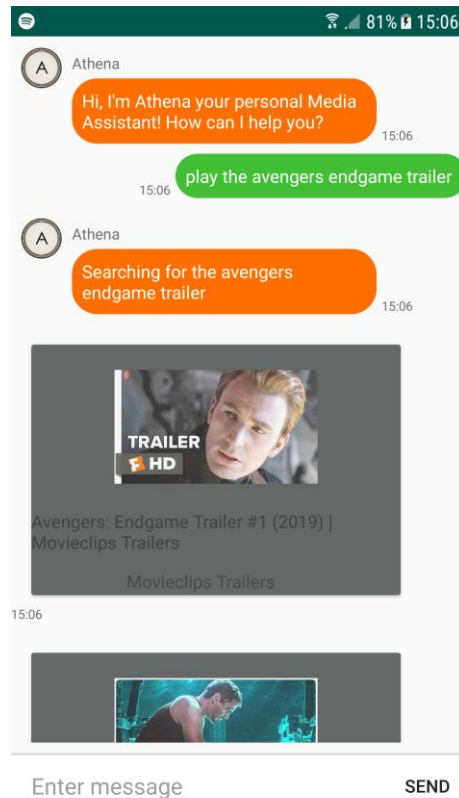
συνήθειες που έχει μάθει για κάθε χρήστη. Επίσης τον βοηθά να ελέγχει αποτελεσματικότερα την ασφάλεια του σπιτιού.

Για την ώρα ο Jarvis δεν έχει βγει στη διάθεση του ευρύ κοινού. Όμως ο δημιουργός του έχει πει πως έχει σχέδια για την τροποποίηση και βελτίωσή του ώστε μια μέρα να γίνει πιθανότατα το επόμενο μεγάλο του εγχείρημα.

Παρουσίαση και χρήση εφαρμογής

Η εφαρμογή ονομάζεται Athena, από το όνομα της ψυφιακού βοηθού που δημιουργήθηκε στο Dialogflow, Athena. Ο χρήστης, όπως προαναφέραμε, μπορεί να το χρησιμοποιήσει για να βρει και να αναπαράγει μουσική από το Spotify και βίντεο από το YouTube.

Όταν ο χρήστης ξεκινήσει την εφαρμογή, πραγματοποιείται η σύνδεση της Athena με την εφαρμογή του Spotify στη συσκευή μας, για να γίνει η ταυτοποίηση με το λογαριασμό του χρήστη. Αφού γίνει η σύνδεση, ο χρήστης πλέον είναι σε θέση να συνομιλήσει με το chatbot.

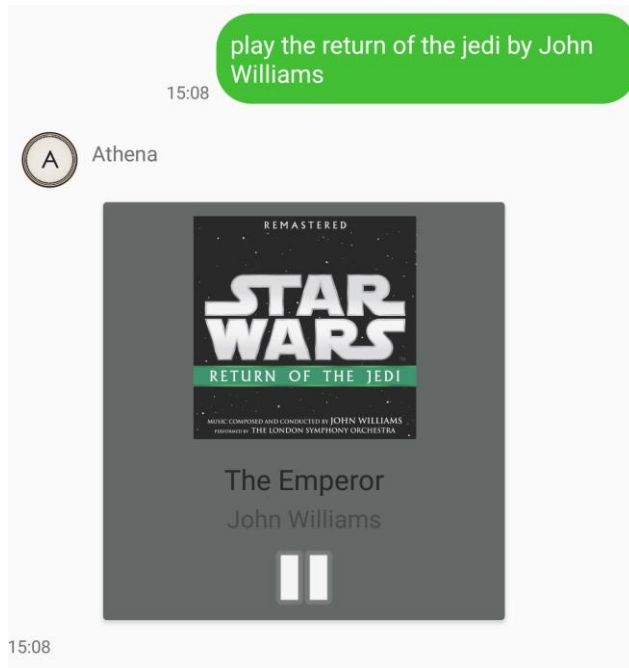


Εικόνα 3. UI χρήστη.

Όπως βλέπουμε και από την εικόνα 3 το περιβάλλον του χρήστη (UI) είναι αυτό μιας κλασσικής εφαρμογής μηνυμάτων. Ο χρήστης πατώντας πάνω στο πεδίο που γράφει “Enter message” μπορεί να γράψει το μήνυμα που θέλει και πατώντας το κουμπί “Send”, να το στείλει στην Athena. Εδώ πρέπει να επισημανθεί πως ο χρήστης αν αφήσει κενό το πεδίο εισαγωγής του μηνύματος και πατήσει μόνο το κουμπί “Send” τότε η εφαρμογή ενεργοποιεί την ηχογράφηση της φωνής ώστε να καταγράψει το τί θα πει ο χρήστης στην Athena. Ο χρήστης ειδοποιείται από έναν πρώτο ήχο πως η ηχογράφηση έχει ξεκινήσει και μόλις ο χρήστης διατυπώσει την πρότασή του, τότε ακούγεται ένας δεύτερος ήχος που σηματοδοτεί τη λήξη της ηχογράφησης και την αποστολή του μηνύματος στην Athena.

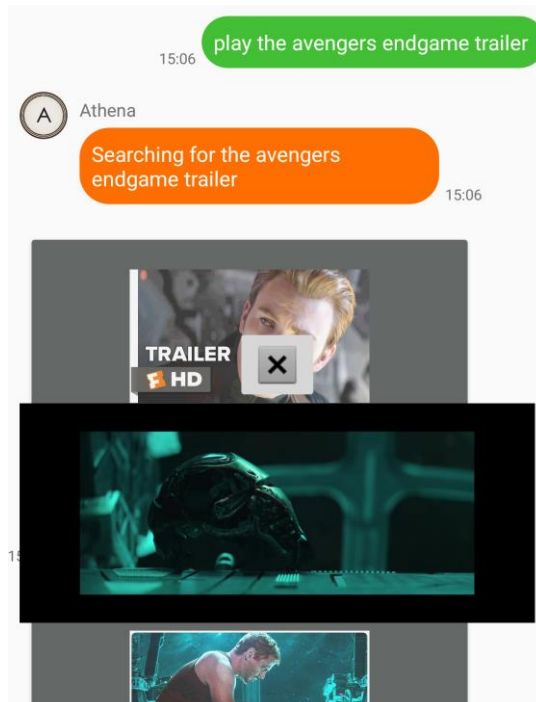
Ο χρήστης μπορεί να ακούσει μουσική από Spotify λέγοντας στην Athena το τίτλο του τραγουδιού και το όνομα του καλλιτέχνη. Με αυτόν το τρόπο περιορίζουμε σημαντικά το

σφάλμα στην αναζήτηση του Spotify για κάποιο τραγούδι. Αν η Athena βρει στο Spotify αυτό που ζήτησε ο χρήστης τότε στο UI εμφανίζεται μια μικρή κονσόλα με την φωτογραφία του Album το όνομα του κομματιού και του καλλιτέχνη μαζί με ένα pause/play κουμπί.



Εικόνα 4. Spotify κονσόλα διαχείρισης.

Αν η Athena δεν αναγνωρίσει το όνομα κάποιου καλλιτέχνη τότε αναζητάει αυτό που είπε ο χρήστης στο YouTube επιστρέφοντας τα 5 πρώτα αποτελέσματα σε μορφή βίντεο. Ο χρήστης στη συνέχεια μπορεί να επιλέξει πιο από τα 5 βίντεο θέλει να δει. Τότε εμφανίζεται μια κονσόλα βίντεο του YouTube η οποία αναπαράγει το βίντεο που επιλέχθηκε. Ο χρήστης μπορεί να κλείσει το βίντεο ανά πάσα στιγμή πατώντας το "X" κουμπί πάνω από τη κονσόλα του βίντεο.



Εικόνα 5. YouTube κονσόλα διαχείρισης.

Αρχιτεκτονική συστήματος

Υπάρχουν πολλές εταιρίες που παρέχουν και εργαλεία για την ανάπτυξη και την κατασκευή Chatbot. Η IBM προσφέρει με πληρωμή την πλατφόρμα τις για την υλοποίηση chatbot και το γνωστό της σύστημα τεχνητής νοημοσύνης Watson. Τέλος ήρθε και η Google με το Api.ai το οποίο αργότερα άλλαξε ονομασία και πήρε το όνομα Dialogflow. Παρακάτω παρουσιάζονται ορισμένα στοιχεία για τα δύο αυτά εργαλεία. Εμείς στην υλοποίησή χρησιμοποιήσαμε το Dialogflow διότι η χρήση του είναι απεριόριστη και δωρεάν.

Watson

Ο Watson [3] είναι ένα πληροφοριακό σύστημα που δημιούργησε η IBM και ονόμασε από τον πρώτο της CEO Thomas J. Watson. Στην αρχή φτιάχτηκε ως ένα σύστημα απαντήσεων σε ερωτήσεις χρησιμοποιώντας συστήματα προηγμένης επεξεργασίας φυσικής γλώσσας (Natural Language Processing), ανάκτησης πληροφορίας (Information Retrieval), μηχανικής μάθησης (Machine Learning) και άλλων συστημάτων και μοντέλων.

Αργότερα και βλέποντας τις δυνατότητες του Watson, η IBM εξέλιξε και διεύρυνε τις δυνατότητες του. Πλέον είναι ένα από τα βασικότερα συστήματα πληροφορίας που διαθέτει η IBM η οποία τον προωθεί στον κλάδο της ασφάλειας, στην ιατρική σε τομείς όπως η διάγνωση και ανίχνευση καρκίνου, σε chatbots και σε μεγάλες εταιρίες.

Dialogflow

Το Dialogflow [9] λειτουργεί με 2 βασικά στοιχεία, χαρακτηριστικά. Τα ερωτήματα – εντολές των χρηστών που τα ονομάζει Intents και οι λέξεις κλειδιά που απαρτίζουν μια οντότητα, αυτά τα σύνολα - οντότητες τα ονομάζει Entities.

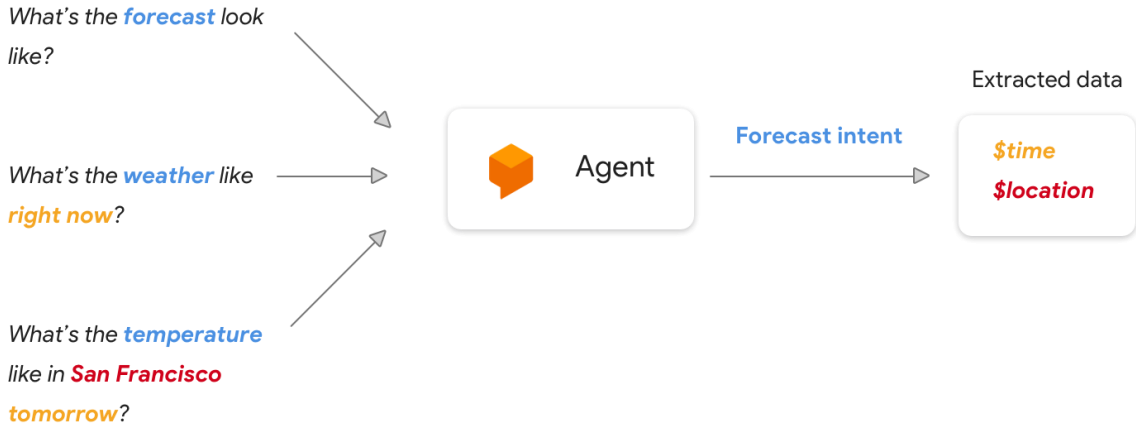
Τα Entities αποτελούνται από λέξεις που πιθανό είναι να χρησιμοποιήσει ένας χρήστης για να περιγράψει κάποιο αντικείμενο, χαρακτηριστικό, εντολή, να διαμορφώσει ένα ερώτημα. Για παράδειγμα ένα entity με το όνομα Music_Genres (είδη μουσικής), περιέχει μέσα του ονόματα από όλα τα είδη μουσικής. Αντίστοιχα ένα entity με όνομα MyArtists (καλλιτέχνες) περιλαμβάνει ονόματα από καλλιτέχνες της μουσικής βιομηχανίας όπως ονόματα τραγουδιστών. Εκτός από τα Entities που δημιουργούμε εμείς υπάρχουν και έτοιμα του συστήματος (system entities) για κάποιες βασικές λειτουργίες όπως η ώρα, η ημερομηνία και άλλες.

The screenshot shows the Dialogflow console interface. At the top, there is a search bar for training phrases. Below it, a list of training phrases is displayed, each with a 'play' button and a 'like it' button. The phrases are: 'play i like it by Cardi', 'play lies by Marina', 'play song cry by Jay-Z', 'i want to listen to Poker Face by Lady Gaga', and 'play Halo by Beyonce'. Below the list, a table shows the resolved values for the intents. The table has three columns: PARAMETER NAME, ENTITY, and RESOLVED VALUE. The rows are: actions (resolved to 'play'), any (resolved to 'i like it'), and MyArtists (resolved to 'Cardi').

PARAMETER NAME	ENTITY	RESOLVED VALUE
actions	@actions	play
any	@sys.any	i like it
MyArtists	@MyArtists	Cardi

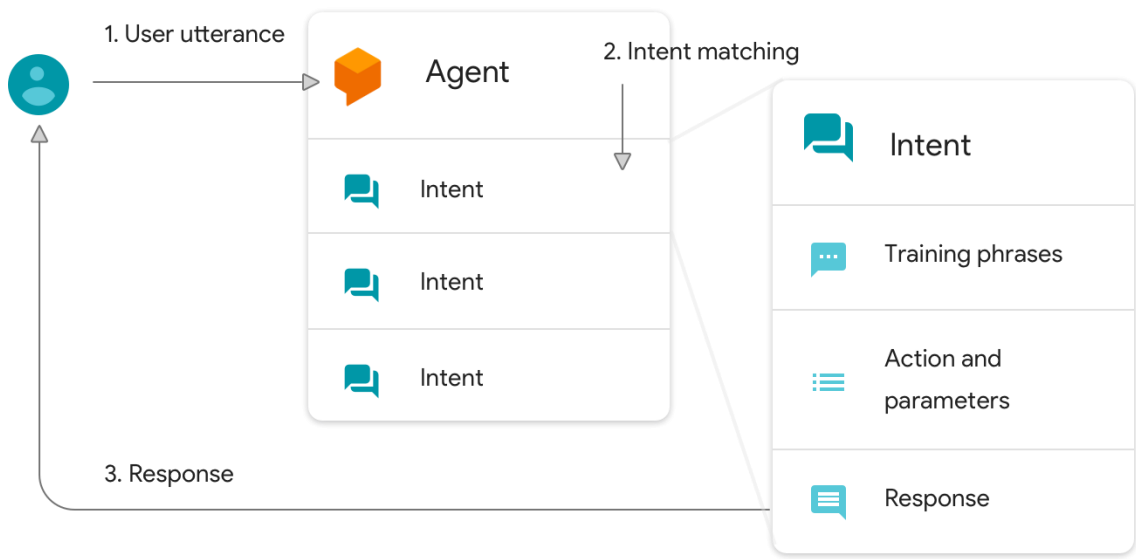
Εικόνα 6. Κοנסόλα Dialogflow. Διαχείριση Intents για αναπαραγωγή συγκεκριμένου τραγουδιού.

Τα Intents δεν είναι τίποτα άλλο από τα ερωτήματα των χρηστών προς το chatbot που συνδέονται με μια απάντηση – δραστηριότητα (answer- action). Τα Intents για το Dialogflow περιέχουν μέσα τους Entities που πιθανό να έχουμε ορίσει εμείς. Ένα παράδειγμα intent για την εφαρμογή μας είναι “ Play someone like you by Adele ”. Στο συγκεκριμένο παράδειγμα έχουμε έναν καλλιτέχνη με όνομα “Adele”, άρα το entity με τα ονόματα των καλλιτεχνών αντιστοιχίζεται με το “Adele” εάν το όνομα αυτό περιέχεται μέσα στην οντότητα. Επίσης έχουμε τη λέξη “Play” που αντιστοιχεί σε entity με ονομασία Actions όπου σε αυτό έχουμε ορίσει λέξεις που δείχνουν προς μία εργασία που πρέπει να κάνει η εφαρμογή μας. Και τέλος ότι περιέχεται ανάμεσα στο entity Actions και στο όνομα του καλλιτέχνη το έχουμε ορίσει ως ένα system entity με όνομα “any” για τους τίτλους των τραγουδιών, ώστε οτιδήποτε μεταξύ των 2 entities αποτελεί τον τίτλο του κομματιού.



Εικόνα 7. Παράδειγμα Intents. Οι λέξεις με χρώμα λέξεις αναγνωρισμένες ως Entities.

Για να μπορέσει το chatbot να μάθει τα διάφορα Intents θα πρέπει να τον ορίσουμε τουλάχιστον 5 διαφορετικά παραδείγματα για το καθένα, ώστε να μάθει να καταλαβαίνει ποια κατηγορία intent χρησιμοποιεί ο χρήστης. Αν κάποιο intent δεν γίνει αντιληπτό, το bot θα χρησιμοποιήσει το default intent και το γνωστό σε όλους “Δεν κατάλαβα το ερώτημα, προσπάθησε ξανά”. Αν το intent γίνει αντιληπτό, μας επιστρέφεται η αντίστοιχη απάντηση που του έχουμε ορίσει. Η απάντηση μπορεί να περιέχει μεταβλητές που προέρχονται από τα entities που αναγνωρίστηκαν από το ερώτημα του χρήστη. Στο προηγούμενο παράδειγμα οι μεταβλητές θα ήταν το όνομα του καλλιτέχνη “Adele”, το όνομα της δράσης “play” και το τραγούδι “someone like you”.



Εικόνα 8. Κύκλος εργασιών του chatbot στο Dialogflow.

Για την σύνδεση του Dialogflow με το βασικό κορμό της εφαρμογής χρησιμοποιήθηκε το Android SDK του Dialogflow το οποίο μας βοήθησε μέσα από έτοιμες λειτουργίες να κάνουμε τη σύνδεση της εφαρμογής μας με τον Chatbot Agent που έχει δημιουργηθεί και ονομαστεί Athena. Για να μπορέσουμε όμως να συνδεθούμε με την Athena πρέπει πρώτα να πάμε στη κονσόλα του Dialogflow και στις ρυθμίσεις της Athena να πάμε και να πάρουμε το " Client access token " το κλειδί για την πρόσβαση μας στο API του Dialogflow. Το πρώτο πράγμα που γίνεται με το που ξεκινήσει η εφαρμογή είναι να συνδεθεί με την Athena αποστέλλοντας το παραπάνω κλειδί. Αφού γίνει η σύνδεση μπορούμε πλέον να στέλνουμε τα μηνύματα (input) του χρήστη και να λαμβάνουμε ένα JSON με την όλη την πληροφορία από το Dialogflow.

Απάντηση Dialogflow, JSON

```
1. {
2. "responseld": "2619727d-3531-40de-9fac-346e3859fbcd",
3. "queryResult": {
4. "queryText": "play someone like you by Adele",
5. "parameters": {
6. "actions": "play",
7. "any": "someone like you",
8. "MyArtists": "",
9. "music-artist": [
10. "Adele"
11. ]
12. },
13. "allRequiredParamsPresent": true,
14. "fulfillmentMessages": [
15. {
16. "text": {
17. "text": [
18. ""
19. ]
20. }
21. }
22. ],
23. "intent": {
24. "name": "projects/moviesassistant-224410/agent/intents/a358a5db-38d6-4fb0-9acd-7ad34231d569",
25. "displayName": "play specific track"
26. },
27. "intentDetectionConfidence": 0.9403985,
28. "languageCode": "en"
29. }
30. }
```

Όπως βλέπουμε και στην απάντηση του Dialogflow, η είσοδος που έδωσε ο χρήστης είναι το πεδίο queryText και από κάτω βρίσκονται οι παράμετροι που αναγνωρίστηκαν από την Athena. Η εφαρμογή αντλεί από το παραπάνω JSON την πληροφορία που χρειάζεται και παράγει το τελικό αποτέλεσμα στο χρήστη. Στη προκειμένη περίπτωση θα βρει στο Spotify το

κομμάτι “Someone like you” από την Adele και θα το αναπαράγει στα ηχεία της κινητής συσκευής.

Ένα πολύ χρήσιμο εργαλείο του Dialogflow, είναι η δυνατότητά του να εκπαιδεύεις το bot αφού οι χρήστες αρχίσουν να συνομιλούν μαζί του. Μπορείς με αυτό το τρόπο να διορθώσεις όποια ερώτηση δεν αναγνώρισε το bot ή συνέδεσε με λάθος απάντηση. Έτσι το bot βρίσκεται σε μια συνεχόμενη εκμάθηση κάνοντάς το ολοένα και πιο σωστό και έξυπνο. Το εργαλείο αυτό ονομάζεται Training και είναι στη διάθεση του προγραμματιστή μέσα από τη κονσόλα του Dialogflow.

Conversation	Requests	No match	Date	
play the meg	4	4	Jan 13	>
play a playlist by Adele	1	0	Today	>
play the meg	2	1	Today	>
play a playlist by Adele	1	0	Today	>
play Happy by Marina	42	1	Jan 11	>

Εικόνα 9. Training εργαλείο. Προεπισκόπηση των εισόδων του χρήστη ανά ημερομηνία.

play Happy by Marina

Jan 11 42 REQUESTS 1 NO MATCH APPROVE

PARAMETER NAME	ENTITY	RESOLVED VALUE
actions	@actions	play
MusicGenres	@MusicGenres	Happy

INTENT: playlist by genre

USER SAYS: play lies by Marina

INTENT: play specific track

USER SAYS: play Happy by Marina

INTENT: playlist by genre

USER SAYS: play lies by Marina

INTENT: play specific track

Εικόνα 10. Επεξεργασία κάποιου λανθασμένου Intent.

Επίσης μέσα από τη κονσόλα του Dialogflow μπορεί ο διαχειριστής της εφαρμογής να βλέπει τα στατιστικά χρήσης των διάφορων intent και το ποσοστό επιτυχίας τους. Επίσης του αναφέρει και το χρόνο απόκρισης του συστήματος στις διάφορες ερωτήσεις που κάνει ο χρήστης στο chatbot.

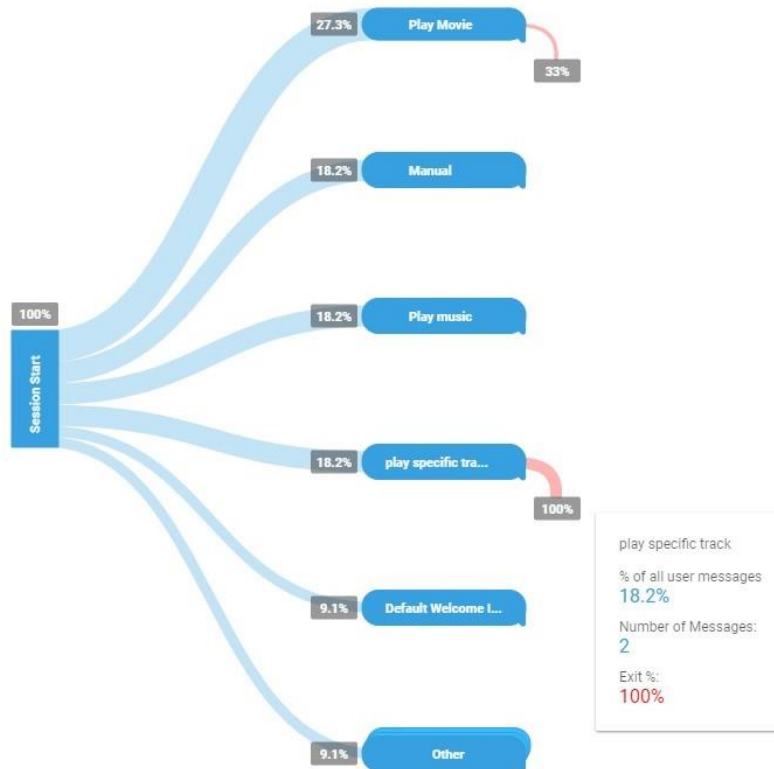


Figure 6. Χάρτης χρήσης του Dialogflow.

Spotify SDK και API

Για την αναπαραγωγή των μουσικών επιλογών του χρήστη η εφαρμογή χρησιμοποιεί το Spotify, το οποίο πρέπει να είναι εγκατεστημένο στη συσκευή του χρήστη. Το Spotify παρέχει τη δυνατότητα στους προγραμματιστές εφαρμογών να έχουν πρόσβαση στα δεδομένα του. Παρέχει μια τεράστια βιβλιοθήκη με τραγουδία και καλλιτέχνες, κάτι που είναι αρκετά χρήσιμο για μια εφαρμογή σαν και τη δική μας.

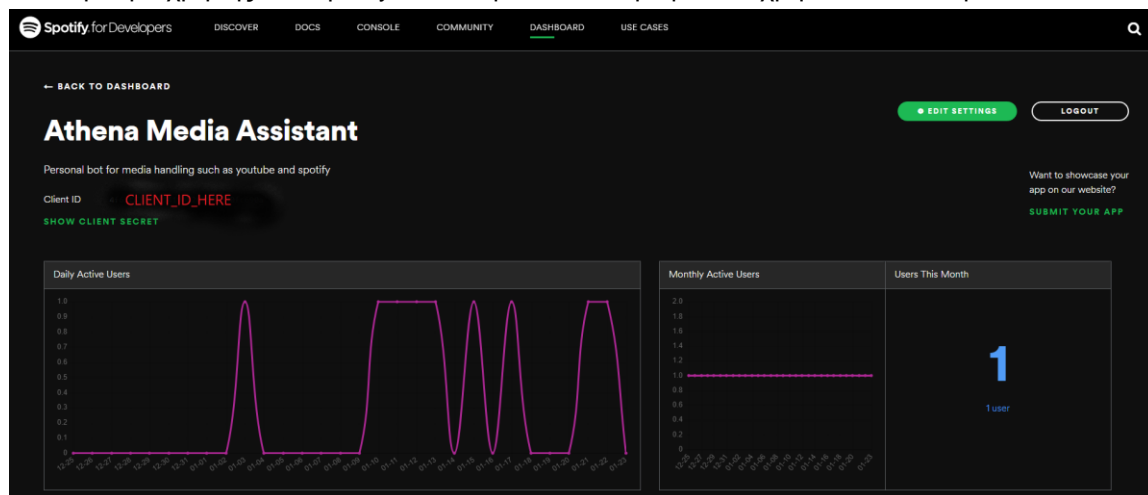
Αφού η Athena αναγνωρίσει το μουσικό κομμάτι που έχει ζητήσει ο χρήστης, το αναζητάει στη βιβλιοθήκη τραγουδιών του Spotify μέσω του Spotify API. Το API [13] για να λειτουργήσει χρειάζεται ένα κλειδί από το Spotify (Client-key). Για να αποκτήσουμε το κλειδί πρέπει να πάμε και να δηλώσουμε την εφαρμογή μας στην κονσόλα για προγραμματιστές του Spotify. Για την εγγραφή της εφαρμογής μας χρειάζεται να γνωρίζουμε το package-name της εφαρμογής μας και το δαχτυλικό της αποτύπωμα (SHA-1 key). Μόλις τα δηλώσουμε θα μας δοθεί το κλειδί για το SDK και το API του Spotify.

Το SDK περιέχει χρήσιμες έτοιμες συναρτήσεις για την αναπαραγωγή και τον έλεγχο της μουσικής που παίζει το Spotify. Σε πρώτο στάδιο η εφαρμογή μας κατά την εκκίνηση της κάνει την σύνδεση με την εφαρμογή του Spotify και αποστέλλει το κλειδί για να δοθεί η άδεια της χρήσης. Ο χρήστης στην οθόνη του θα δει μια μπάρα αναμονής σε μαύρο φόντο, ωστόσο γίνει η αυθεντικοποίηση. Μόλις γίνει η σύνδεση, ο χρήστης μπορεί πλέον να ζητήσει την αναπαραγωγή κάποιου τραγουδιού ή κάποιας λίστας.

Για την αναζήτηση στις μουσικές βιβλιοθήκες του Spotify, η εφαρμογή χρησιμοποιεί το API του Spotify. Το API το καλούμε μέσω της βιβλιοθήκης OkHttp. Η απάντηση έρχεται σε μορφή JSON και περιέχει μεγάλο όγκο από τραγουδία με βάση την αναζήτηση. Για την σωστή επιλογή του κομματιού που ζήτησε ο χρήστης χρησιμοποιούμε το όνομα του καλλιτέχνη που

μας έχει αναφέρει ο χρήστης. Με αυτό τον τρόπο ψάχνουμε να βρούμε μέσα από τη λίστα με τα πιθανά κομμάτια, το κομμάτι που συνδέεται πιο καλά με την επιθυμία του χρήστη. Για παράδειγμα όταν ο χρήστης πει “Play someone like you by Adele” τότε στη κλήση του API θα βάλουμε σαν όρο αναζήτησης το τίτλο του τραγουδιού δηλαδή το “Someone like you” και πως πρόκειται για τραγούδι και όχι για playlist, και στη συνέχεια θα ψάξουμε μέσα στο JSON που μας επιστράφηκε να βρούμε το τίτλο του κομματιού και από τον/την καλλιτέχνη με όνομα “Adele”. Τέλος αφού έχουμε βρει το σωστό κομμάτι, για την αναπαραγωγή του από τη συσκευή, παίρνουμε από το JSON το “URI” του κομματιού και το στέλνουμε στην κονσόλα για αναπαραγωγή. Παρόμοια τακτική ακολουθείται και στη περίπτωση που ο χρήστης ζητήσει κάποια playlist κάποιου καλλιτέχνη ή ακόμα και είδος μουσικής όπως για παράδειγμα “” Play acoustic music.”. Στη συγκεκριμένη περίπτωση θα αναζητήσει στο Spotify API για “acoustic music”, πως ψάχνουμε για playlists και θα επιλέξει την πρώτη λίστα που θα βρει και θα έχει ως είδος μουσικής το είδος που επιθυμεί ο χρήστης.

Από την διαδικτυακή κονσόλα για προγραμματιστές (Dashboard) που προσφέρει το Spotify, ο δημιουργός και διαχειριστής της εφαρμογής μπορεί να βλέπει τα στατιστικά χρήσης του Spotify που έχουν να κάνουν με την εφαρμογή. Μπορεί να βλέπει τους μηνιαίους χρήστες και το βαθμό χρήσης του Spotify API ακόμα και τον αριθμό των χρηστών ανά χώρα.



Εικόνα 11. Spotify κονσόλα διαχείρισης για προγραμματιστές (Dashboard).

YouTube SDK και API

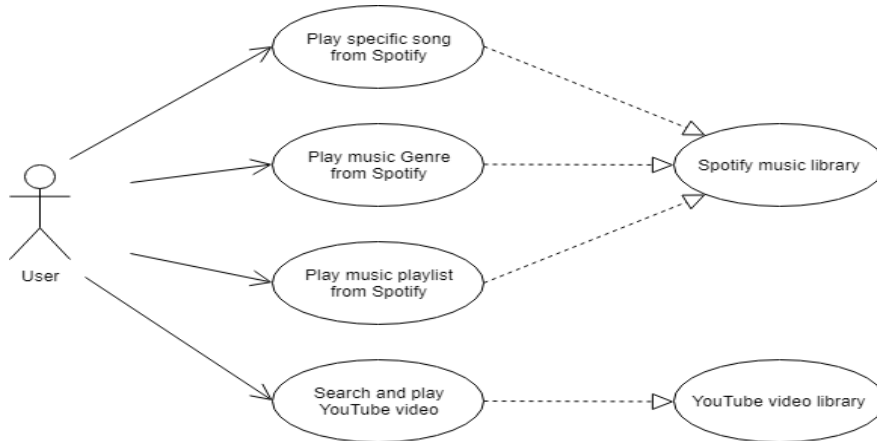
Για την αναπαραγωγή των βίντεο που ζητάει ο χρήστης από την Athena, η εφαρμογή χρησιμοποιεί το YouTube Android Player API [14] και το YouTube Data API [15]. Το πρώτο μας παρέχει τα απαραίτητα εξαρτήματα, όπως είναι το YouTube player, για την αναπαραγωγή των βίντεο. Ενώ το δεύτερο μας δίνει πρόσβαση και τρόπο αναζήτησης των βίντεο που ψάχνουμε, στο αστειρευτο περιεχόμενο από βίντεο που έχει το YouTube.

Πιο συγκεκριμένα, για την χρήση των δύο αυτών βασικών εργαλείων χρειάστηκε η ταυτοποίηση της εφαρμογής μας στην πλατφόρμα της Google. Αφού δηλώσουμε την εφαρμογή μας με παρόμοιο τρόπο όπως κάναμε και για το Spotify, με package name και SHA-1 key, ενεργοποιήσαμε το YouTube Data API για την εφαρμογή μας και μας δόθηκε το απαραίτητο κλειδί (Developers key) για τη χρήση του. Μόνο έτσι μπορούμε να πάρουμε πρόσβαση και να αξιοποιήσουμε τα δεδομένα του YouTube. Τέλος βάλαμε τις απαραίτητες, επίσημες βιβλιοθήκες στην εφαρμογή μας, για την ευκολότερη διαχείριση των YouTube δεδομένων.

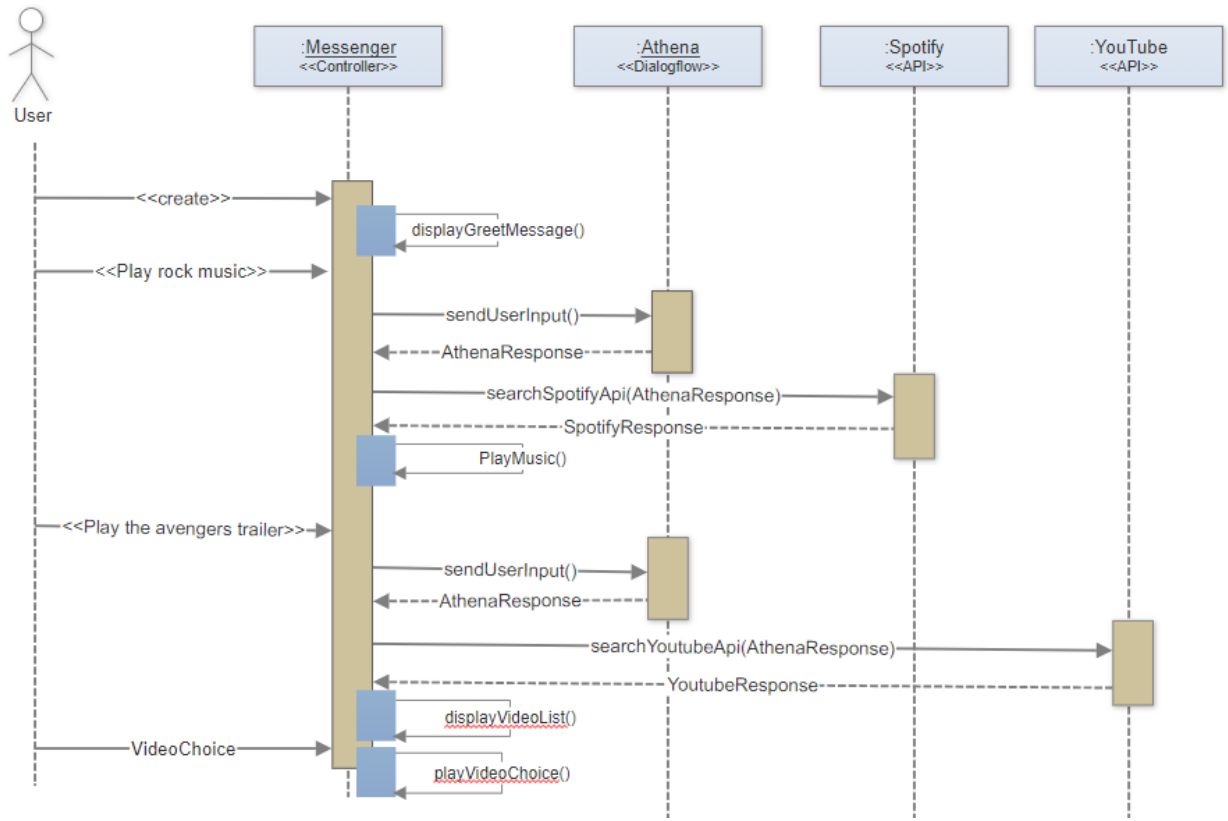
Το YouTube με αυτές τις δύο βιβλιοθήκες μας δίνει απεριόριστη πρόσβαση στα δεδομένα του. Μας δίνεται η δυνατότητα να αναζητήσουμε οποιοδήποτε βίντεο ακόμα και η πρόσβαση σε κανάλια (YouTube channels). Τα δεδομένα έρχονται, μετά από κλήση στο API, με τη μορφή JSON αντικειμένων. Μέσα από αυτά τα αντικείμενα παίρνουμε τις πληροφορίες για τα βίντεο που μας επιστράφηκαν, όπως είναι το όνομα του καναλιού που τα έχει ανεβάσει, την

εικόνα του βίντεο (Video Thumbnail), την κατηγορία στην οποία ανήκει και φυσικά το Id του βίντεο το οποίο και χρησιμοποιούμε για να τροφοδοτήσουμε τον player, ώστε να αρχίσει η αναπαραγωγή του βίντεο μέσω της εφαρμογής μας.

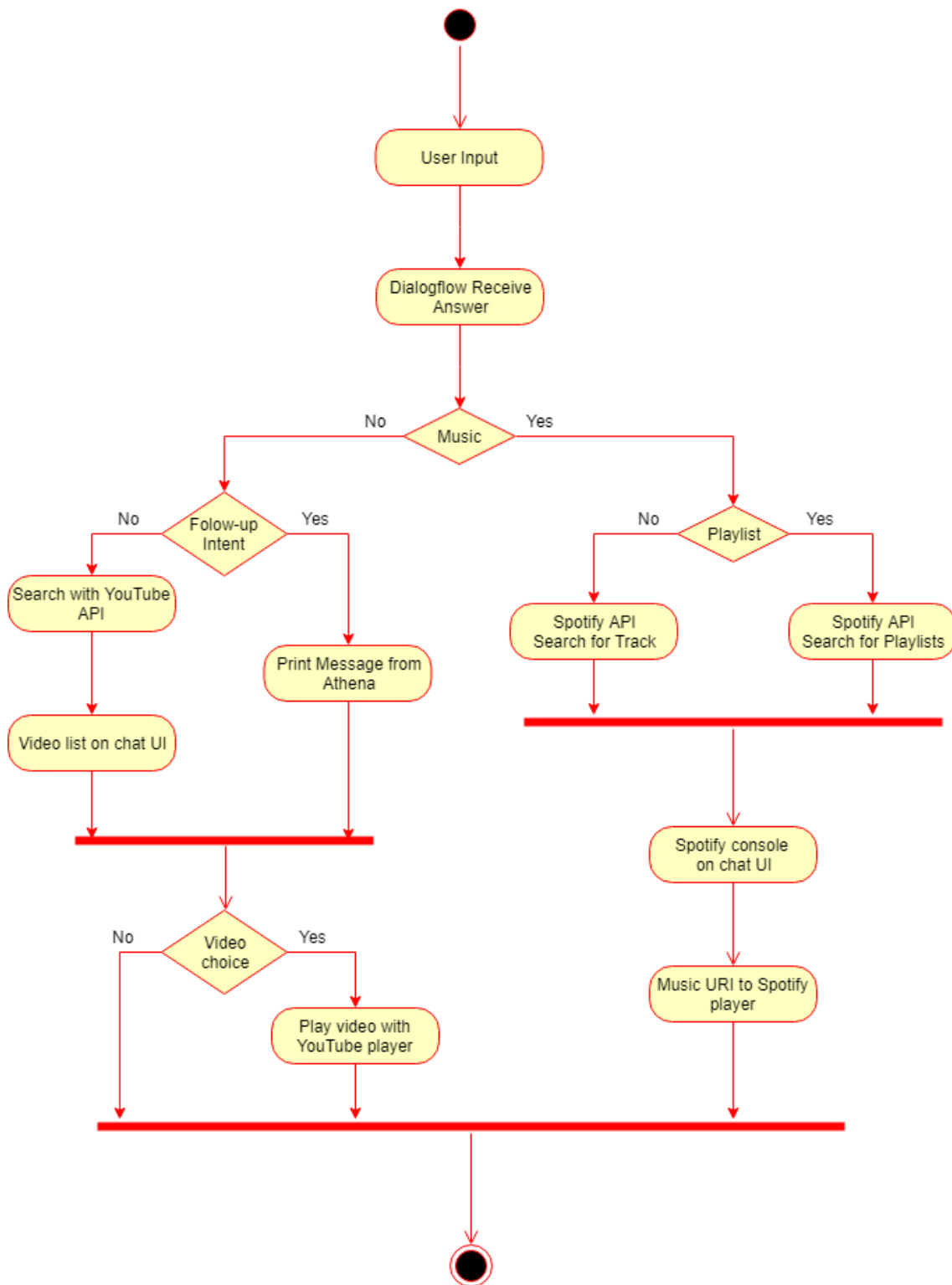
Διαγράμματα UML



Εικόνα 12. Διάγραμμα Χρήσης (Use Case).



Εικόνα 13. Sequence Diagram



Εικόνα 14. Διάγραμμα εργασιών για την αναπαραγωγή media.

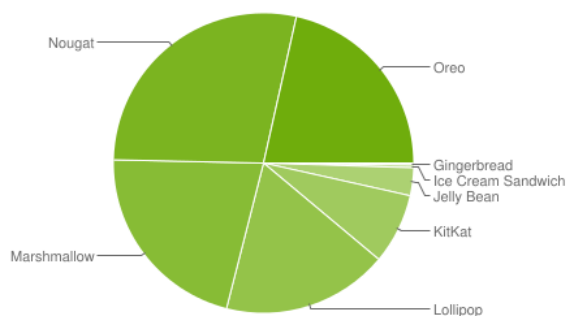
Android



Εικόνα 15. Λογότυπο της εφαρμογής, Athena.

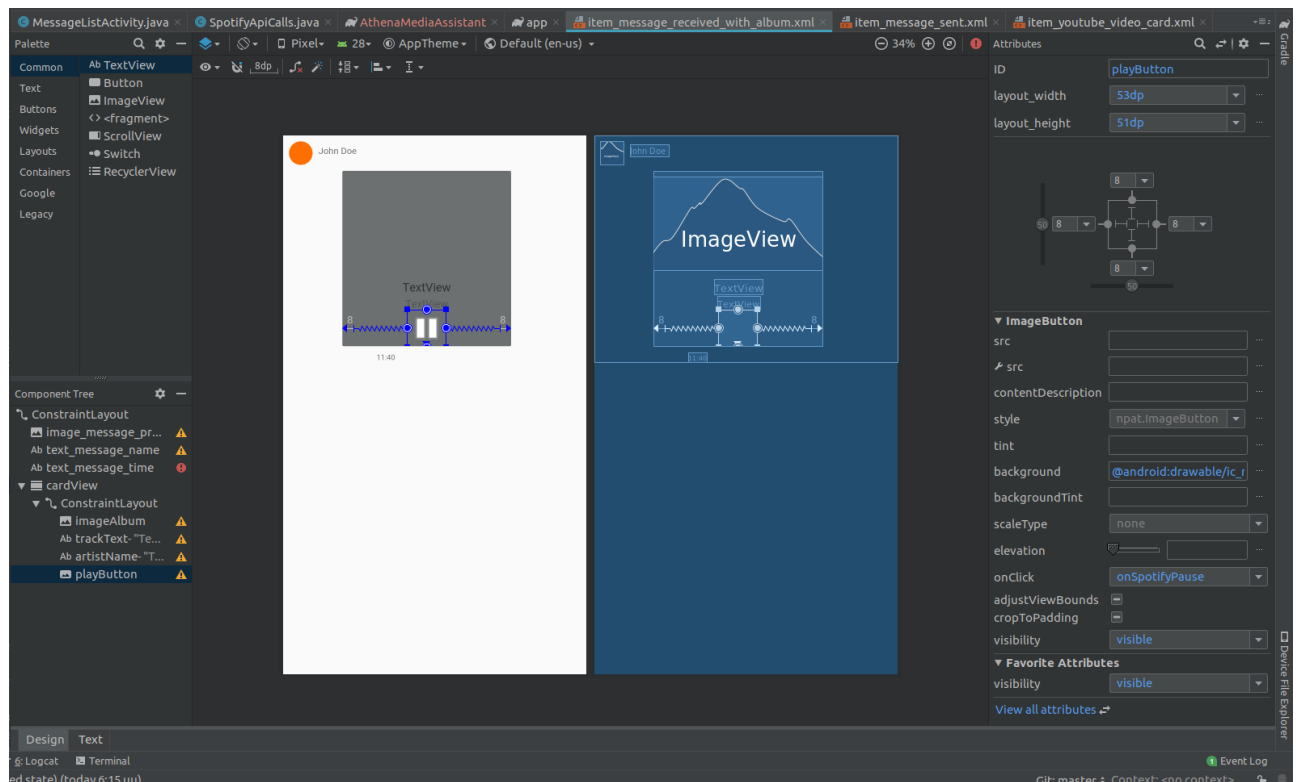
Η Athena με λογότυπό την εικόνα 15, είναι μια εφαρμογή για Android συσκευές με έκδοση συστήματος Android 6.0 δηλαδή Marshmallow και πάνω. Επέλεξα την έκδοση αυτή διότι αυτή και οι επόμενες κατέχουν αυτή τη στιγμή το μεγαλύτερο μερίδιο των Android συσκευών, εικόνα 16. Πρέπει να προσέχουμε πάντα την έκδοση του Android για μια εφαρμογή που θέλουμε να φτιάξουμε διότι παίζει καθοριστικό ρόλο για το πόσοι θα είναι οι εν δύναμη χρήστες της εφαρμογής μας.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.1%
4.2.x		17	1.5%
4.3		18	0.4%
4.4	KitKat	19	7.6%
5.0	Lollipop	21	3.5%
5.1		22	14.4%
6.0	Marshmallow	23	21.3%
7.0	Nougat	24	18.1%
7.1		25	10.1%
8.0	Oreo	26	14.0%
8.1		27	7.5%



Εικόνα 16. Ποσοστά Android εκδόσεων [22].

Για την δημιουργία της εφαρμογής χρησιμοποίησα το Android Studio IDE, ένα ελεύθερο και ολοκληρωμένο προγραμματιστικό περιβάλλον ανάπτυξης εφαρμογών για την πλατφόρμα του Android. Είναι βασισμένο στο IntelliJ IDEA ένα γνωστό περιβάλλον προγραμματισμού το οποίο όμως είναι με συνδρομή. Το Android Studio διευκολύνει πολύ την ανάπτυξη των εφαρμογών μας, προσφέροντας ένα γραφικό περιβάλλον, όπως φαίνεται στην εικόνα 17 για την ανάπτυξη του UI του χρήστη. Η δημιουργία του User Interface (UI) γίνεται πολύ εύκολη με τη χρήση έτοιμων εξαρτημάτων widgets τα οποία ο προγραμματιστής έχει την δυνατότητα είτε να τα γράψει με κώδικα είτε απευθείας με ένα drag & drop να τα προσθέσει στην οθόνη του κινητού και να τα προγραμματίσει. Κάθε εξάρτημα έχει την ικανότητα να τροποποιηθεί να αλλάξουν οι διαστάσεις του, το χρώμα του, η θέση του στην οθόνη του χρήστη και άλλες πολλές δυνατότητες.



Εικόνα 17. Εργαλείο του Android Studio για την δημιουργία του περιβάλλοντος του χρήστη.

Στην συνέχεια, όπως προ είπαμε χρησιμοποιήθηκαν οι βιβλιοθήκες για Spotify και YouTube οι οποίες ορίστηκαν στο Gradle της εφαρμογής. Το Gradle είναι ένα εργαλείο για το χτίσιμο (build) του προγράμματός μας. Σε αυτό δηλώνουμε τις βιβλιοθήκες που χρησιμοποιήσαμε, δηλαδή το YouTube, Spotify και το Dialogflow.

Όλες οι εφαρμογές περιέχουν ένα αρχείο το οποίο ονομάζουν AndroidManifest.xml. Σε αυτό το αρχείο δηλώνονται τα στοιχεία της εφαρμογής μας όπως το όνομα τα δικαιώματα (permissions), το λογότυπο της εφαρμογής και τα διάφορα Activities της εφαρμογής. Ένα Activity είναι μία συγκεκριμένη ενέργεια που μπορεί να κάνει ένας χρήστης τις εφαρμογής, δηλαδή το Activity δημιουργεί κάποιο UI για να συναναστραφεί με τον χρήστη. Η εφαρμογή Athena αποτελείται από ένα Activity, διότι πρόκειται για ένα UI πάνω στο οποίο απεικονίζονται τα διάφορα μηνύματα του χρήστη και της Athena.

Η συγκεκριμένη εφαρμογή για να μπορέσει να επικοινωνήσει με το chatbot Athena, το οποίο είναι ουσιαστικά μια Cloud υπηρεσία της Google, για να ψάξει στο Spotify και στο YouTube χρειαζόμαστε τη χρήση του διαδικτύου. Επίσης είναι απαραίτητη η ηχογράφηση της φωνής του χρήστη για να μπορεί να μεταφέρει φωνητικά το ερώτημα στην Athena. Όλα τα παραπάνω για να μπορέσουν να λειτουργήσουν πρέπει πρώτα να δοθούν τα κατάλληλα δικαιώματα για τη χρήση τους. Τα δικαιώματα αυτά ο προγραμματιστής τα δηλώνει στο αρχείο Δημιουργία Προσωπικού Ψηφιακού Βοηθού για κινητές συσκευές

του Manifest, ώστε η εφαρμογή με την εγκατάστασή της να τα ζητήσει από το χρήστη. Στις νεότερες εκδόσεις του Android για να μπορέσουμε να πάρουμε τα δικαιώματα αυτά πρέπει να τα ζητήσουμε από τον χρήστη μέσα από την εφαρμογή. Δηλαδή αφού ο χρήστης ανοίξει για πρώτη φορά την εφαρμογή, τότε στην οθόνη του χρήστη θα εμφανιστή ένα μήνυμα που θα τον ενημερώνει και θα του ζητάει να πάρει την απόφαση για τον αν θα επιτρέψει στην εφαρμογή να πάρει τα συγκεκριμένα δικαιώματα. Αν ο χρήστης δεχτεί τότε η εφαρμογή κρατάει την απάντησή του και προχωρά με τη λειτουργία της. Αν ο χρήστης αρνηθεί τότε η εφαρμογή τον ενημερώνει πως δεν μπορεί να λειτουργήσει σωστά χωρίς αυτά και σταματά.

Κώδικας του Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.uniwa.cse.athenamediaassistant">

    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET"/> <!--
dikaivmata gia ti xrisi toy diktyoy-->
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
<!-- dikaivmata gia tin ixografisi ixoy-->
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
<!-- dikaivmata gia tin prosvasi se logariasmous-->

    <application
        android:allowBackup="true"
        android:icon="@drawable/athena_logo"
        android:label="@string/app_name"
        android:roundIcon="@drawable/athena_logo"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:launchMode="singleInstance"
        tools:ignore="GoogleAppIndexingWarning">

        <activity android:name=".MessageListActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>

        </activity>

    </application>

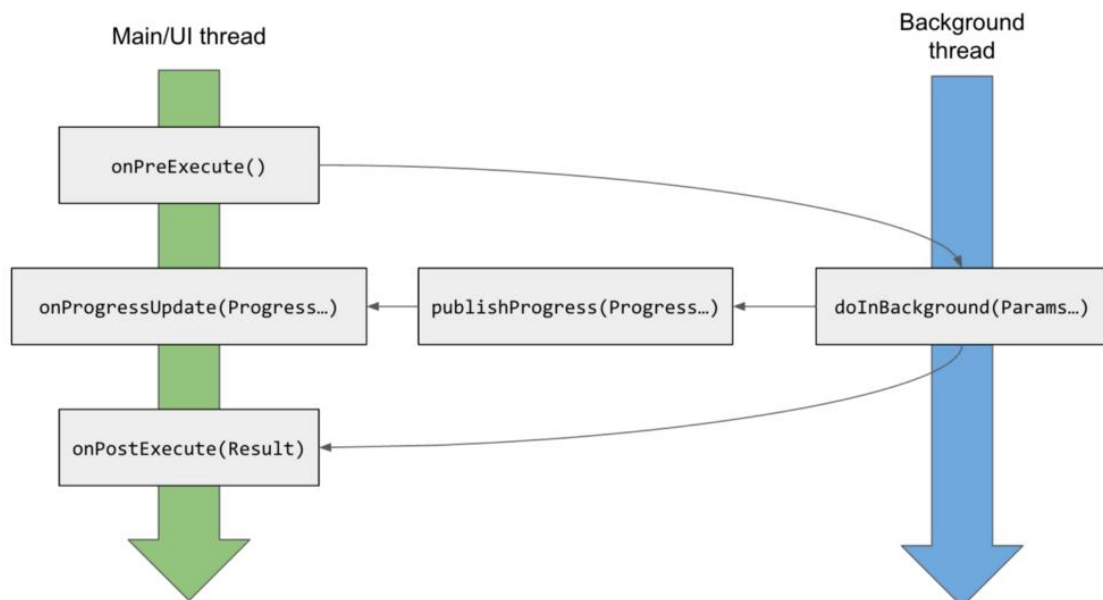
</manifest>
```

Αφού ο χρήστης δώσει τα δικαιώματα για την χρήση των παραπάνω, τότε η εφαρμογή ξεκινά τη σύνδεση με το Spotify. Ο χρήστης πρέπει να περιμένει να πραγματοποιηθεί η σύνδεση αυτή για να μπορέσει να αναζητήσει τραγούδια. Αν η σύνδεση είναι επιτυχής ο χρήστης μπορεί να στείλει μηνύματα στην Athena. Όμως αν η σύνδεση δε πραγματοποιήθηκε, τότε όταν ο χρήστης πατήσει το κουμπί Send για να στείλει κάποιο μήνυμα τότε θα προσπαθήσει να συνδεθεί ξανά. Με αυτό το τρόπο η εφαρμογή δεν κολλάει, γιατί αν ο χρήστης έστειλε μήνυμα χωρίς να έχει γίνει η σωστή σύνδεση με το Spotify τότε η εφαρμογή θα κολλούσε με αποτέλεσμα το λειτουργικό να την έκλεινε θεωρώντας την επιβλαβή για το σύστημα.

Ο χρήστη την επιλογή να γράψει η στείλει φωνητικά το μήνυμά του. Αυτό γίνεται με το κουμπί Send. Όπως έχουμε πει και σε προηγούμενο κεφάλαιο πατώντας το κουμπί το χρήστης έχει γράψει κάτι στο πεδίο εισαγωγής μηνυμάτων τότε στέλνετε το μήνυμα γραπτώς. Όμως αν πατώντας το κουμπί ο χρήστης δεν έχει γράψει τίποτα στο πεδίο εισαγωγής μηνυμάτων, τότε ενεργοποιείται η ηχογράφηση φωνής. Αυτό έγινε εύκολα με έναν έλεγχο στο περιεχόμενο του UI widget EditText που βρίσκετε στο κάτω μέρος της οθόνης του χρήστη.

Ασύγχρονες κλήσεις της εφαρμογής, Spotify API

Στο Android για να καταφέρει κάποιος κάνει κάποια κλήση ενός API θα πρέπει να χρησιμοποιήσει μια ασύγχρονη διεργασία [17]. Η αποστολή και επικοινωνία με την Athena γίνεται με την χρήση μιας AsyncTask, μιας ασύγχρονης διεργασίας που εκτελείται από ένα άλλο νήμα εκτός του βασικού νήματος (Main Thread / UI Thread). Αν δεν χρησιμοποιούσαμε ένα δεύτερο νήμα για την εκτέλεση της επικοινωνίας τότε η εφαρμογή μας θα κόλλαγε μέχρις ότου η επικοινωνία ολοκληρωνόταν. Αυτό συμβαίνει διότι στο Android όλες οι ενέργειες εκτελούνται από το βασικό νήμα, γι' αυτό το λόγο το βασικό νήμα ονομάζεται και UI Thread. Οπότε αν εκτελούσαμε την επικοινωνία πάνω του, ο χρήστης δε θα μπορούσε να κάνει τίποτα και η οθόνη του δε θα ανταποκρινόταν, θα την έβλεπε παγωμένη και μη ανταπόκριση. Θέλει ιδιαίτερη προσοχή όταν κατασκευάζουμε μια εφαρμογή για Android, η εφαρμογή μας να μην φαίνεται κολλημένη.



Εικόνα 18. Σχήμα λειτουργιών της AsyncTask [17].

Μια AsyncTask έχει τρεις βασικές λειτουργίες, την onPreExecute(), την doInBackground(Params) και την onPostExecute(Result). Η onPreExecute εκτελείται λίγο πριν ξεκινήσει το νέο νήμα οπότε εδώ πραγματοποιούμε πιθανές αρχικοποιήσεις στις μεταβλητές μας. Η doInBackground καλείται αμέσως αφού το νήμα ξεκινήσει και σε αυτή πραγματοποιείται

η δουλειά που έχει ανατεθεί στο νήμα για να γίνει στο παρασκήνιο. Η `onPostExecute` εκτελείται στο βασικό νήμα και ουσιαστικά επιστρέφει το αποτέλεσμα της `doInBackground` για να το αξιοποιήσουμε. Στη εικόνα 18, βλέπουμε ένα σχήμα του πως λειτουργεί μια ασύγχρονη διεργασία του τύπου `AsyncTask`.

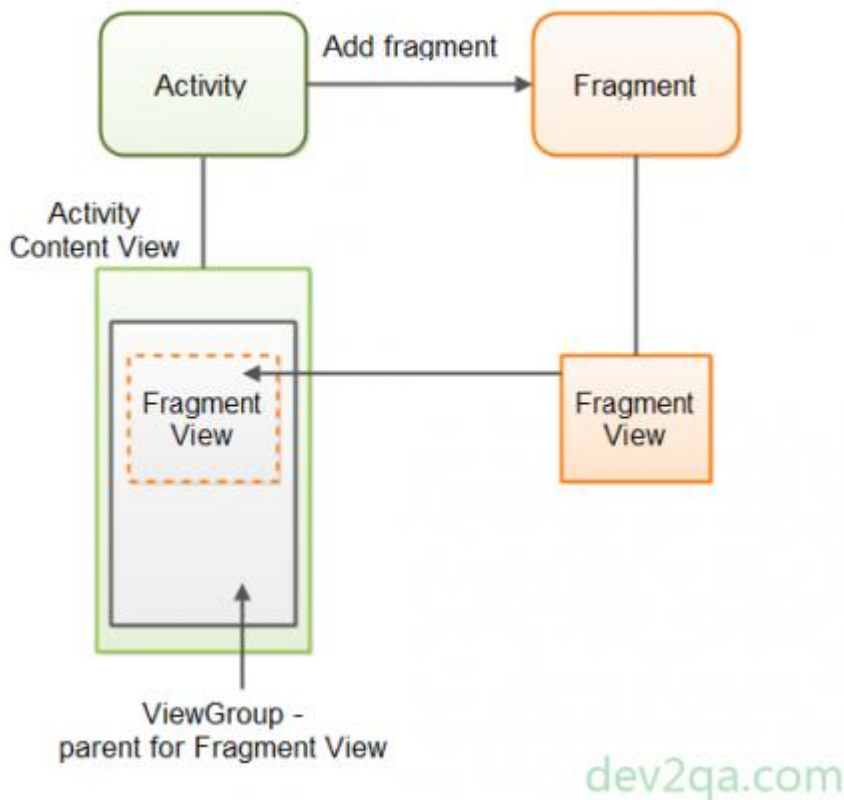
Η εφαρμογή μας λοιπόν όπως είπαμε χρησιμοποιεί την `AsyncTask` για να επικοινωνήσει με την `Athena`, με το `Spotify` και το `YouTube`. Όταν ο χρήστης κάνει κάποιο ερώτημα τότε η εφαρμογή καλεί στην `doInBackground` την κλήση για την αποστολή του ερωτήματος στο `Dialogflow`, δηλαδή στην `Athena` από την οποία θα πάρουμε κάποια απάντηση. Η απάντηση αυτή περνάει στο βασικό νήμα μέσω της `onPostExecute`, ώστε να πάρουμε τη πληροφορία που μας έστειλε η `Athena` και να πράξουμε αναλόγως. Αν η `Athena` κατάλαβε πως πρόκειται για κάποιο μουσικό κομμάτι τότε η πληροφορία αυτή, τίτλος κομματιού και καλλιτέχνη για παράδειγμα, στέλνεται μέσω ενός άλλου νήματος στο `Spotify` με χρήση του API. Ακολουθώντας την ίδια διαδικασία η απάντηση του `Spotify` περνάει στο βασικό νήμα το οποίο χρησιμοποιεί την πληροφορία που έλαβε για να σχεδιάσει κατάλληλα το UI, βάζοντας την κονσόλα διαχείρισης του `Spotify` στην οθόνη του χρήστη με όνομα του τραγουδιού που παίζει, του καλλιτέχνη καθώς και τη φωτογραφία του άλμπουμ.

Την φωτογραφία του άλμπουμ την κατεβάζει από το url που έλαβε από το `Spotify`, με χρήση της βιβλιοθήκης `Glide`. Η βιβλιοθήκη αυτή βασίζεται στη λογική της `AsyncTask`. Δημιουργεί ένα νήμα στο `background` της εφαρμογής, κατεβάζει την εικόνα που του ζητάμε και ως αποτέλεσμα την εμφανίζει σε ένα widget `ImageView` που του έχουμε ορίσει εμείς κατά τη δημιουργία του UI, εικόνα 17.

YouTube API

Για το `YouTube` ακολουθήθηκε παρόμοια προσέγγιση με χρήση `AsyncTask` για την εκπόνηση των διάφορων ενεργειών. Πιο συγκεκριμένα στην εφαρμογή χρησιμοποιούμε τη κλάση `MakeRequestTask` η οποία κάνει extent την `AsyncTask`. Μέσα σε αυτή εκτελείται η `doInBackground` η οποία κάνοντας χρήση του `YouTube SDK` κάνει κλήση στο API ψάχνοντας το βίντεο που ζήτησε ο χρήστης από την `Athena`. Στη συνέχεια, στην `onPostExecute` περνάμε την πληροφορία που επιστράφηκε στο UI Thread και απεικονίζουμε τα 5 πρώτα βίντεο στην οθόνη του χρήστη, ώστε να επιλέξει αυτός πιο θέλει να δει.

Μόλις ο χρήστης επιλέξει το βίντεο τότε θα εμφανιστεί ένα `Video Fragment` στην οθόνη του, το οποίο θα παίξει το βίντεο. Αυτό γίνεται διότι το `Youtube` δεν επιτρέπει κάποιο `overlay` να υπάρχει πάνω από τον `Youtube player`. Αν δε βάζαμε δηλαδή το `Fragment`, όταν ο χρήστης ανεβοκατέβαινε στην οθόνη των μηνυμάτων το βίντεο θα σταμάταγε και το `Youtube` θα κολλούσε και πιθανόν να έκλεινε και η εφαρμογή. Αυτό μπορούμε να το δούμε εύκολα να συμβαίνει στην επίσημη έκδοση του `YouTube` για `Android` συσκευές. Όταν επιχειρήσουμε να κλειδώσουμε τη συσκευή ή βάλουμε κάποιο άλλο UI πάνω από την εφαρμογή του `YouTube`, το βίντεο σταματάει την αναπαραγωγή του. Αυτό συμβαίνει ακόμα και όταν μια εφαρμογή μηνυμάτων ανοίξει με την αποστολή κάποιου μηνύματος. Για τον παραπάνω λόγο είναι απαραίτητη η χρήση ενός `Fragment`, για την καλύτερη και ομαλότερη χρήση της εφαρμογής.



Εικόνα 19. Σχεδιάγραμμα ενός Fragment [18].

Ο έλεγχος του Video Fragment γίνεται μέσα στην συνάρτηση `showVideo(videoEntry)` όπως φαίνεται και στον κώδικα από κάτω. Αρχικοποιούμε το fragment με αυτό που έχουμε δημιουργήσει στο UI και στη συνέχεια του περνάμε το Id αριθμό του βίντεο που επέλεξε ο χρήστης για να δει. Στη συνέχεια ελέγχουμε αν το `videoFragment` είναι ορατό στην οθόνη του χρήστη. Αν δεν είναι το εμφανίζουμε στην οθόνη πάνω από το βασικό UI, δηλαδή πάνω από τα μηνύματα. Το κλείσιμο του βίντεο γίνεται με το κουμπί "X" πάνω από το `videoFragment`. Μόλις πατηθεί το κουμπί αυτό, σταματάμε την αναπαραγωγή του βίντεο και εξαφανίζουμε το fragment από την οθόνη του χρήστη μέσω ενός μικρού animation.

Κώδικας διαχείρισης Video Fragment.

```
private void showVideo(VideoEntry videoEntry){

    videoFragment =
        (VideoFragment)
        getFragmentManager().findFragmentById(R.id.video_fragment_container);

    videoFragment.setVideoId(videoEntry.getVideoId());

    // The videoBox is INVISIBLE if no video was previously selected, so
    // we need to show it now.
    if (videoBox.getVisibility() != VISIBLE) {
        if (getResources().getConfiguration().orientation ==
            Configuration.ORIENTATION_PORTRAIT) {
            // Initially translate off the screen so that it can be
```

```

animated in from below.
        videoBox.setTranslationY(videoBox.getHeight());
    }
    videoBox.setVisibility(VISIBLE);

}

}

public void onClickClose(View view) {
    System.out.println("close");

    videoFragment.pause();
    // Edw to animation gia to kleisimo toy video apo tin othoni
    ViewPropertyAnimator animator = videoBox.animate()
        .translationYBy(videoBox.getHeight())
        .setDuration(ANIMATION_DURATION_MILLIS);
    runOnAnimationEnd(animator, new Runnable() {
        @Override
        public void run() {
            videoBox.setVisibility(View.INVISIBLE);
        }
    });
}

private void runOnAnimationEnd(ViewPropertyAnimator animator, final
Runnable runnable) {

    animator.addListener(new AnimatorListenerAdapter() {
        @Override
        public void onAnimationEnd(Animator animation) {
            runnable.run();
        }
    });
}
}

```

User Interface

Ο κώδικας για το video fragment φαίνεται παρακάτω. Πρόκειται για ένα εμφωλευμένο LinearLayout στο βασικό View της κύριας Activity που ονομάζεται MessageListActivity. Η βασική αυτή Activity αποτελείται από ένα RecyclerView όπως βλέπουμε και στην εικόνα 20. Το RecyclerView [19] είναι η πιο εξελιγμένη έκδοση του γνωστού ListView. Πρόκειται για ένα UI widget το οποίο είναι κατάλληλο για την απεικόνιση μεγάλου όγκου πληροφορίας ενώ μπορεί να αλλάζει συνεχώς και με μεγαλύτερη άνεση στις αλλαγές των δεδομένων που απεικονίζονται απ' ότι το ListView. Επίσης επιτρέπει τη χρήση animations κάτι που παλαιότερα ήταν πολύ δύσκολο να επιτευχθεί με το ListView.

Για να περάσουμε αντικείμενα μέσα στο RecyclerView όπως και με το ListView χρειαζόμαστε έναν RecyclerView Adapter. Ο Adapter αυτός θα παίρνει κάποια ορίσματα όπως είναι ένα αντικείμενο με πληροφορίες για το μήνυμα που πρέπει να εμφανίσει κάθε φορά, από ποιόν είναι το μήνυμα, το χρόνο αποστολής του μηνύματος, αν υπάρχει κάποια εικόνα όπως είναι για τα άλμπουμ των τραγουδιών και άλλα. Ο Adapter ξεχωρίζει το ποιος στέλνει το μήνυμα

από έναν αριθμό που λαμβάνει από το 1 μέχρι το 4. Το 1 για μηνύματα που στέλνει ο χρήστης προς την Athena, τα οποία εμφανίζονται στα δεξιά της οθόνης με πράσινο χρώμα. Το 2 για τα μηνύματα που στέλνει η Athena προς τον χρήστη, τα οποία εμφανίζονται στα αριστερά της οθόνης με πορτοκαλί χρώμα. Το 3 αφορά την απεικόνιση μουσικών κομματιών από το Spotify, δηλαδή η κονσόλα διαχείρισης του Spotify. Το 4 αφορά τα βίντεο από το YouTube. Παρακάτω βλέπουμε το κώδικα στον Adapter που διαχωρίζει τα παραπάνω στα ανάλογα Views.

Κώδικας διαχωρισμού των Views στον Adapter

```
@Override
public int getItemViewType(int position) {
    UserMessage message = (UserMessage) mMessageList.get(position);

    if (message.getSender().getUserId() == 1) {
        // If the current user is the sender of the message
        return VIEW_TYPE_MESSAGE_SENT;
    } else if (message.getSender().getUserId() == 2){
        // If some other user sent the message
        return VIEW_TYPE_MESSAGE_RECEIVED;
    }else if (message.getSender().getUserId() == 3){
        return VIEW_TYPE_MESSAGE_ALBUM;
    }else{
        return VIEW_TYPE_YOUTUBE;
    }
}

// Inflates the appropriate layout according to the viewType.
@Override
public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
    View view;

    if (viewType == VIEW_TYPE_MESSAGE_SENT) {
        view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_message_sent, parent, false);
        return new SentMessageHolder(view); // epistrofi view gia minima
apo xristi
    } else if (viewType == VIEW_TYPE_MESSAGE_RECEIVED) {
        view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_message_received, parent, false);
        return new ReceivedMessageHolder(view); // epistrofi view gia
minima apo Athena
    } else if (viewType == VIEW_TYPE_MESSAGE_ALBUM) {
        view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_message_received_with_album,
parent, false);
        return new AlbumMessageHolder(view); // epistrofi view gia minima
Spotify
    } else if (viewType == VIEW_TYPE_YOUTUBE) {
        view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_youtube_video_card, parent,
false);
        return new YoutubeMessageHolder(view); // epistrofi view gia
```

```

minima Youtube
    }

    return null;
}

// Passes the message object to a ViewHolder so that the contents can be
// bound to UI.
@Override
public void onBindViewHolder(RecyclerView.ViewHolder holder, int
position) {
    UserMessage message = (UserMessage) mMessageList.get(position);

    switch (holder.getItemViewType()) {
        case VIEW_TYPE_MESSAGE_SENT:
            ((SentMessageHolder) holder).bind(message);
            break;
        case VIEW_TYPE_MESSAGE_RECEIVED:
            ((ReceivedMessageHolder) holder).bind(message);
            break;
        case VIEW_TYPE_MESSAGE_ALBUM:
            ((AlbumMessageHolder) holder).bind(message);
            break;
        case VIEW_TYPE_YOUTUBE:
            ((YoutubeMessageHolder) holder).bind(message);
            break;
    }
}
}

```

Πρώτα και αφού γίνει ο διαχωρισμός για το ποιο View πρέπει να δημιουργηθεί στην getItemViewType(), προχωράμε στην δημιουργία του ViewHolder στην onCreateViewHolder. Η onBindViewHolder βοηθάει στο δέσιμο του αντικειμένου του μηνύματος στο κατάλληλο View, ώστε να μπορέσει να πάρει την απαραίτητη πληροφορία που πρέπει να απεικονίσει το κάθε διαφορετικό View.

Ο xml κώδικας του Video Fragment.

```

<LinearLayout
    android:id="@+id/video_box"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_gravity="top"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="256dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toTopOf="@+id/view"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/recyclerview_message_list"
    app:layout_constraintTop_toTopOf="@id/constraintId"

```



```

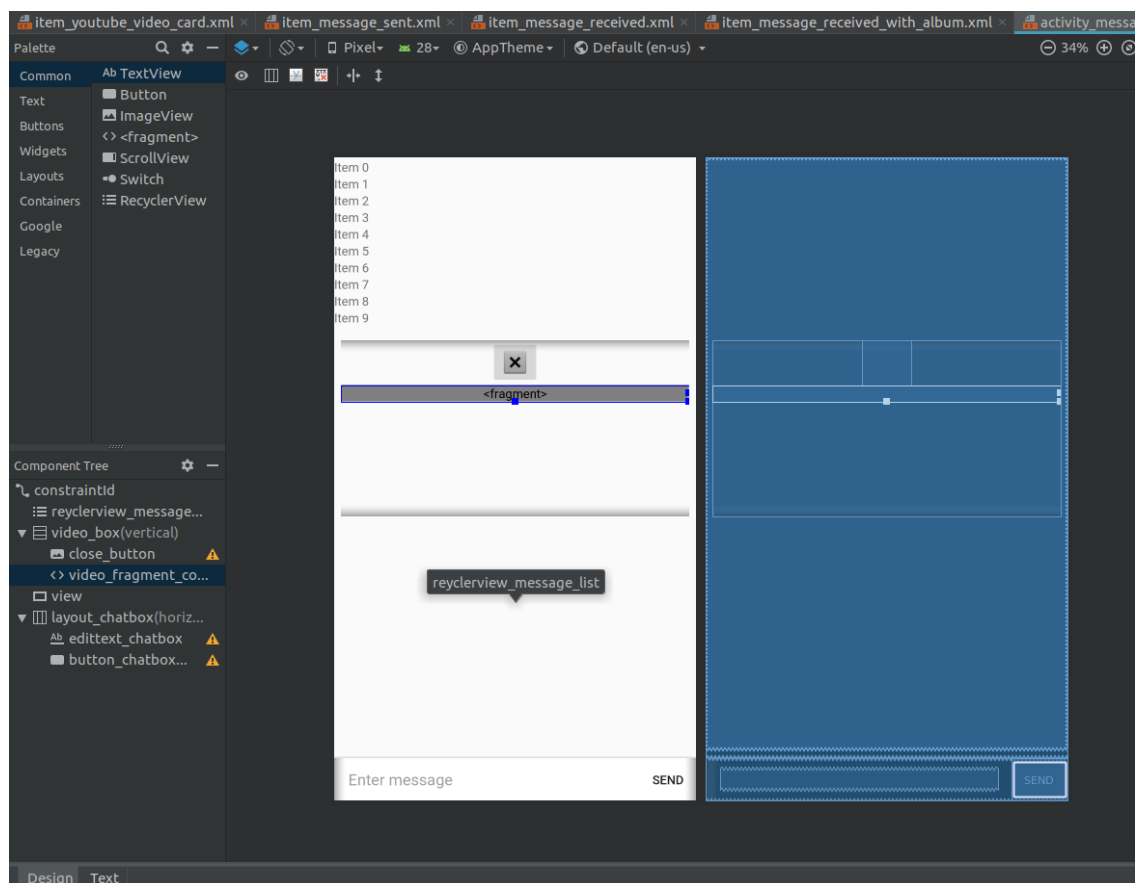
app:layout_constraintVertical_bias="0.918">

<ImageButton
    android:id="@+id/close_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:onClick="onClickClose"
    android:src="@android:drawable/btn_dialog" />

<fragment
    android:id="@+id/video_fragment_container"
    class="com.uniwa.cse.athenamediaassistant.MessageListActivity$VideoFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

</LinearLayout>

```



Εικόνα 20. Περιβάλλον χρήστη (UI) με το Video Fragment.

Κάθε φορά που ο χρήστης βλέπει κάποιο βίντεο, έχει τη δυνατότητα να το παρακολουθήσει σε λειτουργία πλήρους οθόνης. Αυτό το καταφέρνει πατώντας από τον Player του YouTube κουμπί της πλήρους οθόνης που βρίσκεται κάτω δεξιά. Τότε η οθόνη θα γυρίσει

στο πλάι μεγαλώνοντας το μέγεθος του player, κάνοντας την εμπειρία πιο απολαυστική και ξεκούραστη για το χρήστη. Στον κώδικα για την πλήρη οθόνη που βρίσκεται παρακάτω, βλέπουμε τον κώδικα που είναι υπεύθυνος για την αλλαγή στα μεγέθη της οθόνης της συσκευής ώστε να μπαίνει σωστά η πλήρης οθόνη.

Κώδικας για την πλήρη οθόνη.

```
private void layout() {
    boolean isPortrait =
        getResources().getConfiguration().orientation ==
        Configuration.ORIENTATION_PORTRAIT;

    if (isFullscreen) {
        // Reset any translation that was applied in portrait.

        closeButton.setVisibility(GONE);
        videoBox.setTranslationY(0);
        setLayoutSize(videoFragment.getView(), MATCH_PARENT,
otherViews.getHeight());
        setLayoutSizeAndGravity(videoBox, MATCH_PARENT,
otherViews.getHeight(), Gravity.BOTTOM | Gravity.START);
    } else if (isPortrait) {
        closeButton.setVisibility(VISIBLE);
        setLayoutSize(videoFragment.getView(), MATCH_PARENT,
WRAP_CONTENT);
        setLayoutSizeAndGravity(videoBox, MATCH_PARENT, WRAP_CONTENT,
Gravity.BOTTOM);
    } else {
        videoBox.setTranslationY(0); // Reset any translation that was
applied in portrait.
        int screenWidth =
dpToPx(getResources().getConfiguration().screenWidthDp);
        int videoWidth = screenWidth - screenWidth/4 -
dpToPx(LANDSCAPE_VIDEO_PADDING_DP);
        setLayoutSize(videoFragment.getView(), videoWidth, WRAP_CONTENT);
        setLayoutSizeAndGravity(videoBox, videoWidth, WRAP_CONTENT,
Gravity.BOTTOM | Gravity.CENTER_VERTICAL);
    }
}

private void setLayoutSize(View view, int width, int height) {
    ViewGroup.LayoutParams params = view.getLayoutParams();
    params.width = width;
    params.height = height;
    view.setLayoutParams(params);
}

private void setLayoutSizeAndGravity(View view, int width, int height,
int gravity) {
    ViewGroup.LayoutParams params = view.getLayoutParams();
    params.width = width;
    params.height = height;
```

```

        view.setLayoutParams(params);
    }

    private int dpToPx(int dp) {
        return (int) (dp * getResources().getDisplayMetrics().density +
0.5f);
    }

    @Override
    public void onConfigurationChanged(Configuration newConfig) {
        super.onConfigurationChanged(newConfig);

        layout();
    }

    @Override
    public void onFullscreen(boolean isFullscreen) {
        this.isFullscreen = isFullscreen;

        layout();
    }
}

```

Όταν υπάρξει μια αλλαγή στον προσανατολισμό της οθόνης, όταν δηλαδή πατηθεί το κουμπί της πλήρους οθόνης τότε ο listener για τη πλήρη οθόνη (onFullscreen(isFullscreen)) θα καλέσει την layout(). Στην αρχή θα γίνει ο έλεγχος για το αν περνάμε σε πλήρη οθόνη. Αν ναι, τότε αλλάζουμε τις διαστάσεις του layout (UI) με τη βοήθεια των συναρτήσεων setSize(), setSizeAndGravity() και τη dpToPx() ώστε να γεμίζουν την οθόνη του χρήστη, έτσι το βίντεο παίζει σε πλήρη οθόνη. Η setSize() αλλάζει το πλάτος και το ύψος του UI σε αναλογία πλήρους οθόνης. Η dpToPx() μετράει το μήκος και πλάτος του layout ώστε να γίνουν οι ανάλογες μετατροπές από πλήρη οθόνη σε πορτρέτο και ανάποδα.

Η εφαρμογή παίζει μόνο σε προσανατολισμό πορτρέτου. Δηλαδή όταν ο χρήστης γυρίσει τη συσκευή του στο πλάι, δεν θα αλλάξει ο προσανατολισμός του UI. Αυτό το έγινε συνειδητά για την καλύτερη και ευκολότερη χρήση του UI της εφαρμογής. Αυτό το ρυθμίσαμε στο αρχείο του Manifest.xml, που έχουμε δείξει πιο πάνω. Πιο συγκεκριμένα στο σημείο που ορίσαμε το Activity της εφαρμογής βάλαμε το screenOrientation = portrait :

```

<activity android:name=".MessageListActivity"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:screenOrientation="portrait">

```

Συμπεράσματα και μελλοντικές επεκτάσεις

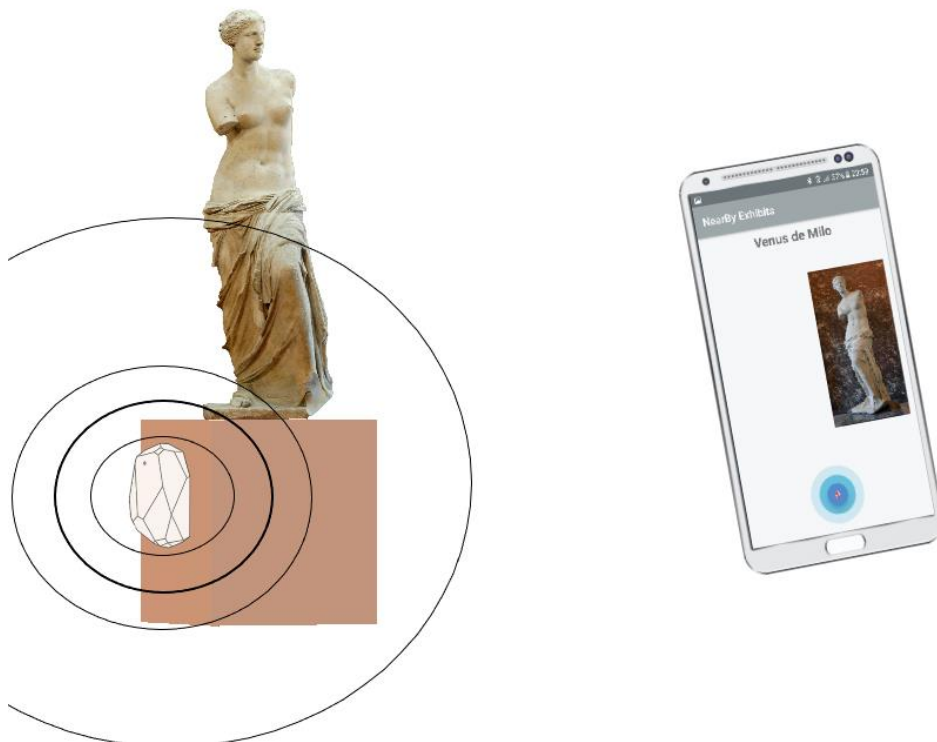
Η εφαρμογή αυτή έχει ακόμα πολύ χώρο για επέκταση και βελτίωση, λόγω των δυνατοτήτων ενός ψηφιακού βοηθού. Έχει την προοπτική να γίνει μια άλλη Alexa. Σε αργότερες αναβαθμίσεις θα μπορούσε να έχει πρόσβαση σε άλλες συσκευές αναπαραγωγής όπως είναι οι έξυπνες τηλεοράσεις ή κάποιο ηχοσύστημα για την αναπαραγωγή των media. Επίσης θα μπορούσε να μαθαίνει κρατώντας μνήμη των επιλογών του χρήστη. Με αυτό τον τρόπο θα διαμόρφωνε κάποιο ιστορικό και θα πρότεινε στον χρήστη, αν ο ίδιος το επιθυμούσε, με βάση τις προτιμήσεις του στη μουσική και στα βίντεο.

Οι ψηφιακοί βοηθοί ήρθαν για να μείνουν όπως όλα δείχνουν. Η εξάπλωσή τους είναι ραγδαία και οι δυνατότητες τους πολλές όπως δείξαμε. Σε λίγα χρόνια προβλέπεται να βρίσκονται παντού διευκολύνοντας τη καθημερινότητά μας. Ένας τομέας που μπορεί να ωφεληθεί πολύ από την πρόοδο των ψηφιακών βοηθών και γενικότερα των chatbot, είναι αυτός

του πολιτισμού. Θα μπορούσαν να αλλάξουν πολύ την εμπειρία των επισκεπτών σε χώρους όπως είναι τα μουσεία, εξελίσσοντάς τα και κάνοντάς τα πιο ελκυστικά και δια-δραστικά στο κοινό.

Τα τελευταία χρόνια πολλές πόλεις έχουν υιοθετήσει την ιδέα της χρήσης των QR Code σε μνημεία πολιτιστικής κληρονομιάς, αλλά και τα μουσεία ακολουθούν μια παρόμοια τακτική για τα εκθέματα τους. Όμως ήρθε πλέον η ώρα, με την τεχνολογία των ψηφιακών βοηθών να βρίσκεται πλέον σε υψηλό επίπεδο, να μπουν στα μουσεία ως ψηφιακοί ξεναγοί. Σε ένα επιστημονικό paper [16], στο οποίο συμμετείχα και εγώ, πάνω στη χρήση chatbot σε μουσεία, δημιουργήσαμε μια εφαρμογή για κινητές συσκευές η οποία σε βοηθούσε να επικοινωνείς με τα εκθέματα του μουσείου.

Πρόκειται για έναν ψηφιακό βοηθό για σκηνίτες συσκευές ο οποίος έπαιρνε τη μορφή του εκθέματος που ήθελε ο επισκέπτης να μιλήσει. Το όνομα του ψηφιακού βοηθού ήταν Υπατία (Hyratia). Και για την Hyratia όπως και για την Athena χρησιμοποιήσαμε το Dialogflow για την δημιουργία μιας τεχνητής νοημοσύνης. Όμως αυτό μπορεί να εξελιχθεί περαιτέρω με τη δημιουργία ενός αποκλειστικού AI που θα αντλεί τη πληροφορία που αποζητά ο χρήστης απευθείας από μια βάση στο διαδίκτυο, και όλες οι απαντήσεις να προκύπτουν πιο δυναμικά σε σχέση με το Dialogflow της Google.



Εικόνα 21. Εικόνα εφαρμογής ψηφιακού βοηθού για μουσεία [16].

Στην εφαρμογή της Hyratia, ο χρήστης στέκεται μπροστά από το έκθεμα, εικόνα 21 και εμφανίζεται στην οθόνη του κινητού του η φωτογραφία του εκθέματος και του δίνεται η δυνατότητα να μάθει για αυτό το έκθεμα απλά συνομιλώντας μαζί του. Η εφαρμογή αξιοποιεί την τεχνολογία των BLE συσκευών, δηλαδή Bluetooth 4.0 συσκευών χαμηλής ενέργειας, για τον εντοπισμό των εκθεμάτων μέσα σε ένα χώρο. Ο χρήστης κάνει μια ερώτηση στο έκθεμα χρησιμοποιώντας τη φωνή του. Τότε το chatbot απαντάει στην ερώτηση του χρήστη για το έκθεμα. Με αυτόν τον τρόπο χρήστες μπορούν να μάθουν για την ιστορία του εκθέματος όπως την ηλικία του, το όνομα του δημιουργού, το σκοπό της δημιουργίας του και πολλά άλλα χαρακτηριστικά ξεχωριστά για κάθε έκθεμα. Κάθε έκθεμα μπορεί να έχει και διαφορετική προσωπικότητα κάνοντας την εμπειρία της συνομιλίας μαζί του πιο πραγματική. Επίσης μπορούν να κάνουν ερωτήσεις στους επισκέπτες τραβώντας τους την προσοχή και βάζοντας τους να σκεφτούν. Επιπρόσθετα μπορούν να παίξουν και παιχνίδια μαζί τους, βάζοντας τους

διάφορες αποστολές όπως να μάθουν κάποιο μουσικό κωδικό από κάποιο άλλο έκθεμα μέσα στο μουσείο και να επιστρέψουν να το δώσουν στο αρχικό έκθεμα που τους το ζήτησε και να κερδίσουν κάποιους πόντους. Επίσης οι βοηθοί μπορούν να μιλούν στη γλώσσα που καταλαβαίνει ο χρήστης. Έτσι σε όποιο μουσείο και αν βρίσκεται κάποιος θα μπορεί να μάθει για την ιστορία και τα έργα τέχνης στη δική του γλώσσα χωρίς περιορισμούς και λάθι. Όλα τα παραπάνω κάνουν τα μουσεία να φαίνονται πιο ελκυστικά ιδικά σε άτομα νεαρής ηλικίας, παρέχοντας παράλληλα τη γνώση που πάντα είχαν ως βασικό σκοπό.

Οι ψηφιακοί βοηθοί μπορούν να καλυτερέψουν πολύ τη ζωή μας, με την διαρκή εξέλιξή τους να είναι γρήγορη και σίγουρη. Το βασικό τους συστατικό, που τους καθιστά εξαιρετικά ανθεκτικούς στο πέρασμα του χρόνου, η τεχνητή νοημοσύνη εξελίσσεται με γρήγορους ρυθμούς. Όπως είδαμε οι εφαρμογές τους είναι πολλές και επηρεάζουν σχεδόν όλους τους τομείς. Αυτό θα έχει ως αποτέλεσμα την περαιτέρω εξέλιξη τους με αποτέλεσμα να τους δούμε και σε πιο σημαντικούς ρόλους και τομείς όπως αυτός της επιστήμης, όπως η ιατρική και η αστρονομία συνδράμοντας σε νέες και σημαντικές ανακαλύψεις.

Βιβλιογραφία

1. Android Wikipedia <https://el.wikipedia.org/wiki/Android>
2. The Complete Guide to Chatbots in 2018, SproutSocial
<https://sproutsocial.com/insights/topics/chatbots/>
3. Watson Wikipedia [https://en.wikipedia.org/wiki/Watson_\(computer\)](https://en.wikipedia.org/wiki/Watson_(computer))
4. Amazon Alexa Wikipedia https://en.wikipedia.org/wiki/Amazon_Alexa
5. 2018 State of Chatbots Report <https://www.drift.com/wp-content/uploads/2018/01/2018-state-of-chatbots-report.pdf>
6. Engaging high school students using Chatbots, June 2014
https://www.researchgate.net/publication/266657029_Engaging_high_school_students_using_Chatbots
7. Digital news project , November 2018, The future of voice and implementation for news
https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2018-11/Newman%20-%20Future%20of%20Voice%20FINAL_2.pdf
8. Google Assistant Wikipedia https://en.wikipedia.org/wiki/Google_Assistant
9. Dialogflow Documentation <https://dialogflow.com/docs>
10. Jarvis by Mark Zuckerberg, December 2016 <https://www.facebook.com/notes/mark-zuckerberg/building-jarvis/10103347273888091/?pnref=story>
11. Apple Siri Official <https://www.apple.com/siri/>
12. Apple Siri Wikipedia <https://en.wikipedia.org/wiki/Siri>
13. Spotify SDK Doc <https://developer.spotify.com/documentation/android/quick-start/>
14. YouTube Player Api <https://developers.google.com/youtube/android/player/>
15. YouTube Data Api <https://developers.google.com/youtube/v3/getting-started>
16. L. Toumanidis, P. Karapetros, C. Giannousis, D. G. Kogias, M. Feidakis and C. Z. Patrikakis, "Developing the museum-monumental experience from linear to interactive using chatbot", In Proc. of the International Conference on Strategic Innovative Marketing and Tourism (ICSIMAT), 17-20 October, 2018, Attika, Greece, Springer.
17. AsyncTask in Android, Trishant Sharma, March 22 2018,
<https://medium.com/@trishantsharma1997/asynctask-in-android-f594a565d676>

18. Android Fragment Lifecycle, dev2qa, December 25 2017, <https://www.dev2qa.com/android-fragment-lifecycle/>
19. Codepath, Using the RecyclerView, <https://guides.codepath.com/android/using-the-recyclerview>
20. Research: 90% of US consumers own a smart home device, May 15 2018, IoT, <https://www.iotechnews.com/news/2018/may/15/research-us-consumers-smart-home-device/>
21. IoT Agenda, Internet of Things, June 2018, <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
22. Android Distribution dashboard, 31/01/2019, <https://developer.android.com/about/dashboards/>