

## **‘BotVoyage’**

# **Ανάπτυξη AI εφαρμογής που εξειδικεύεται στην παροχή ταξιδιωτικών πληροφοριών μέσω της πλατφόρμας του Messenger**

---

Η Διπλωματική Εργασία  
παρουσιάστηκε ενώπιον  
του Διδακτικού Προσωπικού του  
Πανεπιστημίου Αιγαίου

---

Σε Μερική Εκπλήρωση  
των Απαιτήσεων για το Δίπλωμα του  
Μηχανικού Πληροφοριακών και Επικοινωνιακών Συστημάτων

---

των

ΑΛΕΞΑΝΔΡΟΥ ΦΑΪΤΑΤΣΙΔΗ ΚΑΙ ΚΩΝΣΤΑΝΤΙΝΟΥ ΜΠΙΟΛΟΤΗ  
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ ΕΠΙΚΥΡΩΝΕΙ  
ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΤΩΝ  
ΦΑΪΤΑΤΣΙΔΗ ΑΛΕΞΑΝΔΡΟΥ ΚΑΙ ΚΩΝΣΤΑΝΤΙΝΟΥ ΜΠΟΛΟΤΗ:

Ιωάννης Χαραλαμπίδης, Επιβλέπων, 12/06/2018

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

---

Ευριπίδης Λουκής, Μέλος

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

---

Δημήτρης Δρόσος, Επίκουρος, Μέλος

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018

## ΠΕΡΙΛΗΨΗ

Η διπλωματική εργασία αυτή περιλαμβάνει κατά κύριο λόγο την ανάπτυξη μίας πρωτότυπης εφαρμογής αυτοματοποιημένου λογισμικού(bot) και την ανάλυση βασικών εννοιών της επιστήμης των υπολογιστών, όπως η τεχνητή νοημοσύνη και τα bots, οι οποίες είναι απαραίτητο να κατανοηθούν για να διαμορφωθεί μία πλήρη εικόνα της επέκτασης τους στον χώρο και φυσικά της λειτουργίας τους. Επιπρόσθετα περιλαμβάνει έρευνα των κυριότερων bot στο διαδίκτυο και συγκεκριμένα στο χώρο του τουρισμού, όπου και προβάλλονται τα χαρακτηριστικά στοιχεία τους, μερικά εκ των οποίων μας ενέπνευσαν για την ανάπτυξη του δικού μας bot. Έτσι στη συνέχεια πραγματεύεται τη σύλληψη ,διαμόρφωση και επεξήγηση μιας εφαρμογής που ασπάζεται τον τομέα της μηχανικής μάθησης και συγκεκριμένα της επεξεργασίας της φυσικής γλώσσας, αποσκοπώντας στην ύπαρξη ενός αυτοματοποιημένου λογισμικού, που θα προσδίδει με φυσικό τρόπο ταξιδιωτικές πληροφορίες, στο χρήστη της πλατφόρμας του Messenger.

© 2018

ΤΩΝ

ΑΛΕΞΑΝΔΡΟΥ ΦΑΪΤΑΤΣΙΔΗ ΚΑΙ ΚΩΝΣΤΑΝΤΙΝΟΥ ΜΠΟΛΟΤΗ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

# **‘BotVoyage’**

## **Development of an AI application which is specialized in providing travel information via the Messenger platform**

### **ABSTRACT**

This thesis includes mostly the development of an original automated software application (bot) and in-depth analysis of basic computer science notions such as artificial intelligence and internet bots, which are essential to be perceived, so as to be fully comprehensible as far as their section and their operability are concerned. Furthermore, it includes inquiry of the main bots which are used in the worldwide web and principally in the tourism sector, where are depicted the main features, some of which have inspired us for the development of our own bot. Thereafter, it follows the conception, modulation and the illustration of an application which uses machine learning and particularly the natural language processing section, in order to develop an automated software which provides a user with trip advices via the messenger platform.

© 2018

ALEXANDROS FAITATSIDIS AND KONSTANTINOS BOLOTIS

Department of Information and Communication Systems Engineering

UNIVERSITY OF THE AEGEAN

## **ΕΥΧΑΡΙΣΤΙΕΣ - ΑΦΙΕΡΩΣΕΙΣ**

Θέλουμε να ευχαριστήσουμε τον κ. Ιωάννη Χαραλαμπίδη, μόνιμο επίκουρο καθηγητή του τμήματος Μ.Π.Ε.Σ, για τη συνεισφορά του σε οποιοδήποτε πρόβλημα προέκυψε και για τη γενικότερη καθοδήγηση που μας προσέφερε καθ' όλη τη διάρκεια διαμόρφωσης της διπλωματικής μας εργασίας.

Θα θέλαμε επίσης να αναγνωρίσουμε τη βοήθεια που λάβαμε από την κ. Ζωή Λαχανά για την κάλυψη οποιασδήποτε απορίας σε μερικά τεχνικά θέματα που αντιμετωπίσαμε και χωρίς εκείνη, η διπλωματική εργασία που παραδίδεται σήμερα δε θα είχε την εξής πλήρη μορφή.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

|  |     |
|--|-----|
| ΠΕΡΙΛΗΨΗ .....   | iii |
| ABSTRACT .....   | iv  |
| ΕΥΧΑΡΙΣΤΙΕΣ - ΑΦΙΕΡΩΣΕΙΣ .....                             | v   |
| ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ .....                                 | vi  |
| ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ .....                                    | ix  |
| ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ .....                                   | x   |
| ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ .....                                | 1   |
| 1.1 Σκοπός της εργασίας .....                              | 1   |
| 1.2 Αντικείμενο της εργασίας.....                          | 1   |
| 1.3 Δομή της εργασίας .....                                | 2   |
| ΚΕΦΑΛΑΙΟ 2 - ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ .....                     | 3   |
| 2.1 Τεχνητή Νοημοσύνη .....                                | 3   |
| 2.1.1 Ορισμός .....  | 3   |
| 2.1.2 Η τεχνητή νοημοσύνη στη ζωή μας .....                | 3   |
| 2.2 Bot .....  | 5   |
| 2.2.1 Ορισμός .....  | 5   |
| 2.2.2 Τα bot στη ζωή μας .....                             | 5   |
| ΚΕΦΑΛΑΙΟ 3-ΑΝΑΛΥΣΗ ΚΥΡΙΟΤΕΡΩΝ BOT .....                    | 8   |
| 3.1 Ανάλυση των κυριότερων bot στον ταξιδιωτικό χώρο ..... | 8   |
| 3.1.1 Sure.....  | 8   |
| 3.1.2 Hipmunk.....   | 11  |
| 3.1.3 MobyTrip.....  | 13  |
| 3.1.4 Kayak.....   | 16  |
| 3.1.5 Gooster .....  | 20  |
| 3.1.6 Mica, the Hipster Cat Bot.....                       | 21  |
| 3.1.7 Nearby Places Bot .....                              | 24  |
| 3.1.8 1800 Order.....                                      | 28  |
| 3.1.9 SkyScanner .....                                     | 29  |
| 3.1.10 Waylo: Hotel Price Prediction.....                  | 32  |
| 3.2 Πίνακας σύγκρισης των χαρακτηριστικών.....             | 34  |
| 3.3 Συμπεράσματα ανάλυσης .....                            | 35  |
| ΚΕΦΑΛΑΙΟ 4-ΜΕΘΟΔΟΛΟΓΙΑ .....                               | 36  |
| 4.1 Σύλληψη και διαμόρφωση ιδέας .....                     | 36  |
| 4.2 Η δυσκολία της ελληνικής γλώσσας – NLP.....            | 37  |
| 4.3 Πλατφόρμες και εργαλεία .....                          | 38  |

|   |    |
|---|----|
| 4.3.1 JavaScript .....  | 38 |
| 4.3.2 Node.js.....  | 39 |
| 4.3.3 Wit.ai .....  | 43 |
| 4.3.4 Npm .....   | 45 |
| 4.3.5 Foursquare (API) .....                                  | 47 |
| 4.3.6 Google Geocoding API .....                              | 49 |
| 4.3.7 Messenger Platform.....                                 | 51 |
| 4.3.8 Visual Studio Code.....                                 | 54 |
| 4.4 Προβλήματα και περιορισμοί εργαλείων .....                | 54 |
| 4.5 Διάγραμμα αρχιτεκτονικής.....                             | 55 |
| ΚΕΦΑΛΑΙΟ 5-ΑΠΑΙΤΗΣΕΙΣ BOTVOYAGE.....                          | 56 |
| 5.1 Διαγράμματα .....   | 56 |
| 5.1.1 Use Case Diagrams.....                                  | 56 |
| 5.2 Απαιτήσεις συστήματος .....                               | 58 |
| 5.2.1 Σύνολο απαιτήσεων.....                                  | 58 |
| 5.2.2 Πίνακες απαιτήσεων.....                                 | 59 |
| ΚΕΦΑΛΑΙΟ 6-ΤΕΧΝΙΚΗ ΑΝΑΛΥΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ.....                | 68 |
| 6.1 Τι είναι το Bot Framework .....                           | 68 |
| 6.2 Azure Bot Service .....                                   | 70 |
| 6.2.1 Η ιδέα του χτίζω ένα bot .....                          | 71 |
| 6.2.2 Azure Bot Service Documentation .....                   | 72 |
| 6.3 Σενάρια Bot .....   | 75 |
| 6.3.1 Commerce Bot Scenario.....                              | 75 |
| 6.3.2 Cortana Skill Bot Scenario .....                        | 75 |
| 6.3.3 Enterprise Productivity Bot Scenario .....              | 75 |
| 6.3.4 Information Bot Scenario .....                          | 75 |
| 6.3.5 Internet of Things Bot Scenario.....                    | 76 |
| ΚΕΦΑΛΑΙΟ 7- BOTVOYAGE.....                                    | 77 |
| 7.1 Απεικόνιση και ανάλυση των λειτουργιών .....              | 77 |
| 7.1.1 Φαγητό.....   | 77 |
| 7.1.2 Ποτό , Ξενοδοχεία, Αξιοθέατα κ.λπ. ....                 | 81 |
| 7.1.3 Πτήσεις.....  | 83 |
| 7.1.4 Αλλαγή γλώσσας.....                                     | 85 |
| 7.1.5 Help .....  | 86 |
| 7.1.6 Start Over .....  | 86 |
| 7.2 Απεικόνιση και ανάλυση διαχειριστικών περιβαλλόντων ..... | 87 |
| 7.2.1 Azure Portal.....                                       | 87 |

|  |     |
|--|-----|
| 7.2.2 Visual Studio Code.....                | 88  |
| 7.2.3 Bot Framework Emulator .....           | 89  |
| 7.2.4 Wit.ai .....                           | 90  |
| 7.2.5 Facebook for Developers Platform.....  | 92  |
| ΚΕΦΑΛΑΙΟ 8 -ΣΥΜΠΕΡΑΣΜΑΤΑ-ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ..... | 96  |
| 8.1 Συμπεράσματα επί των εργαλείων .....     | 96  |
| 8.2 Συμπεράσματα επί της εφαρμογής.....      | 97  |
| 8.3 Επόμενα βήματα.....                      | 99  |
| ΒΙΒΛΙΟΓΡΑΦΙΑ.....                            | 100 |
| ΠΑΡΑΡΤΗΜΑ-[ΟΔΗΓΙΕΣ ΑΝΑΠΤΥΞΗΣ].....           | 103 |
| ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ ΣΗΜΕΙΩΜΑ .....            | 108 |



## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

|  |    |
|--|----|
| Πίνακας 1 - Σύγκριση διάφορων bot..... | 35 |
| Πίνακας 2 - Απαίτηση 1 .....           | 59 |
| Πίνακας 3 - Απαίτηση 2 .....           | 59 |
| Πίνακας 4 - Απαίτηση 3 .....           | 60 |
| Πίνακας 5 - Απαίτηση 4 .....           | 60 |
| Πίνακας 6 - Απαίτηση 5 .....           | 60 |
| Πίνακας 7 - Απαίτηση 6 .....           | 61 |
| Πίνακας 8 - Απαίτηση 7 .....           | 61 |
| Πίνακας 9 - Απαίτηση 8 .....           | 61 |
| Πίνακας 10 - Απαίτηση 9 .....          | 61 |
| Πίνακας 11 - Απαίτηση 10 .....         | 62 |
| Πίνακας 12 - Απαίτηση 11 .....         | 62 |
| Πίνακας 13 - Απαίτηση 12 .....         | 62 |
| Πίνακας 14 - Απαίτηση 13 .....         | 63 |
| Πίνακας 15 - Απαίτηση 14 .....         | 63 |
| Πίνακας 16 - Απαίτηση 15 .....         | 63 |
| Πίνακας 17 - Απαίτηση 16 .....         | 63 |
| Πίνακας 18 - Απαίτηση 17 .....         | 64 |
| Πίνακας 19 - Απαίτηση 18 .....         | 64 |
| Πίνακας 20 - Απαίτηση 19 .....         | 65 |
| Πίνακας 21 - Απαίτηση 20 .....         | 65 |
| Πίνακας 22 - Απαίτηση 21 .....         | 65 |
| Πίνακας 23 - Απαίτηση 22 .....         | 66 |
| Πίνακας 24 - Απαίτηση 23 .....         | 66 |
| Πίνακας 25 - Απαίτηση 24 .....         | 66 |
| Πίνακας 26 - Απαίτηση 25 .....         | 67 |

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

|  |    |
|--|----|
| Εικόνα 1 – Τεχνητή νοημοσύνη - <a href="https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/">https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/</a> ..... | 4  |
| Εικόνα 2 – robots.txt - <a href="https://www.google.com/robots.txt">https://www.google.com/robots.txt</a> .....  | 5  |
| Εικόνα 3 – Sure_1 - Messenger.....   | 9  |
| Εικόνα 4 – Sure_2 - Messenger.....   | 10 |
| Εικόνα 5 – Sure_3 - Messenger.....   | 10 |
| Εικόνα 6 – Sure_4 - Messenger.....   | 11 |
| Εικόνα 7 – Hipmunk_1 - Messenger.....  | 12 |
| Εικόνα 8 – Hipmunk_2 - Messenger.....  | 12 |
| Εικόνα 9 – Hipmunk_3 - Messenger.....  | 13 |
| Εικόνα 10 – MobyTrip_1 - Messenger.....  | 14 |
| Εικόνα 11 – MobyTrip_2 - Messenger.....  | 15 |
| Εικόνα 12 – MobyTrip_3 - Messenger.....  | 16 |
| Εικόνα 13 – Kayak_1 - Messenger.....   | 17 |
| Εικόνα 14 – Kayak_2 - Messenger.....   | 18 |
| Εικόνα 15 – Kayak_3 - Messenger.....   | 19 |
| Εικόνα 16 – Kayak_4 - Messenger.....   | 19 |
| Εικόνα 17 -Gooster_1 - Messenger.....  | 20 |
| Εικόνα 18 – Gooster_2 - Messenger .....  | 21 |
| Εικόνα 19 - Mica_1 - Messenger .....   | 22 |
| Εικόνα 20 – Mica_2 - Messenger.....  | 22 |
| Εικόνα 21 - Mica_3 - Messenger .....   | 23 |
| Εικόνα 22 – Mica_4 - Messenger.....  | 23 |
| Εικόνα 23 – Mica_5 - Messenger.....  | 24 |
| Εικόνα 24 – Nearby_1 - Messenger .....   | 25 |
| Εικόνα 25 – Nearby_2 - Messenger .....   | 25 |
| Εικόνα 26 – Nearby_3 - Messenger .....   | 26 |
| Εικόνα 27 – Nearby_4 - Messenger .....   | 27 |
| Εικόνα 28 – Nearby_5 - Messenger .....   | 27 |
| Εικόνα 29 – Nearby_6 - Messenger .....   | 28 |
| Εικόνα 30 – Nearby_7 - Messenger .....   | 28 |
| Εικόνα 31 – 1800 - Messenger.....  | 29 |
| Εικόνα 32 – SkyScanner_1 - Messenger.....  | 29 |
| Εικόνα 33 – SkyScanner_2 - Messenger.....  | 30 |

|  |    |
|--|----|
| Εικόνα 34 – SkyScanner_3 - Messenger.....  | 30 |
| Εικόνα 35 -SkyScanner_4 - Messenger.....   | 31 |
| Εικόνα 36 – SkyScanner_5 - Messenger.....  | 31 |
| Εικόνα 37 – SkyScanner_6 - Messenger.....  | 32 |
| Εικόνα 38 – Waylo_1 - Messenger .....  | 32 |
| Εικόνα 39 – Waylo_2 - Messenger .....  | 33 |
| Εικόνα 40 – Waylo_3 - Messenger .....  | 33 |
| Εικόνα 41 – Waylo_4 - Messenger .....  | 34 |
| Εικόνα 42 – Node.js Map - Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2017).<br>Node. js in Action. Manning Publications.....   | 41 |
| Εικόνα 43 – I/O bound - Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2017).<br>Node. js in Action. Manning Publications.....   | 42 |
| Εικόνα 44 – Wit.ai_ex1 – Wit.ai( το δικό μας).....   | 45 |
| Εικόνα 45 – Wit.ai_ex2 – Wit.ai( το δικό μας).....   | 45 |
| Εικόνα 46 – npm_init – Node.js( το δικό μας).....  | 46 |
| Εικόνα 47 – package.json – Node.js( το δικό μας).....  | 46 |
| Εικόνα 48 – package install – Node.js( το δικό μας) .....  | 47 |
| Εικόνα 49 – chalk – Node.js( το δικό μας).....   | 47 |
| Εικόνα 50 – Foursquare response - <a href="https://developer.foursquare.com/docs/api/venues/details">https://developer.foursquare.com/docs/api/venues/details</a> .....  | 49 |
| Εικόνα 51 – geocoding steps.....   | 50 |
| Εικόνα 52 - Messenger Platform Webview - <a href="https://developers.facebook.com/docs/messenger-platform/webview">https://developers.facebook.com/docs/messenger-platform/webview</a> .....   | 51 |
| Εικόνα 53 - Messenger Platform Payments –.....   | 52 |
| Εικόνα 54 - Messenger Platform Webhooks –.....   | 53 |
| Εικόνα 55 - Bot Framework - <a href="http://www.garypretty.co.uk/2018/02/21/whats-next-for-the-microsoft-bot-framework/">http://www.garypretty.co.uk/2018/02/21/whats-next-for-the-microsoft-bot-framework/</a> .....  | 68 |
| Εικόνα 56 - Bot Framework with Channels – .....  | 69 |
| Εικόνα 57 Bot Framework_Whole - <a href="https://www.quora.com/What-is-Microsofts-Bot-Framework">https://www.quora.com/What-is-Microsofts-Bot-Framework</a><br>.....   | 69 |
| Εικόνα 58 - Info Bot - <a href="https://dev.botframework.com/">https://dev.botframework.com/</a> .....   | 70 |
| Εικόνα 59 - Bot steps - <a href="https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-3.0">https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-3.0</a> .....  | 71 |
| Εικόνα 60 - Bot Service- .....   | 73 |
| Εικόνα 61 - .NET - <a href="https://docs.microsoft.com/en-us/azure/bot-service/dotnet/bot-builder-dotnet-quickstart?view=azure-bot-service-3.0">https://docs.microsoft.com/en-us/azure/bot-service/dotnet/bot-builder-dotnet-quickstart?view=azure-bot-service-3.0</a> ..... | 74 |
| Εικόνα 62 - Φαγητό_1 - Messenger .....   | 77 |
| Εικόνα 63 - Φαγητό_2 - Messenger .....   | 78 |

|  |    |
|--|----|
| Εικόνα 64 - Φαγητό_3 - Messenger .....                     | 78 |
| Εικόνα 65 – Αποστολή τοποθεσίας - Messenger .....          | 79 |
| Εικόνα 66 - Προβολή Φαγητών - Messenger .....              | 80 |
| Εικόνα 67 - Ανακατεύθυνση σε Foursquare - Messenger .....  | 80 |
| Εικόνα 68 - Κατανόηση οντότητας burger - Messenger .....   | 81 |
| Εικόνα 69 - Προβολή burger - Messenger .....               | 81 |
| Εικόνα 70 - Κατανόηση ποτού - Messenger .....              | 81 |
| Εικόνα 71 - Κατανόηση κρασιού - Messenger.....             | 82 |
| Εικόνα 72 - Κατανόηση ξενοδοχείου - Messenger .....        | 82 |
| Εικόνα 73 - Προβολή ξενοδοχείων - Messenger.....           | 82 |
| Εικόνα 74 - Κατανόηση αξιοθέατων - Messenger .....         | 82 |
| Εικόνα 75 - Προβολή αξιοθέατων - Messenger .....           | 83 |
| Εικόνα 76 - Ροή πτήσεων - Messenger .....                  | 84 |
| Εικόνα 77 - Αποτελέσματα Kayak - Messenger .....           | 84 |
| Εικόνα 78 - Λάθη στις πτήσεις - Messenger .....            | 85 |
| Εικόνα 79 - Αλλαγή γλώσσας - Messenger.....                | 85 |
| Εικόνα 80 - Help_1 - Messenger.....                        | 86 |
| Εικόνα 81 - Help_2 - Messenger.....                        | 86 |
| Εικόνα 82 - StartOver_1 - Messenger .....                  | 87 |
| Εικόνα 83 - StartOver_2 - Messenger .....                  | 87 |
| Εικόνα 84 - Azure Dashboard – Azure Portal( δικό μας)..... | 87 |
| Εικόνα 85 - App Dashboard – Azure Portal( δικό μας) .....  | 88 |
| Εικόνα 86 - Visual Studio Code – (δικό μας).....           | 89 |
| Εικόνα 87 - Bot Framework Emulator – (δικό μας).....       | 89 |
| Εικόνα 88 - Wit Main Panel – Wit.ai( δικό μας).....        | 90 |
| Εικόνα 89 – Wit Entity Panel – Wit.ai( δικό μας).....      | 90 |
| Εικόνα 90 – Wit Entity Panel_2 – Wit.ai( δικό μας).....    | 91 |
| Εικόνα 91 – Wit Inbox Panel – Wit.ai( δικό μας) .....      | 91 |
| Εικόνα 92 - Wit Settings Panel – Wit.ai( δικό μας) .....   | 92 |
| Εικόνα 93 - Facebook Dev Panel – ( δικό μας).....          | 92 |
| Εικόνα 94 - Facebook Dev App Panel – ( δικό μας) .....     | 93 |
| Εικόνα 95 - Facebook Dev Settings – ( δικό μας).....       | 93 |
| Εικόνα 96 - Facebook Dev Roles – ( δικό μας).....          | 94 |
| Εικόνα 97 - Facebook Dev Products – (δικό μας).....        | 95 |
| Εικόνα 98 - Facebook Dev Webhooks – (δικό μας).....        | 95 |

# ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

## 1.1 Σκοπός της εργασίας

Στη σημερινή εποχή λόγω της χαοτικής ποσότητας της πληροφορίας που εδρεύει στο διαδίκτυο, γεγονός που επηρεάζει φυσικά και την ποιότητα της, ο χρήστης δε βρίσκεται πολλές φορές σε θέση να πραγματοποιήσει τη σωστή επιλογή για την ενημέρωση του. Παράλληλα έχει κλιμακωθεί η χρήση αυτοματοποιημένων τμημάτων λογισμικού τα οποία εξυπηρετούν πλήθος ανθρωπίνων αναγκών, είτε αυτές είναι οικονομικές, προσωπικές, πληροφοριακές. Ωστόσο ο χώρος που θα καλύψει η εργασία μας σκοπεύει να είναι ο ταξιδιωτικός, καθώς ολοένα και μεγαλύτερος αριθμός ανθρώπων έχουν την αμεσότητα και τη δυνατότητα να ταξιδεύουν ανά την υδρόγειο με την παρούσα τεχνολογία και τα υψηλά βιοποριστικά ποσοστά. Έτσι επιλέγουν να χρησιμοποιούν τον παραπάνω τύπο λογισμικού, το οποίο εν τέλει μπορεί να αναχθεί σε έναν έξυπνο κι εύκολο σε χρήση 'σύμβουλο'. Ειδικότερα, στα πλαίσια λειτουργίας του, είναι απαραίτητη η ύπαρξη συγκεντρωμένης πληροφορίας με κατανοητή δομή, όπου θα παρέχεται με διασκεδαστικό τρόπο, σε μορφή συνομιλίας, χωρίς ο χρήστης να προσκολλάται σε λεπτομέρειες που θα τον αποκλίνουν από την κάλυψη της πραγματικής του ανάγκης.

Στόχος της διπλωματικής μας εργασίας είναι η εύρεση διαφόρων ταξιδιωτικών πληροφοριών, και η παροχή αυτών στον χρήστη μας, για τη διευκόλυνση του σε σχέση με κύριες ανάγκες του, όπως είναι η ψυχαγωγία, η διασκέδαση, η διαμονή, η νοσηλεία και τα ταξίδια. Απαραίτητο στοιχείο της εργασίας μας, είναι η ευρεία εφαρμογή της σε κάθε σημείο του κόσμου για κάθε άνθρωπο μέσα σε αυτόν.

Ο απώτερος σκοπός μας είναι η απλοποίηση κάποιων διαδικασιών που περιλαμβάνονται σε ένα ταξίδι, και επ' ουδενί η καθοδήγηση των επιλογών του χρήστη αποκλειστικά βάση ενός ψηφιακού μοτίβου, καθώς συνιστά απαραίτητη η ύπαρξη της εξερεύνησης και του αυτοσχεδιασμού, ώστε να διατηρηθεί η καθ' εαυτού παραδοσιακή υπόσταση του ταξιδιού.

## 1.2 Αντικείμενο της εργασίας

Το αντικείμενο της εργασίας είναι η ανεύρεση των κατάλληλων εργαλείων και πλατφορμών ανάπτυξης, οι οποίες θα οδηγήσουν στο καλύτερο δυνατό αποτέλεσμα, ώστε να προσδώσουμε ενημέρωση στο χρήστη με αξιοπιστία, ευκολία και προφανώς ταχύτητα. Με λίγα λόγια η διεπαφή μας θα πρέπει να χαρακτηρίζεται με αμεσότητα, για να δίνεται η εντύπωση ότι πραγματοποιείται μία συνομιλία με έναν φίλο κι όχι με ένα απόσταγμα τεχνητής νοημοσύνης. Ειδικότερα, μεγάλο ενδιαφέρον παρουσιάζεται στην κατάλληλη επιλογή λογισμικού κατανόησης της φυσικής γλώσσας, καθώς θα πρέπει να συνδυάζεται εναρμονισμένα με τις δύο γλώσσες της εφαρμογής μας, τα ελληνικά και τα αγγλικά. Προφανώς, λόγω της πολυσύνθετης δομής της γλώσσας, τα ελληνικά χρειάζονται ιδιαίτερη ενασχόληση για την επίτευξη του επιθυμητού αποτελέσματος. Επιπρόσθετα αναπόσπαστο και απαραίτητο τμήμα της εργασίας μας αποτελεί η ανεύρεση της κατάλληλης πηγής δεδομένων, η οποία θα εμπεριέχει αυτό που ακριβώς χρειάζεται ο χρήστης στον πιο αξιόπιστο βαθμό χωρίς περιττή πληροφορία. Τέλος, ίσως το σημαντικότερο των βημάτων υλοποίησης είναι η επιλογή μίας ευρέως αναγνωρίσιμης πλατφόρμας, πάνω στην οποία θα συνδυαστούν τα παραπάνω, ώστε να γίνει η σύνθεση της εφαρμογής μας και να παραδοθεί άμεσα στο χρήστη. Έτσι

ταυτόχρονα, εκπληρώνεται ένα σημαντικό μέρος του στόχου μας που αναλύθηκε παραπάνω, η κλιμάκωση δηλαδή σε παγκόσμιο επίπεδο.

### 1.3 Δομή της εργασίας

Στο κεφάλαιο 2, θα ξεκινήσουμε με την επεξήγηση των απαραίτητων ορισμών για την κατανόηση του υπόβαθρου που συνθέτει την ειδικότερη υπόσταση της εφαρμογής μας, οι οποίοι βασίζονται σε πρωταρχικής σημασίας έννοιες, όπως η τεχνητή νοημοσύνη και τα bots. Στη συνέχεια θα πραγματοποιηθεί μία ιστορική αναδρομή και μία ανάλυση περί αυτών.

Στο κεφάλαιο 3, θα προχωρήσουμε στην ανάλυση των κυριότερων περιπτώσεων χρήσης στον εξειδικευμένο χώρο της διπλωματικής μας εργασίας, τον ταξιδιωτικό και θα πραγματοποιηθεί μια οπτική παρουσίαση των ήδη υπαρχόντων.

Στο κεφάλαιο 4, θα παρουσιαστεί ο τρόπος με τον οποίο συλλήφθηκε η ιδέα μας και θα παρουσιαστούν εκτενώς τα απαραίτητα εργαλεία και πλατφόρμες σε συνδυασμό με κάποια προβλήματα και περιορισμούς που ανάχθηκαν στην επιφάνεια

Στο κεφάλαιο 5, θα προβάλλουμε μία πιο οπτική απεικόνιση της εφαρμογής μας, ώστε να γίνει αντιληπτός από τον αναγνώστη ο τρόπος με τον οποίο μπορεί να πραγματοποιήσει οποιαδήποτε δυνατή λειτουργία. Ειδικότερα περιλαμβάνονται διαγράμματα απεικόνισης περιπτώσεων χρήσης. Στη συνέχεια καθορίζουμε τις απαιτήσεις συστήματος, είτε υλοποιήθηκαν, είτε πρόκειται να υλοποιηθούν στο μέλλον.

Στο κεφάλαιο 6, αναπτύσσεται μία εκτενώς θεωρητική τεχνική προσέγγιση σε ένα πολύ βασικό κι αναπόσπαστο τμήμα της διπλωματικής μας εργασίας, την πλατφόρμα υλοποίησης.

Στο κεφάλαιο 7 απεικονίζονται με εικόνες όλο το σύνολο των λειτουργιών και των διαχειριστικών περιβαλλόντων των εργαλείων μας.

Στο κεφάλαιο 8 περιγράφεται το ερωτηματολόγιο και εν συνεχεία αναλύονται τα αποτελέσματα αυτού.

Στο κεφάλαιο 9 παρουσιάζονται τα συμπεράσματα επί των εργαλείων, της εφαρμογής και φυσικά επί του ερωτηματολογίου, ώστε να πραγματοποιηθεί η σύνθεση των μελλοντικών βημάτων της εφαρμογής μας, η οποία και αναλύεται.

Τέλος, περιλαμβάνεται η βιβλιογραφία και το παράρτημα, απαραίτητα για να υπάρξει μια σωστή και ολοκληρωμένη δομή.

## ΚΕΦΑΛΑΙΟ 2 - ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

### 2.1 Τεχνητή Νοημοσύνη

#### 2.1.1 Ορισμός

Η τεχνητή νοημοσύνη [2] (Artificial Intelligence ,AI}) αναφέρεται στον κλάδο της πληροφορικής και εστιάζει στη δημιουργία έξυπνων μηχανών που λειτουργούν και συμπεριφέρονται όπως οι άνθρωποι. Μερικές από τις δραστηριότητες στο χώρο των υπολογιστών με τη δράση της τεχνητής νοημοσύνης έχουν σχεδιαστεί για να περιλαμβάνουν:

1. Κατανόηση ομιλίας.
2. Μάθηση.
3. Σχεδιασμό.
4. Επίλυση προβλημάτων.

#### 2.1.2 Η τεχνητή νοημοσύνη στη ζωή μας

##### 2.1.2.1 Κλάδοι συσχέτισης της AI

Η τεχνολογία γνώσης [3](Knowledge engineering) είναι ένα σημαντικό κομμάτι στην έρευνα της τεχνητής νοημοσύνης. Τα εξαρτήματα μπορούν να δρουν και να συμπεριφέρονται συχνά όπως οι άνθρωποι μόνο όταν έχουν στην κατοχή τους άφθονες πληροφορίες που σχετίζονται με τον έξω κόσμο. Η τεχνητή νοημοσύνη πρέπει να έχει σύνδεση σε αντικείμενα, σε κατηγορίες, σε ιδιότητες και σχέσεις μεταξύ όλων αυτών για να μπορέσουν να εφαρμόσουν την τεχνολογία της γνώσης.

Η εκμάθηση μηχανών [3] (Machine learning ) είναι ένας επίσης μεγάλο κομμάτι της τεχνητής νοημοσύνης. Η μάθηση χωρίς κάποιου είδους εποπτεία απαιτεί την δυνατότητα να αναγνωρίζει μονοπάτια εισαγωγικών ροών, ενώ η μάθηση με υπάρχει επίβλεψη συμπεριλαμβάνει τη σωστή ταξινόμηση και παλινδρόμηση των ροών αυτών. Η σωστή ταξινόμηση καθορίζει την κατηγορία που ανήκει ένα αντικείμενο εισροής, ενώ η παλινδρόμηση ασχολείται με την απόκτηση ενός συνόλου X αριθμητικών παραδειγμάτων εισόδων ή εξόδων των ροών, με αποτέλεσμα να υπάρχουν οι κατάλληλες λειτουργίες που επιτρέπουν την έγκυρη παραγωγή εξόδων από τις αντίστοιχες εισόδους που δέχθηκαν.

Η ρομποτική [3]τέλος εξαρτάται σε μεγάλο βαθμό από την τεχνητή νοημοσύνη. Τα ρομπότ χρειάζονται την κατάλληλη ευφυΐα για να διαχειριστούν εργασίες όπως είναι ο χειρισμός των αντικειμένων και της πλοήγησης, μαζί με υπό-προβλήματα εντοπισμού, σχεδιασμού κινήσεων και χαρτογράφησης.


### 2.1.2.2 Πως μπορεί να γίνει η τεχνητή νοημοσύνη επικίνδυνη;

Από τις διάφορες και ποικίλες έρευνες που έχουν γίνει πάνω στον τομέα της τεχνητής νοημοσύνης , οι περισσότεροι ερευνητές συμφωνούν στο ότι είναι απίθανο να φτάσουμε σε ένα σημείο που να μπορούμε να πούμε ότι μέσω της τεχνητής νοημοσύνης μπορούμε να παρουσιάσουμε συναισθήματα του ανθρώπου όπως για παράδειγμα είναι η αγάπη ή το μίσος. Επίσης δε θα είναι δυνατόν να περιμένουμε μία μηχανή που χρησιμοποιεί την τεχνητή νοημοσύνη να αντιδράσει εκ προθέσεως καλοπροαίρετα ή κακόβουλα. Παρόλο αυτά σύμφωνα με τις μελέτες οι ειδικοί παρουσίασαν τις δύο πιθανούς τρόπους με τους οποίους η τεχνητή νοημοσύνη θα μπορούσε να αποκαλεστεί ως επικίνδυνη[4]:

1. Η τεχνητή νοημοσύνη έχει προγραμματιστεί για να κάνει κάτι καταστροφικό: Τα αυτόνομα όπλα περιέχουν συστήματα τεχνητής νοημοσύνης που προγραμματίζονται για να σκοτώσουν. Στα χέρια λάθος ανθρώπου , την κατάλληλη στιγμή αυτό θα μπορούσε να υπάρξει καταστροφικό προκαλώντας μαζικές απώλειες.

2. Η τεχνητή νοημοσύνη έχει προγραμματιστεί για να κάνει κάτι ωφέλιμο , αλλά αναπτύσσει μια καταστρεπτική μέθοδο για την επίτευξη του στόχου του :Αυτό μπορεί να συμβεί κάθε φορά που αποτυγχάνουμε να δώσουμε τις σωστές εντολές και καθοδήγηση που έχουμε αναθέσει σαν ανθρώπινο στόχο με τη βοήθεια της τεχνητής νοημοσύνης , γεγονός που αποτελεί από μόνο του δύσκολο έως σχεδόν απίθανο να συμβεί. Έστω για παράδειγμα οτι χρησιμοποιούμε ένα αυτοκίνητο το οποίο χρησιμοποιεί την τεχνική αυτή. Έάν το ζητήσουμε να μας καθοδηγήσει όσο το δυνατό πιο γρήγορα στο αεροδρόμιο , αυτό ναι μεν θα ακολουθήσει σωστά τις εντολές που το δόθηκαν αλλά επίσης θα κάνει την οδήγηση του πιο επιθετική και πιο επικίνδυνη για να καταφέρει να υλοποιήσει το στόχο που του δόθηκε.

Όπως συμπεραίνουμε από τους δύο πιο πάνω λόγους όντως η τεχνητή νοημοσύνη δε μπορεί να φθάσει σε επίπεδο ανάπτυξης συναισθημάτων ίδια με αυτά των ανθρώπων παρόλα αυτά

| <b>Myth:</b><br>Superintelligence<br>by 2100 is inevitable | <table border="1"><thead><tr><th>Mon</th><th>Tue</th><th>Wed</th><th>Thr</th><th>Fri</th><th>Sat</th><th>Sun</th></tr></thead><tbody><tr><td></td><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr><tr><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr><tr><td>19</td><td>20</td><td>✓</td><td>22</td><td>23</td><td>24</td><td>25</td></tr><tr><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td></td><td></td></tr></tbody></table> | Mon | Tue | Wed | Thr | Fri | Sat | Sun |  |  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | ✓ | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |  |  | <b>Fact:</b><br>It may happen in<br>decades, centuries<br>or never: AI experts<br>disagree & we<br>simply don't know |  |
|--|---|-----|-----|-----|-----|-----|-----|-----|--|--|--|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|---|----|----|----|----|----|----|----|----|----|--|--|--|---|
| Mon  | Tue   | Wed | Thr | Fri | Sat | Sun |     |     |  |  |  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |  |  |  |   |
|  |   |     | 1   | 2   | 3   | 4   |     |     |  |  |  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |  |  |  |   |
| 5  | 6   | 7   | 8   | 9   | 10  | 11  |     |     |  |  |  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |  |  |  |   |
| 12   | 13  | 14  | 15  | 16  | 17  | 18  |     |     |  |  |  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |  |  |  |   |
| 19   | 20  | ✓   | 22  | 23  | 24  | 25  |     |     |  |  |  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |  |  |  |   |
| 26   | 27  | 28  | 29  | 30  |     |     |     |     |  |  |  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |    |    |    |    |    |    |    |    |    |  |  |  |   |

Εικόνα 1 – Τεχνητή νοημοσύνη - <https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/>



## 2.2 Bot

### 2.2.1 Ορισμός

Το bot διαδικτύου [1], το οποίο είναι γνωστό και ως www robot η απλά bot, είναι ένα πρόγραμμα λογισμικού το οποίο εκτελεί αυτοματοποιημένες εργασίες μέσω του διαδικτύου.

### 2.2.2 Τα bot στη ζωή μας

#### 2.2.2.1 Χρήση

Στις περισσότερες περιπτώσεις τα bots εκτελούν σχετικά απλές λειτουργίες που θα πρέπει να επαναληφθούν εκατοντάδες ή χιλιάδες φορές. Μία κλασσική εφαρμογή των bots είναι οι αράχνες του διαδικτύου (Web spiders), οι οποίες περιφέρονται από ιστοσελίδα σε ιστοσελίδα και χρησιμοποιούνται για την ανάλυση της σε ρυθμό πολλαπλάσιο από ότι θα μπορούσε ένας άνθρωπος. Για παράδειγμα ακόμη και οι μεγαλύτερες μηχανές αναζήτησης (Google , Yahoo, κοκ) χρησιμοποιούν αυτήν την τεχνική για την ανάλυση και την ταξινόμηση των ιστοσελίδων σύμφωνα με διάφορες λέξεις – κλειδιά , ούτως ώστε στη συνέχεια να μπορούν να παρουσιάσουν στο χρήστη τα αποτελέσματα της αναζήτησης σε πολύ μικρό χρονικό διάστημα. Μετέπειτα οι διάφοροι web servers μπορούν να δημιουργήσουν ένα απλό κείμενο με το όνομα «robots.txt», το οποίο θα περιέχει κανόνες που θα πρέπει να τηρήσουν τα bots που επισκέπτονται την σελίδα. Παρόλα αυτά όμως , μπορεί κάποιο bot να αγνοήσει αυτούς του κανόνες[5]

```
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*%
Allow: /?hl=*%gws_rd=ssl$
Disallow: /?hl=*%*&gws_rd=ssl
Allow: /?gws_rd=ssl$
Allow: /?pt1=true$
Disallow: /imgres
Disallow: /u/
Disallow: /preferences
Disallow: /setprefs
Disallow: /default
```

**Εικόνα 2 – robots.txt - <https://www.google.com/robots.txt>**

Παραπάνω φαίνεται ένα κομμάτι του αρχείου robots.txt της google συγκεκριμένα , κατανοώντας έτσι ποιες λειτουργίες έχει ανοιχτές και ποιες όχι. Μία άλλη χρήση των bots είναι σε εφαρμογές όπου απαιτείται υψηλή ταχύτητα απόκρισης, υψηλότερη από αυτή που έχει ο άνθρωπος (Πχ bots

που συμμετέχουν σε δημοπρασίες ) ή σε εφαρμογές όπου απαιτείται η προσομοίωση της ανθρώπινης συμπεριφοράς(πχ bots που συμμετέχουν σε συζητήσεις – chat bots). Ένα χαρακτηριστικό παράδειγμα [6] από το οποίο μπορούμε να κατανοήσουμε τη δύναμη και την ταχύτητα με την οποία μπορούν να δράσουν έρχεται από έναν δημοσιογράφο του CNN. Ο ίδιος παρατήρησε σε ένα μικρό χρονικό διάστημα την επίσκεψη εκατομμυρίων bot στη συγκεκριμένη στήλη CNN IReport, στην οποία έγραφε ο Chris Morrow ο οποίος το διάστημα εκείνο είχε τη φήμη ότι τα άρθρα του ήταν από τα πιο πολυδιαβασμένα. Ο αριθμός των views ανέβαινε με πολύ γρήγορο ρυθμό αλλά αυτό συνδεόταν με την χρησιμοποίηση bot για την ανάδειξη του σε διάσημο και κορυφαίο. Προφανώς από την στιγμή που η στήλη IReport συνειδητοποίησε το τι συνέβαινε, αφαίρεσε τα περισσότερα από τα άρθρα του.

#### 2.2.2.2 Συνήθη ροή διαδικασίας των bot

Όλα τα bot έχουν ένα κοινό σύνολο ιδιοτήτων. Μπορούμε να πούμε ότι ένα bot:

- Κυρίως υπάρχει , άμεσα ή έμμεσα , για οικονομικά οφέλη.
- Μιμείται , σε κάθε βαθμό , τις κινήσεις του ανθρώπου μέσω ενός ηλεκτρονικού υπολογιστή.
- Επαναλαμβάνει μικρές ενέργειες πολλές φορές.
- Ξεκινά τη δραστηριότητα στο συγκεκριμένο χώρο που του ζητείται.
- Εκτελεί μόνο τις ελάχιστες απαραίτητες ενέργειες που απαιτείται για να ολοκληρώσει την αποστολή του.

Όσο αφορά σε ατομικό επίπεδο , η συμπεριφορά του bot εμπίπτει σε μία από τις ακόλουθες γενικές ταξινομήσεις :

1. Στέλνει ένα απλό μήνυμα (AD click bot).
2. Στέλνει μήνυμα και περιμένει για απάντηση (Online Banking Bot).
3. Στέλνει πολλαπλά μηνύματα ασυγχρόνιστα (Denial of Service Bots).
4. Στέλνει πολλαπλά μηνύματα ασυγχρόνιστα και περιμένει για μία ή και περισσότερες απαντήσεις (Online Spam Bots).

Στις περιπτώσεις 2 , 4 όπου το bot περιμένει για απάντηση θα πρέπει η διεύθυνση IP του εκδότη να είναι έγκυρη για να μπορέσει να περάσει στον αποδέκτη και να παραληφθεί κανονικά.

#### 2.2.2.3 Κακόβουλες χρήσεις

Με τέτοια δυναμική και πολυπλοκότητα σίγουρα δε θα αργούσε να εμφανιστεί η πρώτη χρήση ενός bot με κακόβουλη πρόθεση. Έτσι , τα bots χρησιμοποιούνται πολύ συχνά από hackers , για τον συντονισμό και τη διεξαγωγή διαδικτυακών επιθέσεων σε servers ή για άλλους σκοπούς. Η πιο διαδομένη και συχνή μορφή μιας επίθεσης ενός hacker μέσω bots είναι να έχει στη διάθεση του πολλά από αυτά , να σχηματίζει ένα δίκτυο από αυτά (bot net ) και να επιτείθεται ταυτόχρονα σε έναν server ενός διαδικτύου[5].

Οι δημοφιλέστερες κατηγορίες στις οποίες μπορούν να διαχωριστούν τα bots είναι σε :

1. SpamBots: Ψάχνουν κάθε σελίδα του διαδικτύου για διευθύνσεις email , ούτως ώστε αυτές στη συνέχεια να χρησιμοποιηθούν για spamming.
2. Bots που κατεβάζουν ολόκληρα sites και χρησιμοποιούν όλο το διαθέσιμο bandwidth του web server με αποτέλεσμα να μη μπορούν να εξυπηρετηθούν άλλοι χρήστες
3. Bots που χρησιμοποιούνται απο διάφορους ιούς υπολογιστών (trojan ).
4. Τα pc zombies (Υπολογιστής ο οποίος είναι συνδεδεμένος στο διαδίκτυο και ελέγχεται απο έναν άλλον εξωτερικό χρήστη ) χρησιμοποιούνται πολλές φορές για τη δημιουργία bots και τη διεξαγωγή διαδικτυακής επίθεσης σε έναν server.
5. DDoS επίθεση.

#### 2.2.2.4 Οι θετικές επιδράσεις των bot

Οι εταιρίες και οι πελάτες μπορούν να επωφεληθούν απο τα διαδικτυακά bot. Επιτρέπουν στους πελάτες να επικοινωνούν με την εταιρία χωρίς να είναι απαραίτητο να επικοινωνήσουν με κάποιο άνθρωπο. Για παράδειγμα μία Ολλανδική αεροπορική εταιρία , έχει δημιουργήσει ένα chat bot το οποίο επιτρέπει στους πελάτες του να παραλαμβάνει τα εισιτήρια τους μέσω αυτού , να υπενθυμίζει check in, και άλλες χρήσιμες πληροφορίες περί της πτήσης που ενδιαφέρεται[7].

Τα Chatbots χρησιμοποιούνται σε καθημερινή βάση. Για παράδειγμα το Google Assistant όπως και η Siri (η κυρία των iOS) περιέχουν φόρμες ενός chatbot. Οι εφαρμογές αυτές επιτρέπουν στους διάφορες χρήστες να ρωτούν ερωτήσεις και να λαμβάνουν απαντήσεις μέσω Artificial Intelligence (AI) συστήματος.

## ΚΕΦΑΛΑΙΟ 3-ΑΝΑΛΥΣΗ ΚΥΡΙΟΤΕΡΩΝ BOT

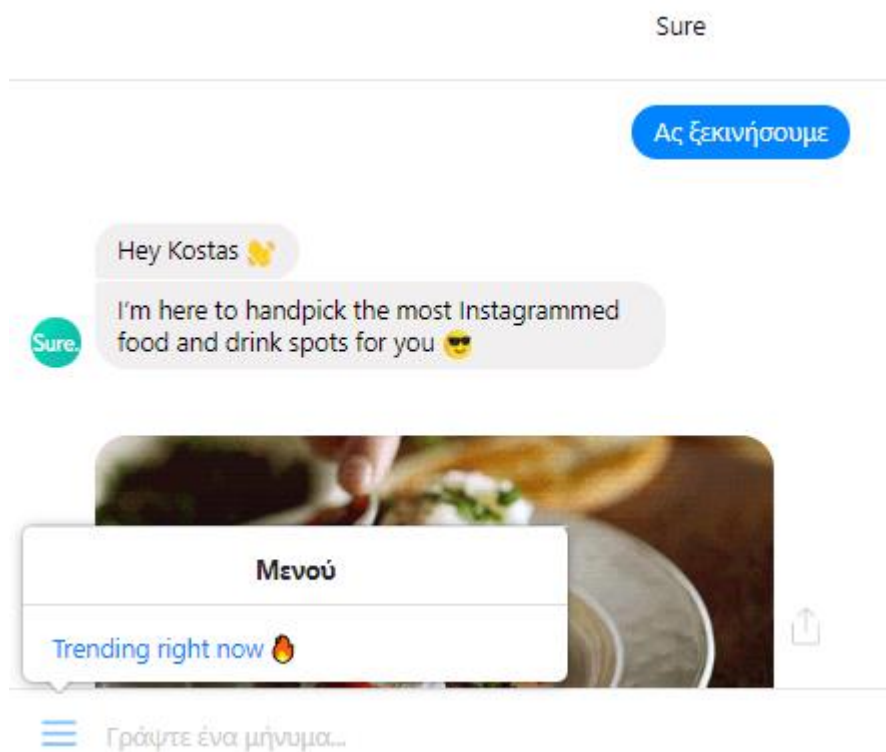
### 3.1 Ανάλυση των κυριότερων bot στον ταξιδιωτικό χώρο

Έχοντας περιγράψει τη γενικότερη εξάπλωση των bots παραπάνω, δεν αποτελεί έκπληξη το γεγονός ότι έχουν ευρεία εφαρμογή στον τομέα του τουρισμού και των ταξιδιών. Θα μπορούσαμε ίσως να τα κατηγοριοποιήσουμε σε δύο κατηγορίες, εκείνα που παρέχουν μία αυτοματοποιημένη υπηρεσία στο πλαίσιο της ιστοσελίδας μίας ταξιδιωτικής εταιρείας και δεύτερον σε εκείνα που έχουν εγκαθιδρυθεί μέσω της πλατφόρμας του messenger, στοχεύοντας να εξυπηρετήσουν τον πελάτη σε θέματα πληροφοριών και κρατήσεων. Όπως θα έχει γίνει αντιληπτό, η εφαρμογή μας ανήκει στη δεύτερη περίπτωση. Σε γενικό πλαίσιο, για να γίνει πιο αποδοτική η παροχή υπηρεσιών, εμπεριέχεται στη λειτουργία τους το αναπόσπαστο στοιχείο της κατανόησης της φυσικής γλώσσας (Natural Language Processing), γεγονός το οποίο όμως δε καθιστά τη λειτουργία τους απολύτως αληθοφανή και φυσική και αναμένεται ακόμα ένα χρονικό διάστημα και μία πιο εξειδικευμένη τεχνογνωσία για τη βελτιστοποίηση της λειτουργικότητας τους [16].

Με σκοπό την έμπνευση και τη σύλληψη μιας ιδέας για την εφαρμογή μας, προχωρήσαμε στην ανάλυση διαφόρων travel bots, από τα οποία άλλα μας εντυπωσίασαν με την απλότητα τους στο σύνολο των λειτουργιών τους, κι άλλα μας απογοήτευσαν, καθώς διέθεταν αρκετά ελαττώματα στην γενικότερη ροή τους. Παρακάτω εναποθέεται η λειτουργικότητα των κυριότερων bot, με την απαραίτητη επεξήγηση και στιγμιότυπα χρήσεως για την πλήρη κατανόηση αυτών [16].

#### 3.1.1 Sure

Η ροή του συγκεκριμένου bot ξεκινά από τη στιγμή που πατήσεις το κουμπί για να ξεκινήσεις τη συνομιλία και σε ενημερώνει για τις γενικότερες δυνατότητες του τις οποίες μπορείς να προσεγγίσεις πατώντας το κουμπί ‘Trending right now’.



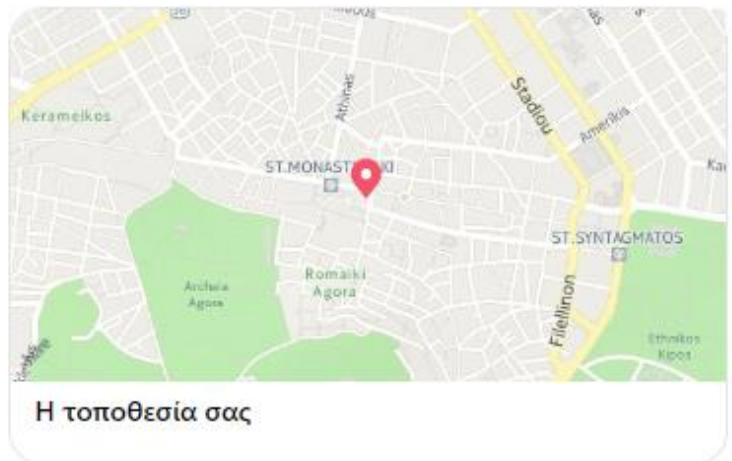
**Εικόνα 3 – Sure\_1 - Messenger**

Στη συνέχεια σου ζητά την άδεια να χρησιμοποιήσει την τοποθεσία που βρίσκεσαι κι αν δεν το επιθυμείς, μπορείς απλώς να πληκτρολογήσεις το μέρος, ώστε να πληροφορηθείς για τα πιο ‘trending’ μέρη.

Trending right now 🔥

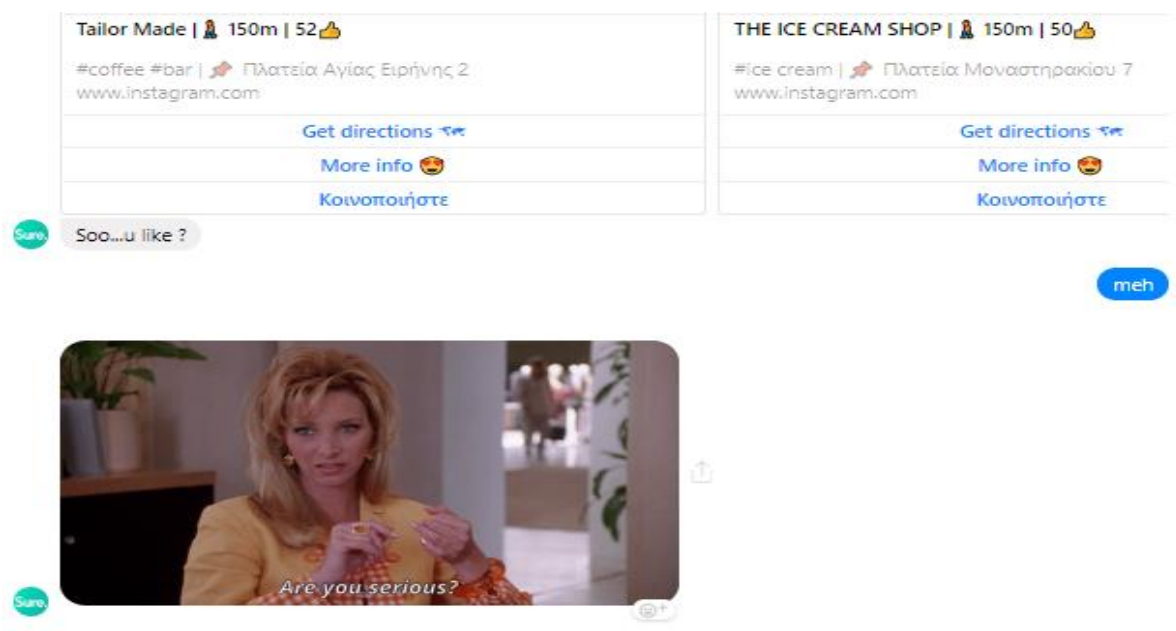
Sure.

Cool, send me a location using this button and I'll tell you what's trending right now 📍



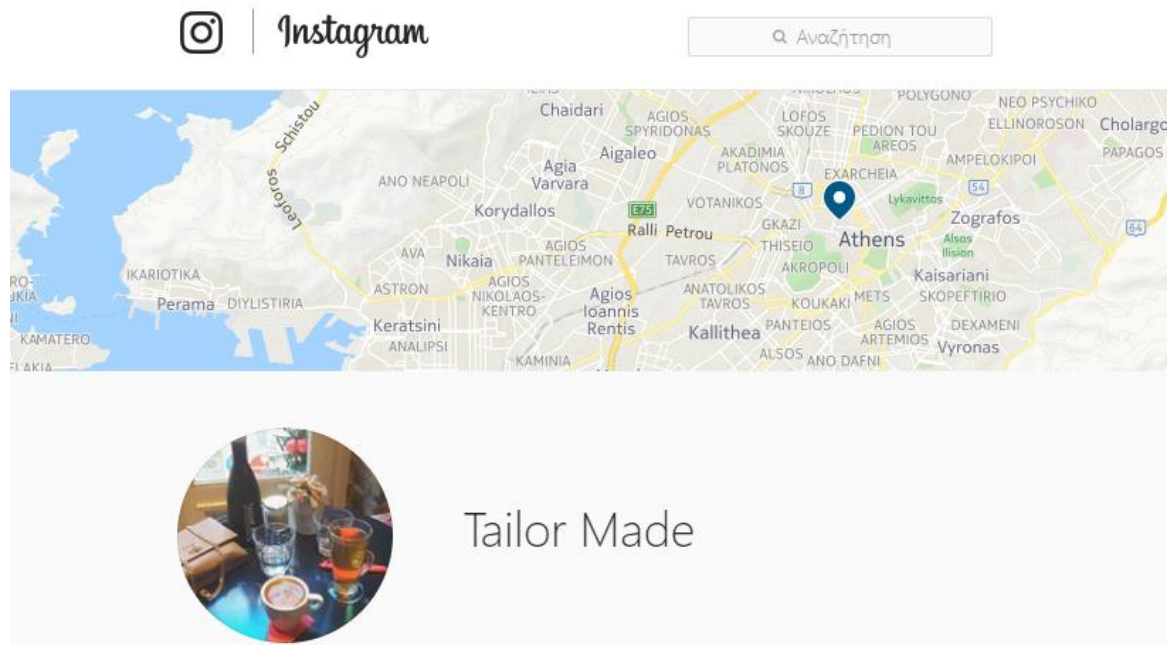
Εικόνα 4 – Sure\_2 - Messenger

Επιπρόσθετα σου δίνει τη δυνατότητα να πλοηγηθείς σε αυτά με βάση την τοποθεσία σου, είτε ακόμη να κλείσεις ένα Uber με στόχο την άφιξη στο μέρος που επιθυμείς. Τα αποτελέσματα στα παρέχει σε μορφή carousel και πηγή των δεδομένων είναι το Instagram.



Εικόνα 5 – Sure\_3 - Messenger

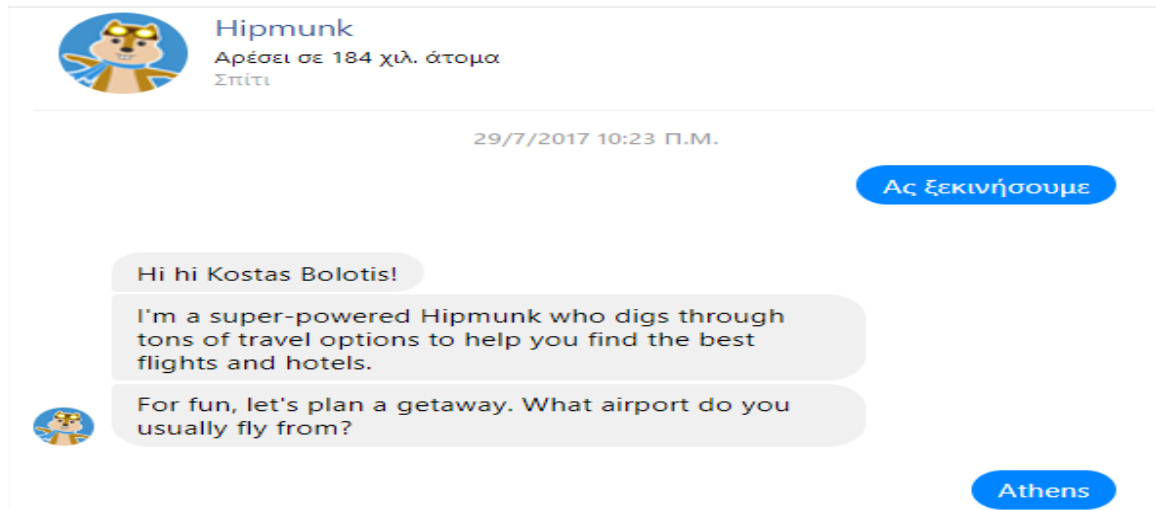
Τέλος ένα ενδιαφέρον χαρακτηριστικό του όπως και φαίνεται στην παραπάνω τελευταία εικόνα, είναι η ύπαρξη feedback, αν σου άρεσε δηλαδή η πρόταση του bot ή όχι και αναλόγως ανταποκρίνεται με ένα gif. Στο εν λόγω bot κρατήσαμε σαν θετικό στοιχείο την απεικόνιση των αποτελεσμάτων σε ροή με μορφή carousel και τις επιπρόσθετες λεπτομέρειες, όπως διάφορα στοιχεία για το κάθε μαγαζί και επίσης το κλικ να οδηγεί στο σύνδεσμο από τον οποίο παίρνουμε τα δεδομένα. Στην προκειμένη περίπτωση γίνεται ανακατεύθυνση στο Instagram.



Εικόνα 6 – Sure\_4 - Messenger

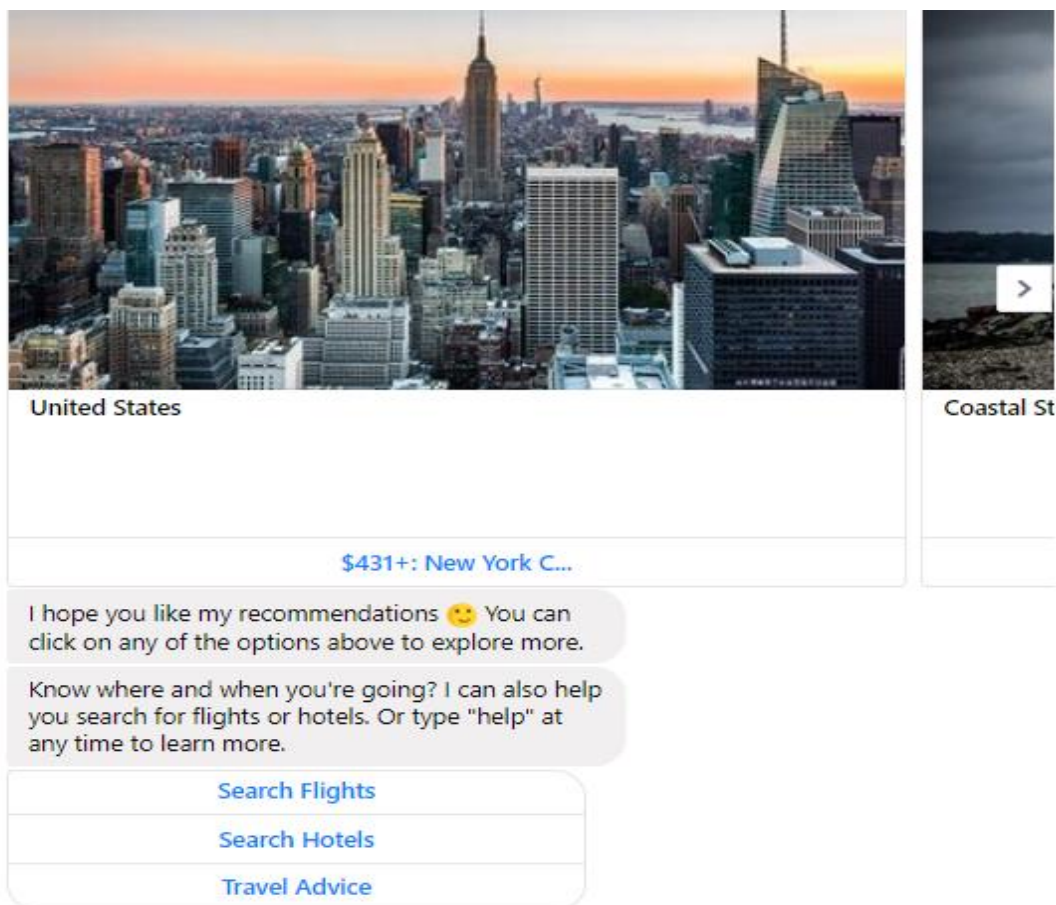
### 3.1.2 Hipmunk

Το Hipmunk, ξεκινά τη ροή του αναγνωρίζοντας το όνομα σου από την πλατφόρμα του Facebook και σε ρωτά για το που θα ήθελες να ταξιδέψεις και αναλόγως τι θα πληκτρολογήσεις, θα σου εμφανίσει κάποιες χώρες σαν ενδεχόμενους προορισμούς.



### Εικόνα 7 – Hipmunk\_1 - Messenger

Στη συνέχεια σου παρέχει τρεις διαφορετικές λειτουργίες, είτε να αναζητήσεις πτήσεις, είτε ξενοδοχεία είτε γενικότερες συμβουλές, όπως και φαίνεται στο παρακάτω στιγμιότυπο.




### Εικόνα 8 – Hipmunk\_2 - Messenger

Το συγκεκριμένο bot έχει αναπτύξει δικιά του βάση δεδομένων για οποιαδήποτε υπηρεσία, όποτε όταν θες για παράδειγμα να αναζητήσεις μία πτήση, αντλεί τα στοιχεία του από το δικό του χώρο,




τα αποστέλλει στον χρήστη κι αν ο χρήστης επιθυμεί να δει λεπτομέρειες, είτε να προχωρήσει στην αγορά ενός εισιτηρίου αρκεί να πατήσει το περιεχόμενο της κάρτας που επιθυμεί.

Departure: Fri, Sep 14, 2 stops

 **CDG** 11:20am → **DPS** 3:15pm +1d

Return: Sun, Sep 16, 2 stops

 **DPS** 4:15pm → **CDG** 7:30am +1d

---

**\$874**  
#1 Cheapest

---

[See Flight Details](#)

Εικόνα 9 – Hipmunk\_3 - Messenger

Από το συγκεκριμένο bot κρατήσαμε ως θετικό στοιχείο η ανακατεύθυνση σε έναν κεντρικό χώρο δεδομένων, λειτουργώντας σαν μεσάζοντας. (Βέβαια το Hipmunk είναι μεσάζων προς τον δικό του ιστότοπο).

### 3.1.3 MobyTrip

Το συγκεκριμένο bot εκπροσωπεί το εν λόγω γραφείο ταξιδιών. Καταρχάς η επικοινωνία ξεκινά πληκτρολογώντας το μέρος που θες να επισκεφθείς, επιβεβαιώνεις ότι είναι σωστό και στη συνέχεια σου δίνει μία επιλογή αριθμού ημερών από μία έως επτά ώστε να σου οργανώσει ένα ταξίδι. Ειδικότερα διαλέγεις και προσθέτεις οποιαδήποτε κατηγορία επιθυμείς, και προγραμματίζεις το ταξίδι των ονείρων σου, όπως και φαίνεται παρακάτω στο στιγμιότυπο.

Athens.Greece

Okay! We'll plan your trip for 'Athens, GR'.

Okay.


Alright, we're ready to roll. We'll flesh out your itinerary next and then give you an option to book hotels at the end. Let me know when you're ready to start.

Let's do it!

How many days would you like your trip to be? Give us any number between 1-7.


5

Great! All that's left to do is to add the types of things you want to do to your trip.  
Type "go" at any time to generate your itinerary.




Attraction

Add 'Attraction'.



Museums

Add 'Museums'.



Cultural

Add 'Museums'.

Museums' feature added to trip!

## Εικόνα 10 – MobyTrip\_1 - Messenger

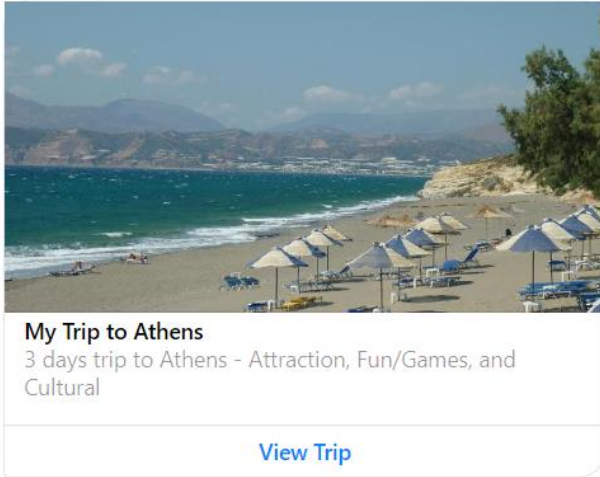
Ωστόσο παρατηρήσαμε ένα βασικό μειονέκτημα, το οποίο κατά τη γνώμη μας είναι αρκετά σημαντικό. Εάν προσθέσεις παραπάνω από μία δραστηριότητες, το λογισμικό δε μπορεί να τις επεξεργαστεί και έτσι ο χρήστης θα πρέπει να περιοριστεί απλώς σε μία.

Το επόμενο βήμα περιλαμβάνει την επιλογή να δεις το ταξίδι σου, ανακατευθύνοντάς σε στον ιστότοπο τους, σε περίπτωση που θες να κλείσεις κάποιο ξενοδοχείο αλλά και να δεις το πρόγραμμα που σου όρισε αναλόγως τη δραστηριότητα που επέλεξες.



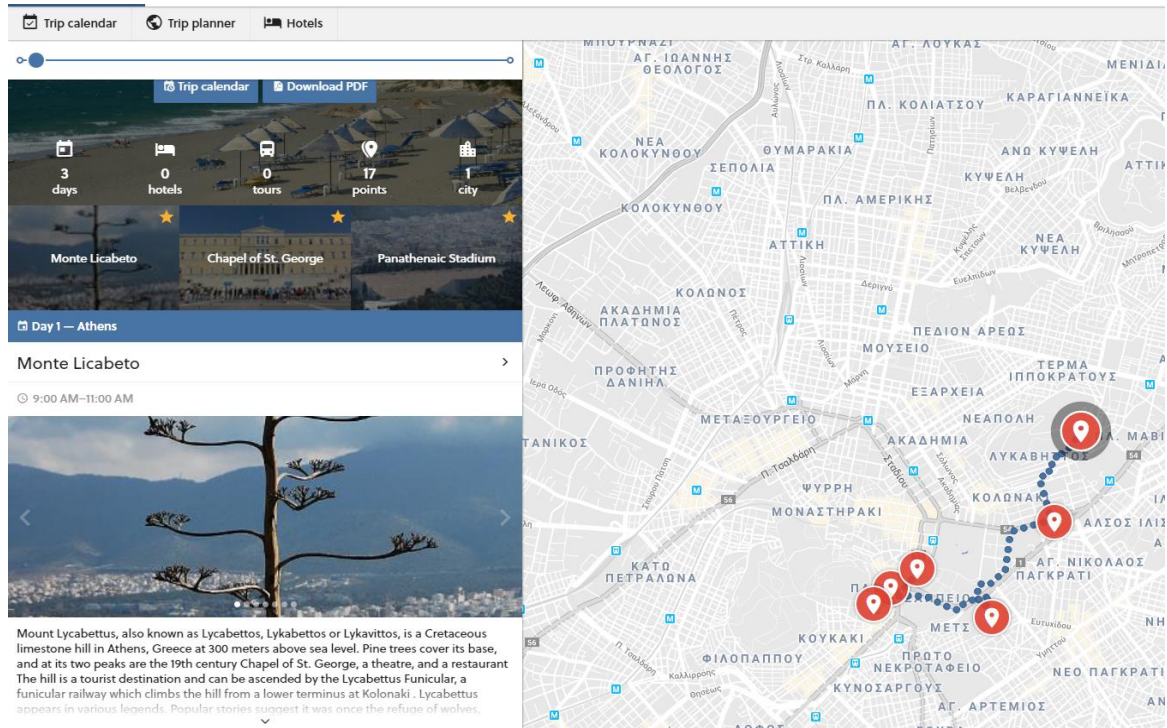
'Attraction' feature added to trip!

Go to trip!



### Εικόνα 11 – MobyTrip\_2 - Messenger

Τέλος στο site τους, όπως και αναφέραμε προηγουμένως, μπορείς να δεις αναλυτικά το τι σου έχει ορίσει να επισκεφθείς και φυσικά το που βρίσκεται το καθετί, συνδυάζοντας ένα όμορφο γραφικό περιβάλλον. Τελευταίο αλλά όχι λιγότερο σημαντικό είναι ότι οι κρατήσεις των ξενοδοχείων γίνονται μέσα από τη πλατφόρμα του booking.com. Το θετικό που κρατήσαμε για την εφαρμογή μας, είναι τα διάφορα είδη τουρισμού σαν ξεχωριστές κατηγορίες.

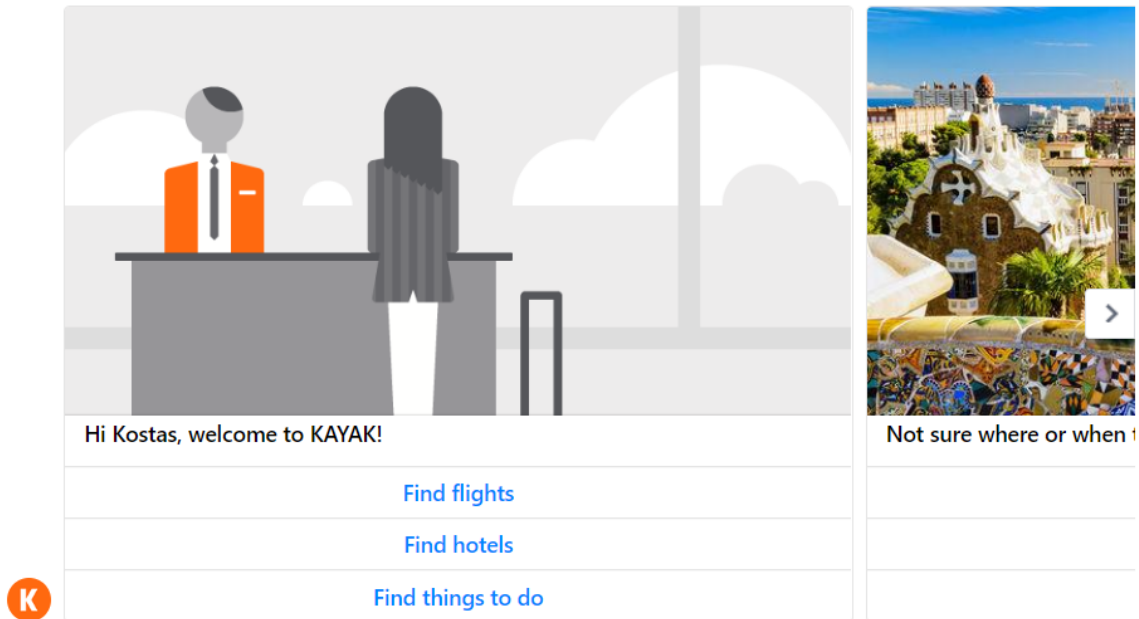


Εικόνα 12 – MobyTrip\_3 - Messenger

### 3.1.4 Kayak

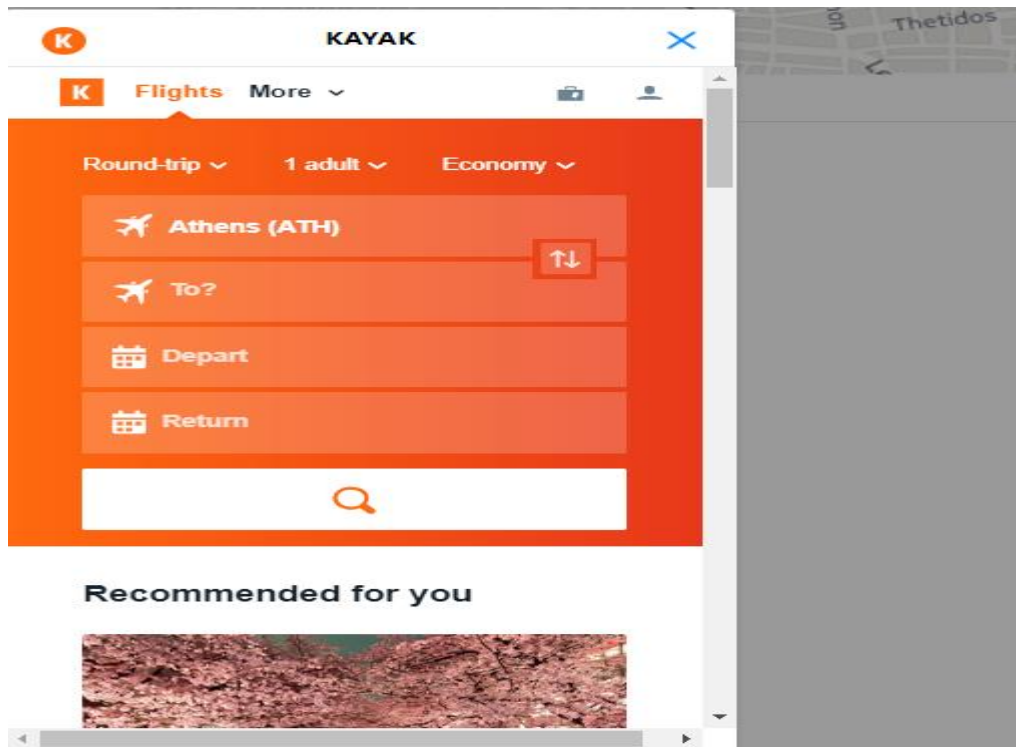
Το Kayak είναι ευρέως γνωστό στον χώρο του τουρισμού και ξεχωρίζει σε σχέση με τα υπόλοιπα που έχουν ήδη αναλυθεί, καθώς προσφέρει στον μέσο χρήστη όλες τις απαραίτητες ανάγκες προς κάλυψη: ξενοδοχεία, πτήσεις, ενοικιάσεις αυτοκινήτων μέχρι και τη δυνατότητα να μπορείς να εντοπίζεις που βρίσκονται οι πτήσεις. Η επικοινωνία ξεκινά όπως και συνηθίζεται με ένα κουμπί και σε ενημερώνει άμεσα με τις λειτουργίες που παρέχει.

Ας ξεκινήσουμε



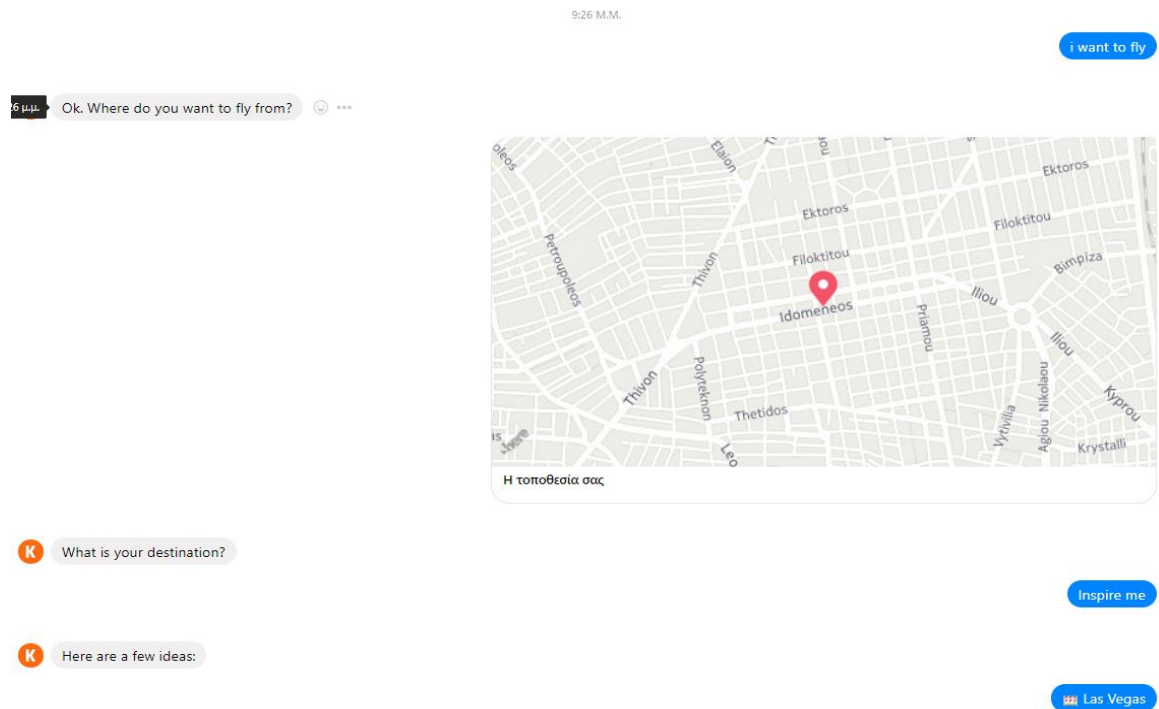
Εικόνα 13 – Kayak\_1 - Messenger

Ας προχωρήσουμε σε μία λειτουργία και ειδικότερα στο 'Find flights'. Μόλις πατήσεις, εμφανίζεται ένα popup παράθυρο το οποίο ουσιαστικά συνιστά ένα αντίστοιχο user interface με τους περισσότερους ιστότοπους που εξειδικεύονται στον χώρο του τουρισμού. Αν και ίσως αποκλίνει λίγο από τη συνήθη ροή των υπόλοιπων bot, είναι κάτι που είναι πολύ ωραίο σε αισθητικό πλαίσιο για το χρήστη.

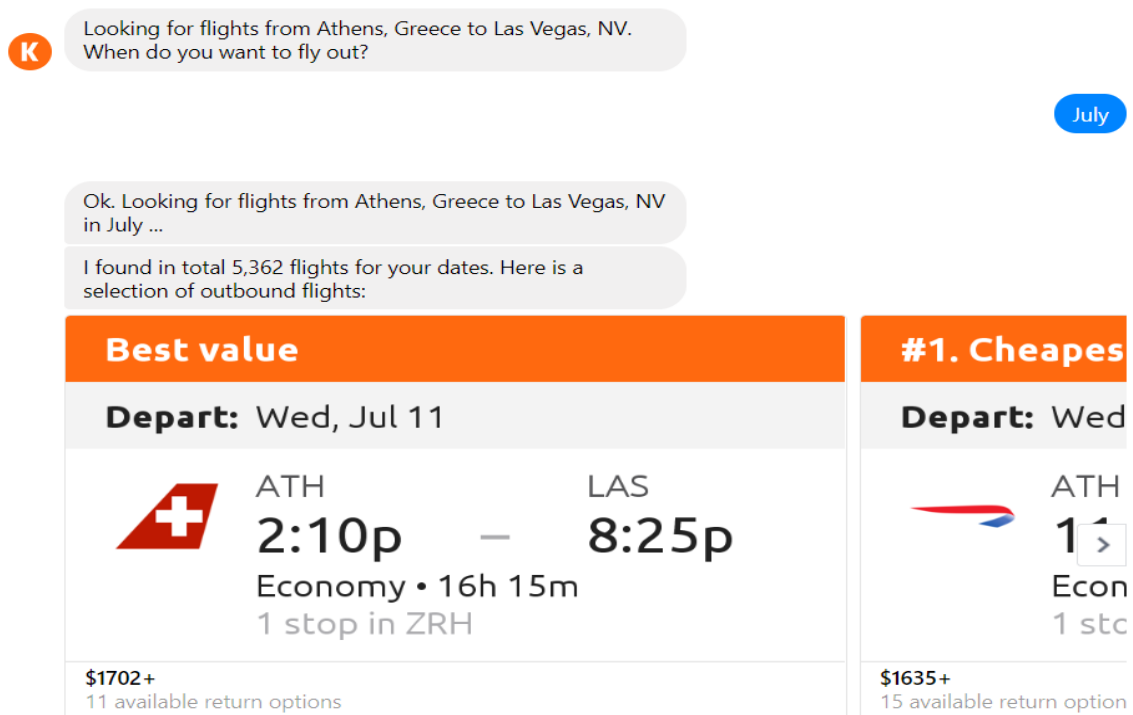


Εικόνα 14 – Kayak\_2 - Messenger

Επιπρόσθετα με το παραπάνω αναδυόμενο παράθυρο, ο χρήστης μπορεί να αποστείλει τοποθεσία που βρίσκεται τον προορισμό που επιθυμεί, να πληκτρολογήσει τον μήνα και να λάβει τα αποτελέσματα. Μία όμορφη λεπτομέρεια που συναντάται είναι η λειτουργία 'Inspire me', που με έναν δικό τους ανεπτυγμένο αλγόριθμο σου προσφέρει ιδέες για μέρη που θα ήθελες να πας.



**Εικόνα 15 – Kayak\_3 - Messenger**



**Εικόνα 16 – Kayak\_4 - Messenger**

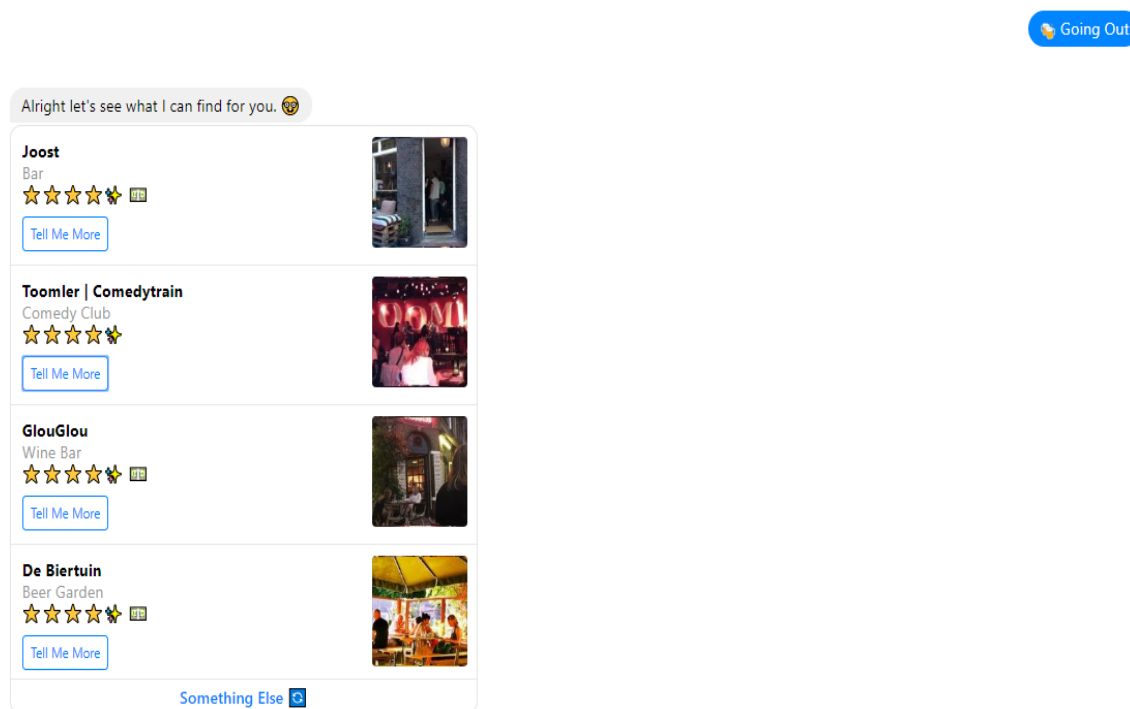
Όπως φαίνεται παραπάνω υποστηρίζει και NLP, μιας και συνειδητοποιεί όταν γράφω ‘I want to fly’, ότι πρόκειται για πτήσεις. Εάν αναλύσουμε εξονυχιστικά τη λειτουργία του συγκεκριμένου



bot, θα παρατηρούσαμε μόνο ένα μειονέκτημα. Στην αναζήτηση, κράτηση πτήσης που αναλύσαμε στο παραπάνω χωρίο, σε προτρέπει να πληκτρολογήσεις το μήνα, χωρίς εξειδικευμένη ημερομηνία, γεγονός όμως που δεν αλλάζει την εικόνα μας και φυσικά το κύρος που διαθέτει, προσφέροντας όσα θέλει ένας χρήστης του. Ως θετικό στοιχείο κρατήσαμε την κατανόηση της φυσικής γλώσσας και την ταχεία και άμεση διαδραστικότητα με το χρήστη.

### 3.1.5 Gooster

Η συνομιλία ξεκινά με το κουμπί εκκίνησης και σου παρέχει τη βασική ροή λειτουργίας του, όπου πρώτο βήμα είναι η πληκτρολόγηση της πόλης. Στη συνέχεια, σου δίνεται μία ειδική κατηγοριοποίηση για να βρεις αυτό που επιθυμείς. Για παράδειγμα μπορείς να γράψεις ‘going out’ και να σου εμφανίσει επιλογές όπως ένα μπαρ, μία μπυραρία, είτε γενικά οτιδήποτε που το κατηγοριοποιεί στον τομέα της διασκέδασης. Η ανάμειξη αρκετών υποκατηγοριών σε μία κατηγορία, κατά τη γνώμη μας μπορεί να προβληματίσει το χρήστη εάν θέλει να αναζητήσει κάτι πιο ειδικό, αλλά έτσι όπως το προβάλλει, συνιστά χρήσιμο. Παρακάτω φαίνεται ένα στιγμιότυπο χρήσης, έχοντας πρώτα τοποθετήσει ως πόλη το Amsterdam.



Εικόνα 17 -Gooster\_1 - Messenger

Επιπρόσθετα παρέχει την επιλογή να αναζητήσεις σε ολόκληρη την πόλη είτε κοντά σε εμένα, σε μία πιο εξειδικευμένη χωρική εμβέλεια, όπως και φαίνεται παρακάτω.



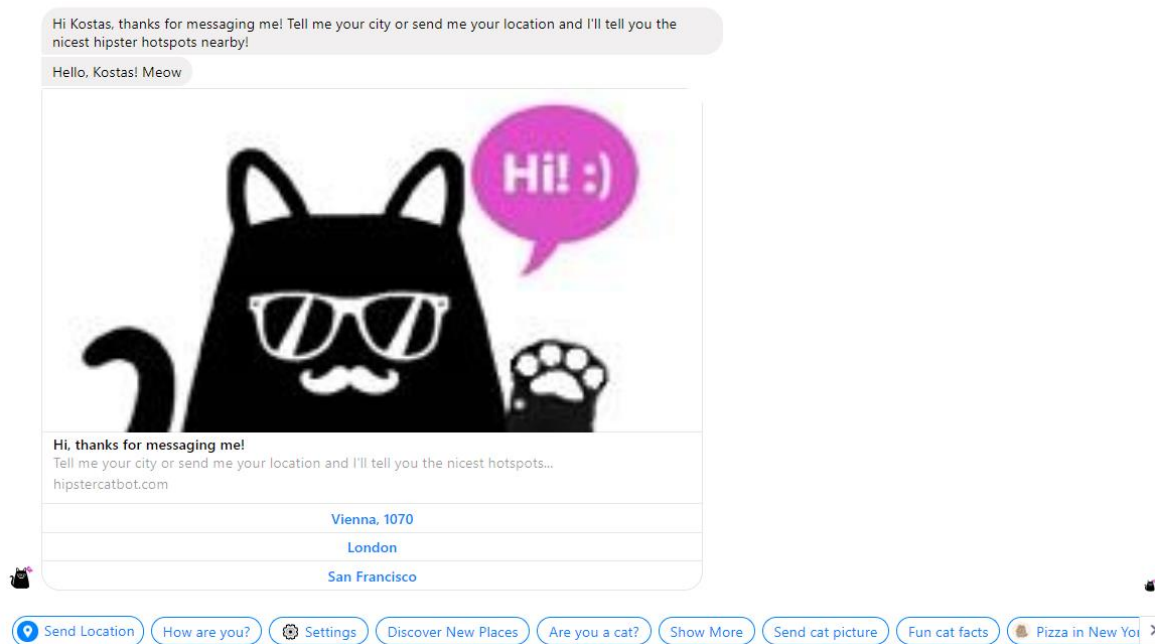


### Εικόνα 18 – Gooster\_2 - Messenger

Τέλος περιλαμβάνεται στη λειτουργία του και η επιλογή top picks, στοιχείο το οποίο μας κέντρισε το ενδιαφέρον, καθώς θα μπορούσε να αναπτυχθεί μέσω της ανάδρασης με τους χρήστες μας. Τέλος, ένα γενικότερο μειονέκτημα ήταν η εμφάνιση, καθώς προσέδιδε πολλή πληροφορία στον χρήστη με περιττές λεπτομέρειες.

#### 3.1.6 Mica, the Hipster Cat Bot

Στέλνοντας ένα μήνυμα χαιρετισμού, το bot σου απαντά και σε παροτρύνει να πληκτρολογήσεις την τοποθεσία σου είτε να επιλέξεις μία από το σύνολο των λειτουργιών του με ένα όμορφο κι ευχάριστο μήνυμα, το οποίο προσδίδει στο χρήστη μία διαφορετική διάθεση εξαρχής.



**Εικόνα 19 - Mica\_1 - Messenger**

Ας πάρουμε για παράδειγμα το 'Send Location'. Επειδή η τοποθεσία μου δε βρίσκεται σε σχετικά κεντρικό μέρος ας δικαιολογήσουμε σε μερικό βαθμό που δε μου επιστρέφει κάποιο μέρος για να επισκεφθώ, όπως και φαίνεται παρακάτω.



Sorry, I don't know any nice venues close by. Please try a different location!

Maybe try a city name or...

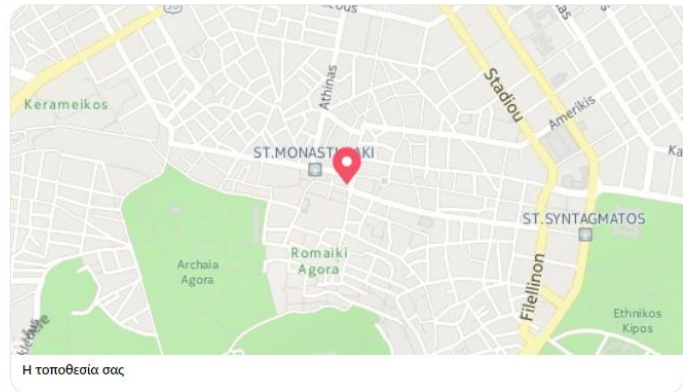
**Εικόνα 20 – Mica\_2 - Messenger**

Αντ' αυτού, σου στέλνει ένα μη σχετικό μήνυμα, που όμως παρουσιάζει ενδιαφέρον καθώς προσεγγίζει ένα ικανοποιητικό επίπεδο τεχνητής νοημοσύνης, αποκλίνοντας όμως από τον αληθινό σκοπό του.

Did you already watch 'Her'?

[Check out this movie...](#)

Ωστόσο, στις περισσότερες τοποθεσίες υπάρχουν αποτελέσματα και μάλιστα απεικονίζονται με μορφή carousel, με όλες τις απαραίτητες λεπτομέρειες. Όπως απεικονίζεται και στο επερχόμενο στιγμιότυπο, τα δεδομένα λαμβάνονται από το TripAdvisor, συνιστώντας έτσι κι αυτό το bot με τη σειρά του έναν μεσάζοντα που διαχειρίζεται και προβάλλει στον χρήστη γενικότερες πληροφορίες.



Let me suggest these awesome hip venues:



**Barrett [Café]**  
Open until 4:00 AM - Πρωτογένους 11, 105 54, Αθήνα  
igx.4sqi.net

[Show details](#)



**Drunk Sinatra [Bar]**  
Open until 4:00 AM - Θησείως 16, 105 62, Αθήνα  
igx.4sqi.net

[Show on TripAdvisor!](#)



**Booze Cooperativa [Bar]**  
Open until 3:00 AM - Κολ  
igx.4sqi.net

## Εικόνα 21 - Mica\_3 - Messenger

Όπως και αναφέραμε παραπάνω μπορείς να δεις κάποιες πιο εξειδικευμένες λειτουργίες όπως πίτσα στη Νέα Υόρκη.

Let me recommend these hip places in Brooklyn:



**Roberta's Pizza [Pizza Place]**  
Open until Midnight - 261 Moore St, 11206, Brooklyn  
igx.4sqi.net



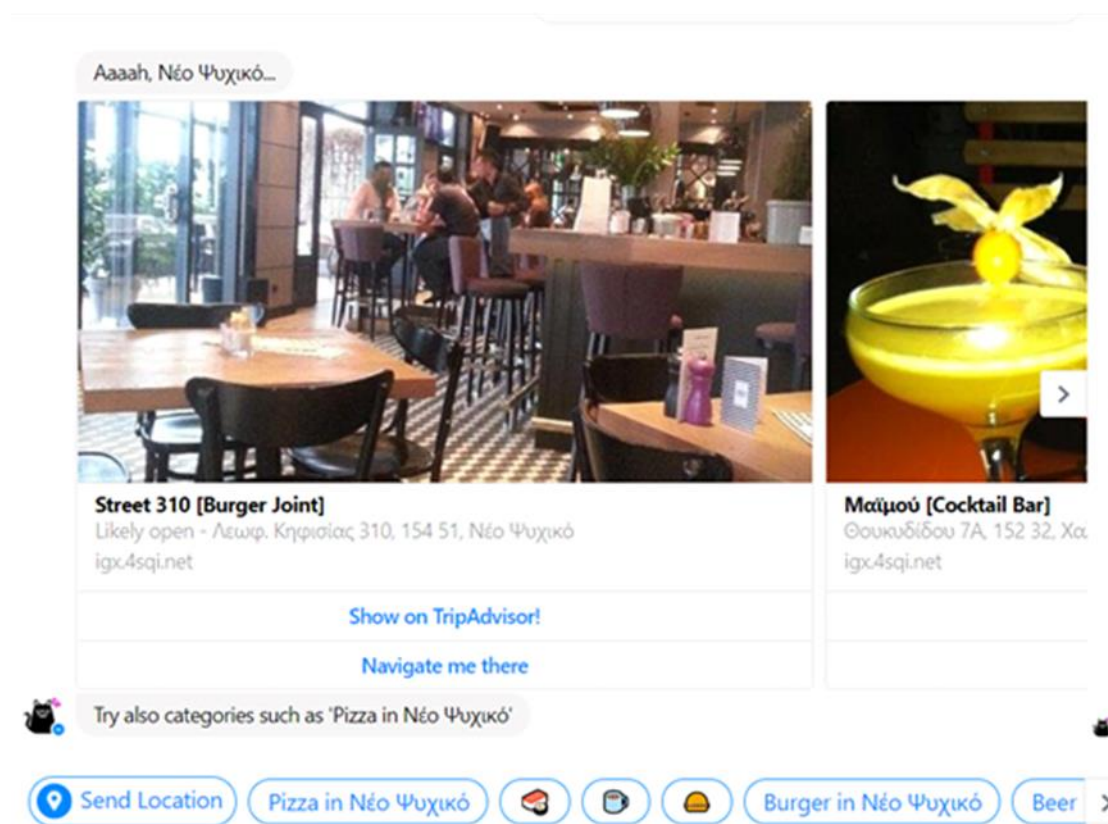
**Joe's Pizza [Pizza Place]**  
Open until 4:00 AM - 7 Carmine St, 10014, New York  
igx.4sqi.net



**Lazzara's Pizza [Italian Re]**  
Open until 9:00 PM - 221  
igx.4sqi.net

## Εικόνα 22 – Mica\_4 - Messenger

Αξίζει να σημειωθεί επίσης, ότι όταν επιλέξεις την τοποθεσία σου, από κάτω σου προσφέρει κι άλλες δυνατότητες οι οποίες μέχρι τώρα δεν είχαν φανερωθεί στο περιβάλλον του χρήστη (Pizza, Burger, Beer).

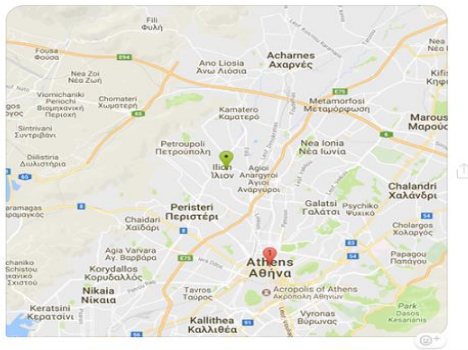


Εικόνα 23 – Mica\_5 - Messenger

Ως θετικά στοιχεία κρατάμε την ταχύτητα στην αναζήτηση και τη γενικότερη ευχρηστία, καθώς με την επιλογή του κουμπιού 'Send Location' απεικονίζονται με βάση το λογισμικό τα πιο ωραία μέρη που πρέπει να επισκεφθεί κάποιος.

### 3.1.7 Nearby Places Bot

Το nearby places bot με την πρώτη οπτική μπορεί πολύ εύκολα να θεωρηθεί πολύ απλοϊκό και ότι δεν πρόκειται να προσφέρει στον χρήστη την πληροφορία που χρειάζεται. Ο διάλογος με το λογισμικό αρχίζει με το κουμπί 'Ας ξεκινήσουμε' και αμέσως σε προτρέπει να στείλεις την τοποθεσία σου, και μόνο αυτή, καθώς δε σου δίνεται η δυνατότητα να πληκτρολογήσεις το μέρος που θέλεις. Αντ' αυτού όταν για παράδειγμα πατήσεις 'Athens' σου στέλνει το πόσο απέχει από εσένα.



(1) Athens  
5.6 km

### Εικόνα 24 – Nearby\_1 - Messenger

Ωστόσο εάν συμφωνήσεις να αποστέλλεις την τοποθεσία σου, χωρίς να διαθέτεις κάποιο προβληματισμό, η λειτουργικότητα της εφαρμογής είναι εξαιρετικά περιεκτική. Ειδικότερα σου προσφέρει λειτουργίες που τα περισσότερα bot που αναλύσαμε μέχρι τώρα δε διέθεταν, όπως Atm, φαρμακεία, πάρκα.



Please, select place type or search by the text message:

ATMs

Pharmacies

Lodging

Food

Stores

Hospitals

Restaurants

Police

Night clubs

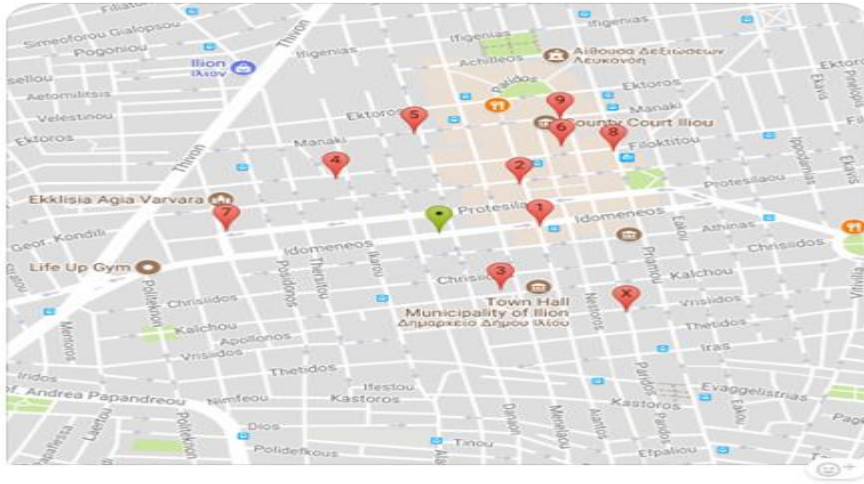
Parks

### Εικόνα 25 – Nearby\_2 - Messenger

Ας επιλέξουμε για παράδειγμα τα φαρμακεία ώστε να δούμε τη ροή που πραγματοποιείται.



Please, select place type or search by the text message:



**(1) Siafaka Katerina**

142 m

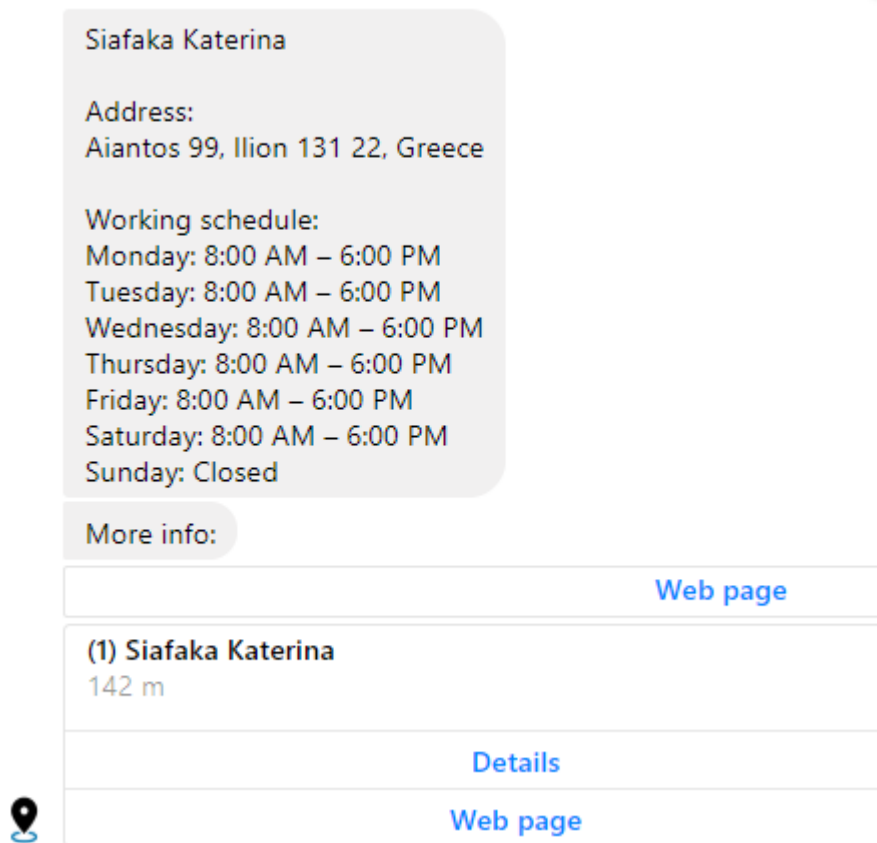


[Details](#)

Εικόνα 26 – Nearby\_3 - Messenger

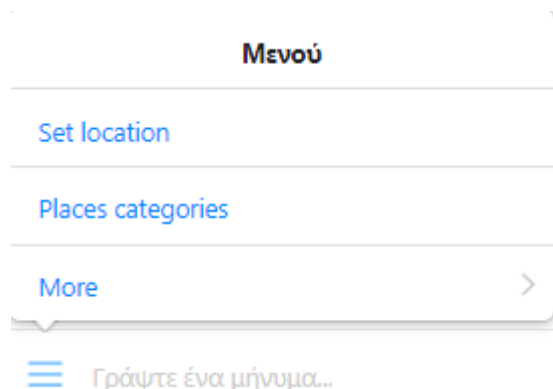
Όπως παρατηρείται μου δείχνει το πόσο απέχει από την τοποθεσία που έχω αποστείλει, και επίσης παρέχεται και η δυνατότητα να επιλέξω παραπάνω λεπτομέρειες, οι οποίες και καταγράφονται αναλυτικά στην παρακάτω εικόνα.



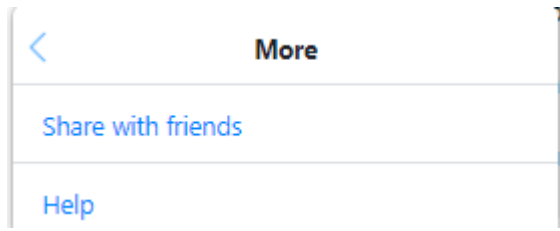


Εικόνα 27 – Nearby\_4 - Messenger

Επιπρόσθετα, εντύπωση μας έκανε και το ‘hamburger menu’ σε σχέση με τα υπόλοιπα, επειδή ήταν αρκετά κατανοητό και εύχρηστο. Περιλαμβάνει την ανάθεση της τοποθεσίας, ώστε να σου βγάλει τις λειτουργίες και επίσης η καρτέλα ‘Places categories’ σου δείχνει τις κατηγορίες χωρίς να χρειάζεται για την ώρα η τοποθεσία.

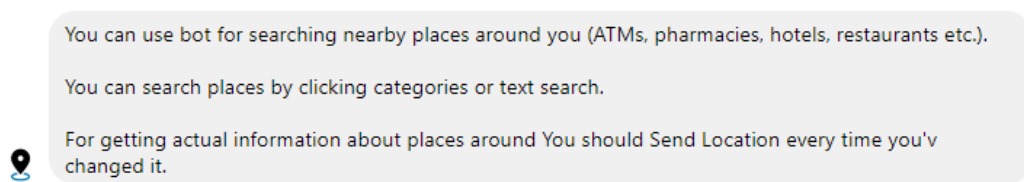


Εικόνα 28 – Nearby\_5 - Messenger



**Εικόνα 29 – Nearby\_6 - Messenger**

Στην καρτέλα More, βρίσκεται το απαραίτητο Help, το οποίο επιστρέφει το παρακάτω αποτέλεσμα για να αναδείξει τη βασική λειτουργικότητα του bot.



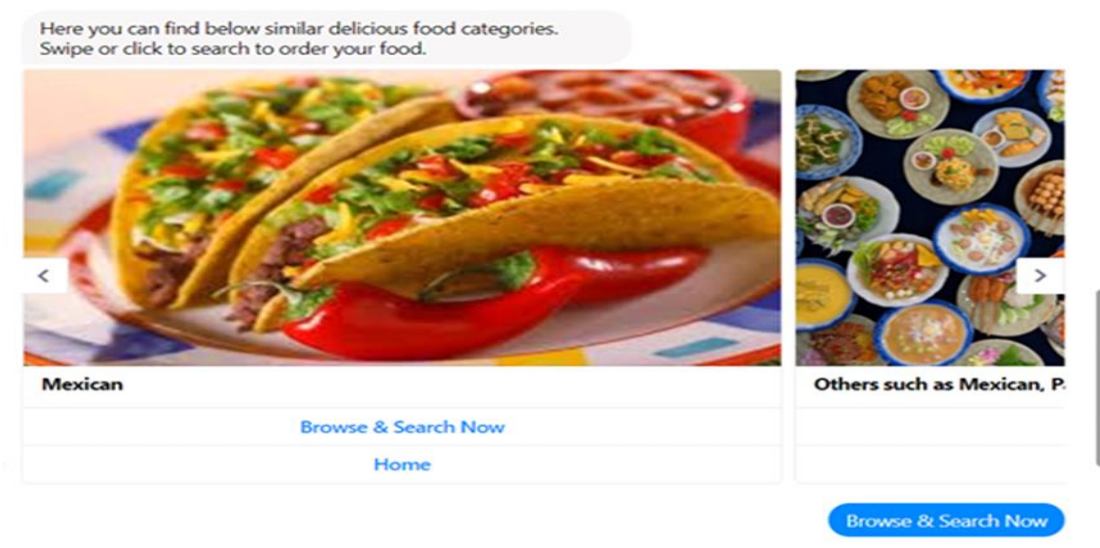
**Εικόνα 30 – Nearby\_7 - Messenger**

Όντας ένα bot που προσφέρει σχεδόν τα πάντα, εκτός από ξενοδοχεία και πτήσεις, μας δημιούργησε τελικά πολύ καλή εντύπωση και κρατάμε ως θετικό στοιχείο την ύπαρξη του πλήρως βοηθητικού 'hamburger menu' μαζί με την ύπαρξη ενός κουμπιού βοήθειας, το οποίο θα δίνει ένα βασικό ερέθισμα για το πως θα χρησιμοποιηθεί το bot, σε περίπτωση που ο χρήστης δε ξέρει πως να ανταποκριθεί.

### 3.1.8 1800 Order

Ένα bot το οποίο σου παρέχει τη δυνατότητα ανάλογα με την κούζίνα που επιθυμείς να φας (μεξικάνικη, ινδική, ...) τη δεδομένη χρονική στιγμή. Αρχικά όπως φαίνεται παρακάτω στις εικόνες σου προσφέρει τη δυνατότητα να επιλέξεις την κούζίνα που επιθυμείς και στη συνέχεια ζητώντας την τοποθεσία σου, κάνει redirect στο google maps όπου σου δείχνει τα κοντινότερα εστιατόρια της συγκεκριμένης επιθυμίας σου. Πλεόνεκτημα του συγκεκριμένου bot είναι ότι παρέχει πληροφορίες σχεδόν για όλες τις κουζίνες του κόσμου.



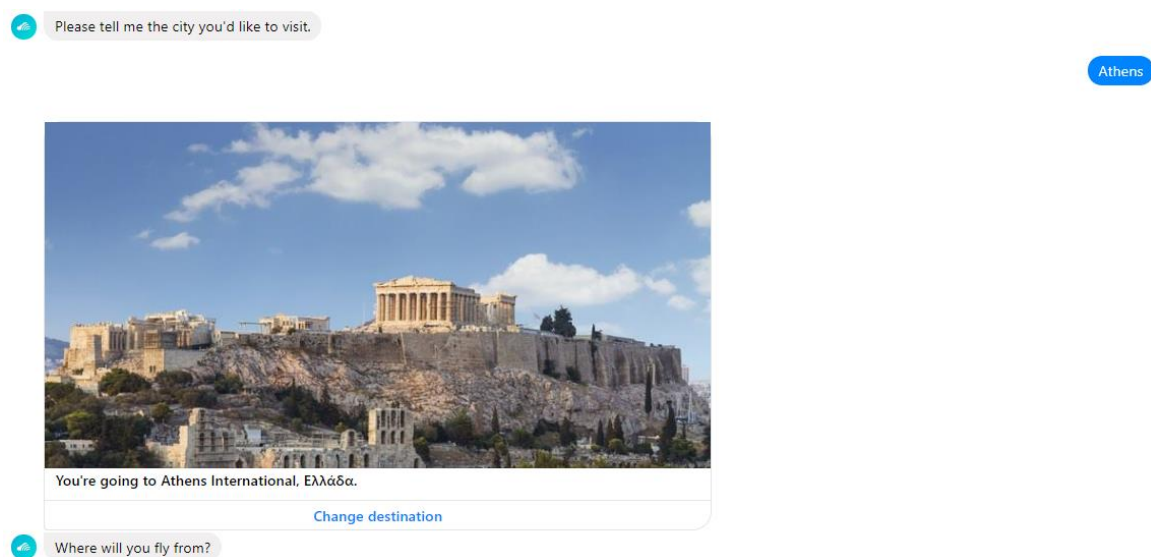


**Εικόνα 31 – 1800 - Messenger**

Το θετικό στοιχείο που κρατήσαμε είναι η ποικιλότητα κατηγοριοποίηση των φαγητών, στοιχείο που εμπεριέχεται φυσικά και στη δική μας εφαρμογή.

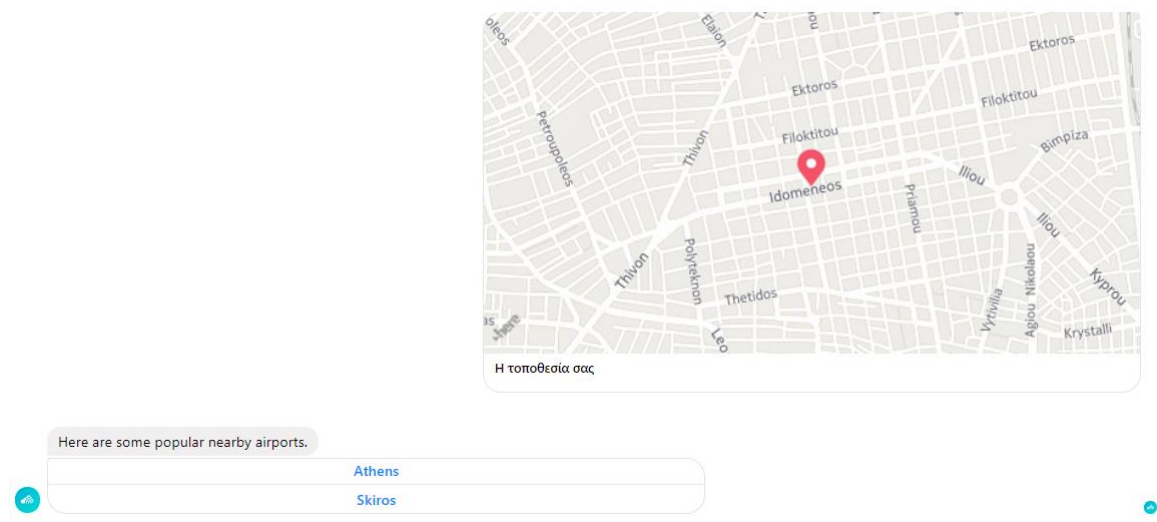
### 3.1.9 SkyScanner

Πρόκειται για το bot που έχει αναπτύξει η διάσημη εταιρεία. Στην αρχή της συνομιλίας ζητείται από τον χρήστη η πόλη στην οποία επιθυμεί να ταξιδέψει και φυσικά το επόμενο βήμα αποτελεί η αποστολή( όχι πληκτρολόγηση) της τωρινής τοποθεσίας του χρήστη ώστε να αναζητηθεί το κοντινότερο αεροδρόμιο.



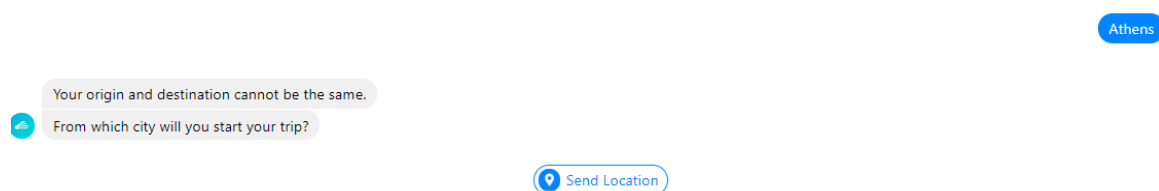
**Εικόνα 32 – SkyScanner\_1 - Messenger**

Παρατηρούμε ότι στα αποτελέσματα απεικονίζεται προφανώς το αεροδρόμιο της Αθήνας, αλλά λογικά με βάση κάποιο κριτήριο και το αεροδρόμιο της Σκύρου. Το γεγονός αυτό δε μας προβληματίζει, αφού μπορούμε να επιλέξουμε το αεροδρόμιο που επιθυμούμε.



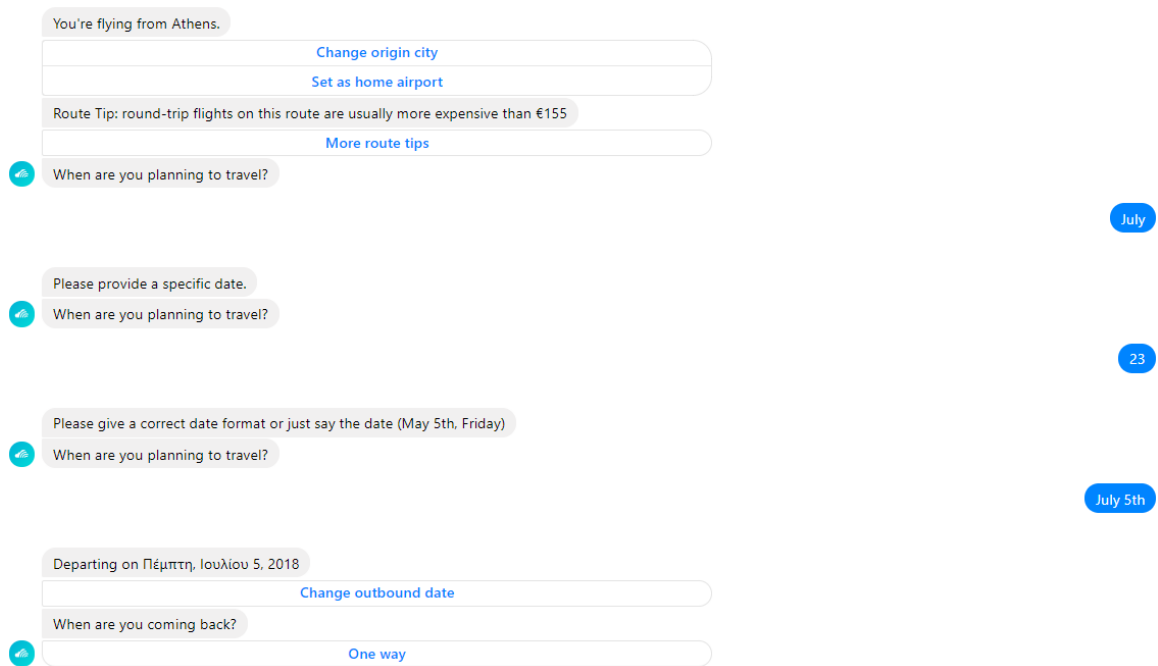
**Εικόνα 33 – SkyScanner\_2 - Messenger**

Επιπρόσθετα όπως ήταν αναμενόμενο εάν επιλέξω την ίδια τοποθεσία για προορισμό και σημείο άφιξης ανταποκρίνεται, και μου ζητά καινούριο σημείο εκκίνησης.



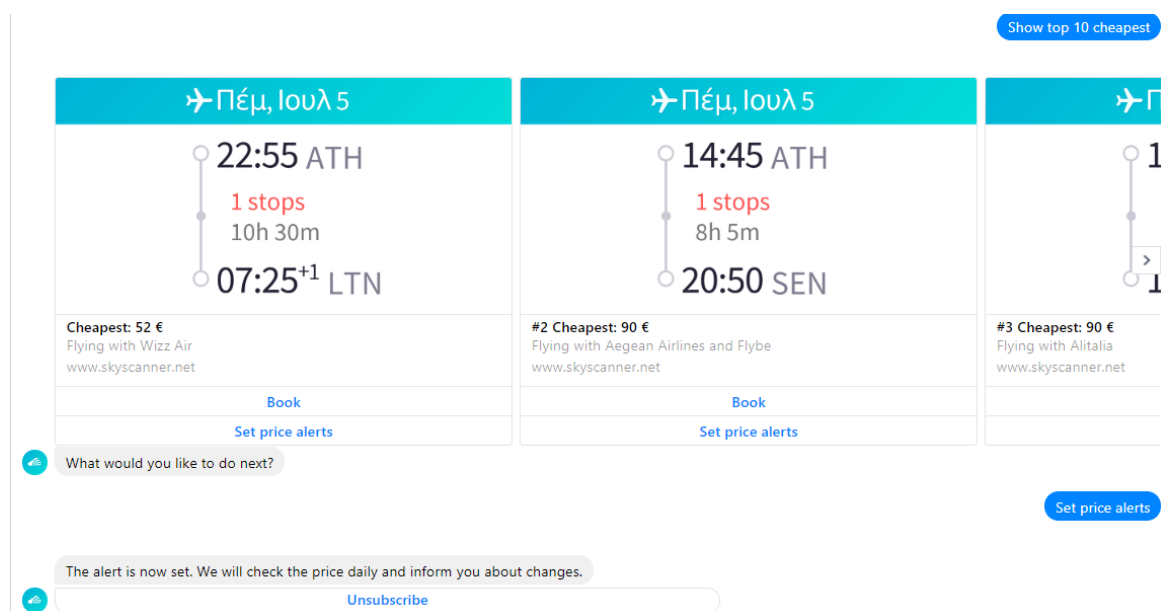
**Εικόνα 34 – SkyScanner\_3 - Messenger**

Επιλέγοντας τελικά σωστά κριτήρια, με σημείο αναχώρησης το Λονδίνο και άφιξης την Αθήνα, προχωρά σε ερωτήσεις για το πότε ακριβώς ταξιδεύω, κι όταν λανθασμένα του απαντώ σε γενικευμένο πλαίσιο μόνο τον μήνα συνεχίζει να ανταποκρίνεται μέχρι να αποκτήσει την ολική ημερομηνία. Στη συνέχεια για να ολοκληρωθεί η ροή και να εμφανιστούν τα αποτελέσματα ο χρήστης καλείται να ορίσει εάν είναι μονή διαδρομή ή με επιστροφή. Εάν φυσικά είναι με επιστροφή, πρέπει να προσδιορίσει και τη δεύτερη ημερομηνία.



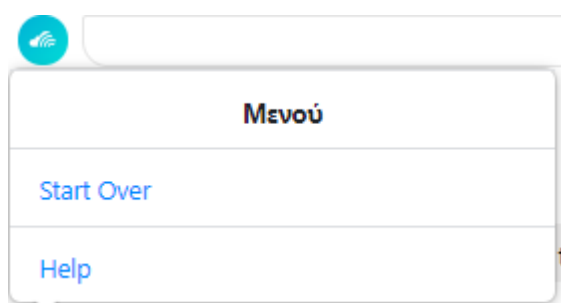
### Εικόνα 35 -SkyScanner\_4 - Messenger

Τα αποτελέσματα φαίνονται στο παρακάτω στιγμιότυπο χρήσης, με τη δυνατότητα να κλείσεις την πτήση που επιθυμείς και να ανακατευθυνθείς στον αυθεντικό ιστότοπο της εταιρείας, είτε να δεις πληροφοριακά στο περιβάλλον του messenger τις 10 πιο φθηνές πτήσεις. Τέλος πολύ σημαντικό είναι, ότι μπορεί ο χρήστης να ορίσει πατώντας το 'Set price alerts' την καθημερινή του ενημέρωση σχετικά με τυχούσες αλλαγές για τη συγκεκριμένη πτήση και προφανώς σε περίπτωση που πλέον δεν το επιθυμεί μπορεί να πατήσει unsubscribe.



### Εικόνα 36 – SkyScanner\_5 - Messenger

Ακόμη, ενδιαφέρον αποτελεί και το ‘hamburger menu’ του συγκεκριμένου bot, καθώς προσφέρει στον χρήστη να ξεκινήσει από την αρχή, σε περίπτωση που νοιώθει μπερδεμένος και δε μπορεί να συνεχίσει τη συνομιλία στο σημείο που βρίσκεται.

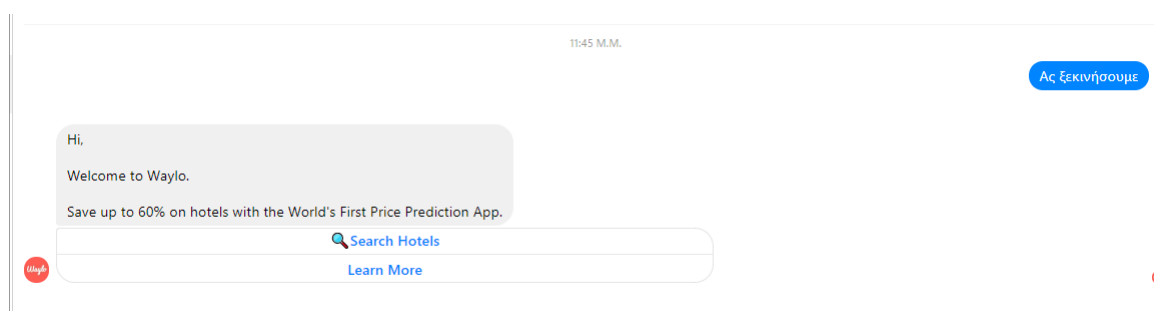


**Εικόνα 37 – SkyScanner\_6 - Messenger**

Αναλύοντας το bot της Skyscanner, είδαμε το πως είναι ένα πλήρως λειτουργικό bot χωρίς ατέλειες σε ένα συγκεκριμένο τομέα των ταξιδιών, τις πτήσεις. Ως θετικό στοιχείο διατηρούμε την αποστολή στον χρήστη της ακριβής πληροφορίας που χρειάζεται και τον απασχολεί, τις φθηνότερες δηλαδή πτήσεις και επίσης τη λειτουργία Start Over, που κρίνουμε απαραίτητη για οποιοδήποτε αυτοματοποιημένο λογισμικό τεχνητής νοημοσύνης.

### 3.1.10 Waylo: Hotel Price Prediction

Το συγκεκριμένο bot όπως και το SkyScanner, ανήκει σε μία εξειδικευμένη κατηγορία στον χώρο των ταξιδιών και συγκεκριμένα στον τομέα των ξενοδοχείων. Υπόσχεται στον χρήστη ότι χρησιμοποιώντας το, θα αποταμιεύσει ένα ποσοστό εξόδων που θα προσεγγίσει το εξήντα τοις εκατό. Η συνομιλία ξεκινά με το παρακάτω μήνυμα.



**Εικόνα 38 – Waylo\_1 - Messenger**

Στη συνέχεια πληροφορεί τον χρήστη ότι πρέπει να αποστείλει την πόλη και την ημερομηνία σε ένα ανάμεικτο μήνυμα με κάποια συγκεκριμένη μορφή ώστε να γίνει κατανοητό από το αυτοματοποιημένο λογισμικό. Επόμενο βήμα είναι η καταχώρηση του πλήθους των ατόμων που θα διαμείνουν και τέλος εάν υπάρχει κάποια ιδιαίτερη προτίμηση για το που ακριβώς μες στην πόλη.

The screenshot shows a chat interface with the Waylo logo. The user has entered a message: "Type the city you are traveling to, Kostas, and the dates e.g. Paris 5-7 July". The system responds with a blue button: "New York 6 July". The user then sends: "Please confirm the number of guests". The system responds with a blue button: "1 adult". The user sends: "Where in NEW YORK, USA would you like to stay?". The system responds with a blue button: "No preference". A summary message appears: "Looks like you're traveling to NEW YORK, USA from Jul 6th to Jul 7th 2018. Getting the best deals for 1 guests. Please stand by while we get you the best hotel deals".

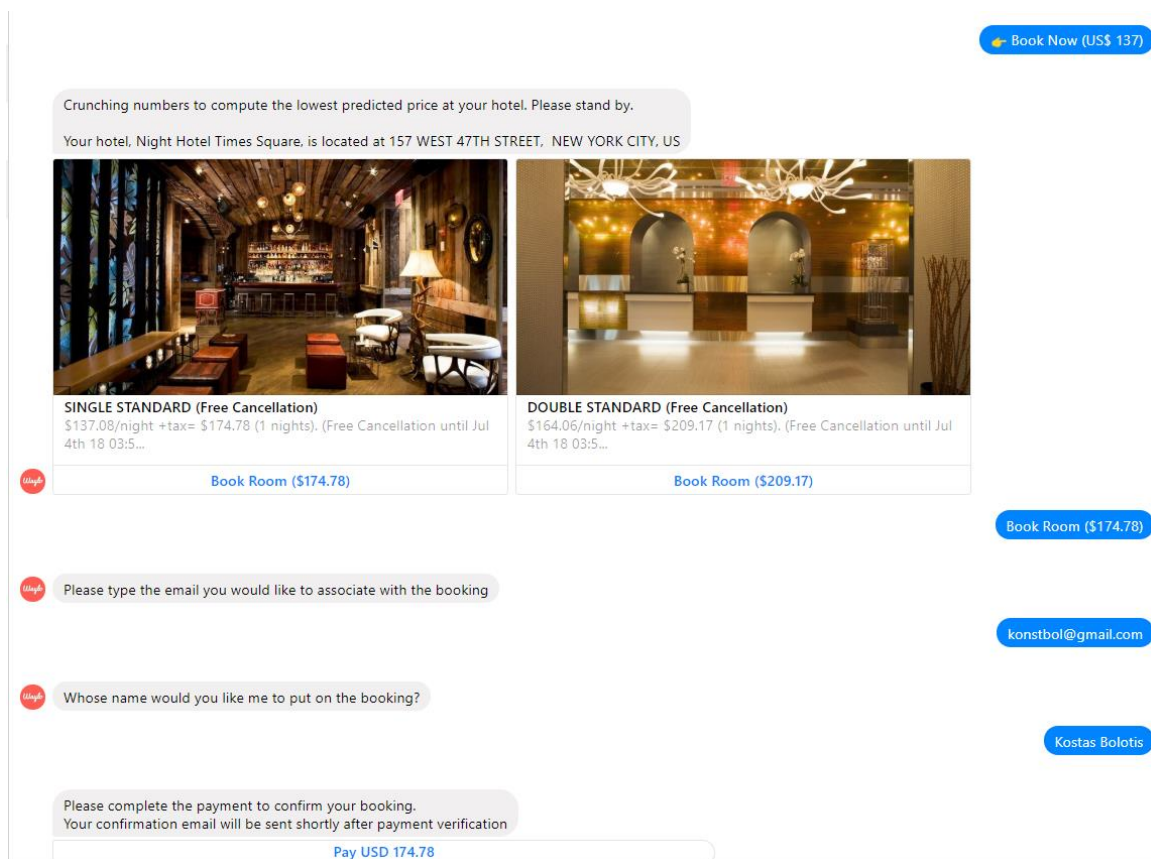
Εικόνα 39 – Waylo\_2 - Messenger

Μετά από ένα χρονικό διάστημα δευτερολέπτων μου εμφανίζει τα πιο φθηνά αποτελέσματα, παρέχοντας επίσης φίλτρα αναζήτησης και δίνοντας φυσικά στον χρήστη τη δυνατότητα να προχωρήσει στη συναλλαγή.

The screenshot displays three hotel listings in a grid. Each listing includes a photo, a title with price and rating, a description with price per night, distance, and dates, and a 'Book Now' button with price. The first listing is 'Night Hotel Times Square (7.4/10)' for \$137/night. The second is 'The Edison Hotel NYC (7.2/10)' for \$162/night. The third is 'Ni Square (7.8/10)' for \$147/night. Below the listings are navigation buttons: 'More Hotels', 'Filter', '4+ star', 'Specific Hotel', and 'Change Search'.

Εικόνα 40 – Waylo\_3 - Messenger

Όσο αφορά τη συναλλαγή, πραγματοποιείται εν μέρει μέσω των μηνυμάτων με πολύ εύκολο τρόπο, παρέχοντας το email σου, δίνοντας το όνομα σου και στο τέλος ανακατευθύνεσαι σε ιστότοπο για να ολοκληρώσεις τη συναλλαγή. Προσωπικά, είναι μία πολύ εύχρηστη ροή, που παρακινεί τον χρήστη να πραγματοποιήσει την πληρωμή μέσω του εν λόγω bot. Παρακάτω φαίνονται πλήρως τα βήματα που περιγράψαμε.



Εικόνα 41 – Waylo\_4 - Messenger

### 3.2 Πίνακας σύγκρισης των χαρακτηριστικών

|              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | BotVoyage |
|--------------|---|---|---|---|---|---|---|---|---|----|-----------|
| Multilingual |   |   |   |   |   |   |   |   |   |    | +         |
| Speech       |   |   |   |   |   |   |   |   |   |    | +         |
| Messenger    | + | + | + | + | + | + | + | + | + | +  | +         |
| Transactions |   | + |   | + |   |   |   | + | + | +  | +         |
| Location     | + |   |   | + |   | + | + | + | + | +  | +         |



|                |   |   |   |   |   |   |   |   |   |   |   |
|----------------|---|---|---|---|---|---|---|---|---|---|---|
| NLP            | + |   |   | + | + | + |   |   | + | + | + |
| Link Account   |   | + | + |   |   | + |   | + |   |   | + |
| Search Variety |   | + | + | + | + | + | + | + | + | + | + |
| Friendly UI    |   |   |   | + | + | + |   |   | + |   | + |
| Operative Flow | + | + | + | + | + | + |   | + | + | + | + |

**Πίνακας 1 - Σύγκριση διάφορων bot**

Υπόμνημα Πίνακα – Επεξήγηση

- Multilingual: Να υποστηρίζει πλήθος γλωσσών
- Speech: Αναγνώριση ομιλίας
- Messenger: Συμβατό με messenger
- Transactions: Να μπορεί να πραγματοποιεί συναλλαγές σε main sites π.χ. booking.com
- Location: Επιλογή αναγνώρισης τοποθεσίας
- NLP: Δυνατότητα κατανόησης με τεχνητή νοημοσύνη φυσικής γλώσσας
- Link Account: Διαμόρφωση προφίλ, ώστε χωρίς spam να προβάλλει προτάσεις
- Search Variety: Πλήρης και ποικίλη αναζήτηση δυνατοτήτων
- Friendly UI: Περιβάλλον χρήσης εύκολο στο μάτι
- Operative Flow: Κατανοητή λειτουργικότητα, με μία συνεχή ροή χωρίς να δημιουργούνται προβλήματα στον χρήστη.

### 3.3 Συμπεράσματα ανάλυσης

Μέσα από όλο το εύρος της ανάλυσης των παραπάνω bot, είναι πολύ εύκολο να παρατηρήσει κανείς τη γενικότερη λογική που εφαρμόζεται στον χώρο των ταξιδιών και του τουρισμού. Καταρχάς απαραίτητο βήμα είναι η αποστολή της τοποθεσίας από τον χρήστη εάν το επιθυμεί, η αλλιώς εάν παρέχεται από το bot, η πληκτρολόγηση της τοποθεσίας που επιθυμεί, χωρίς να διαμοιράζει το ακριβές σημείο που βρίσκεται. Στη συνέχεια, προσφέρεται στον χρήστη ένα πλήθος επιλογών για ενημέρωση σε σχέση με ξενοδοχεία, πτήσεις και διάφορες άλλες δραστηριότητες, αλλά ταυτόχρονα έχει τη δυνατότητα να πραγματοποιήσει συναλλαγές είτε αυτά τα bot είναι εκ φύσεως επιχειρήσεις, ταξιδιωτικά γραφεία, είτε είναι απλώς ενδιάμεσα λογισμικά μεσάζοντες τα οποία απλώς ανακατευθύνουν. Επιπρόσθετα για κάθε επιλογή παρατηρείται μία κατηγοριοποίηση, με αποτέλεσμα να προκύπτει ποικιλομορφία στις λειτουργίες που μπορεί να αξιοποιήσει ο χρήστης. Όσο αφορά τη γραφική απεικόνιση, είδαμε περιπτώσεις που ξεχώρισαν αισθητικά και άλλες που απλώς ήταν λειτουργικές. Τέλος όπως έχουμε ήδη αναφέρει, παρατηρήσαμε την ύπαρξη NLP στις περισσότερες των περιπτώσεων. Συνδυάζοντας όλα τα ειπωθέντα, έχουμε ως στόχο τη διαμόρφωση ενός αυτοματοποιημένου λογισμικού που να περιλαμβάνει τα καλύτερα στοιχεία από τα κυριότερα bot.

## ΚΕΦΑΛΑΙΟ 4-ΜΕΘΟΔΟΛΟΓΙΑ

### 4.1 Σύλληψη και διαμόρφωση ιδέας

Σε πρώτο επίπεδο, ας ξεκινήσουμε με την αρχική έμπνευση σε σχέση με την ενασχόληση με τον τομέα των bot. Το πρώτο ερέθισμα ήταν το 2016, όταν παρευρεθήκαμε στο σεμινάριο της κοινότητας του Student Guru που εκπροσωπεί το πανεπιστήμιο μας, το οποίο είχε θέμα μία γενική προσέγγιση στον χώρο των bot. Έπειτα, προβληματιστήκαμε εάν υπάρχει κάποια ιδέα η οποία θα μπορούσε να προσφέρει κάποια λειτουργικότητα στους φοιτητές, αλλά και συνάμα να μάθουμε για την εν λόγω τεχνολογία μιας και μας κίνησε το ενδιαφέρον. Ως αποτέλεσμα προχωρήσαμε στην υλοποίηση ενός bot, το οποίο ενημερώνει τους φοιτητές για το πρόγραμμα της εστίας σε καθημερινό πλαίσιο. Φτάνοντας στην τρέχουσα χρονιά, αναζητώντας ένα θέμα για τη διπλωματική, θεωρήσαμε θεμιτή την επιλογή να υλοποιήσουμε αντίστοιχα ένα bot που θα αποσκοπεί στην παροχή μίας καινοτόμου ιδέας, υπηρεσίας. Μη έχοντας ακριβώς στο μυαλό μας με ποιο χώρο ακριβώς να ασχοληθούμε, αναζητήσαμε στο google περί χρηματοοικονομικών bot, τα οποία ασχολούνται με παρακολούθηση μετοχών και κυρίως με συμβουλές περί επενδύσεων. Μετά την ανάλυση αυτών, επακολούθησε μία επιφανειακή αναζήτηση τους στο messenger για να διαλευκανθεί η λειτουργικότητά τους. Εν τέλει προβληματιστήκαμε αρκετά, διότι η ροή χρήσης παρουσίαζε μία σχετική δυσκολία για έναν χρήστη ο οποίος δεν έχει γνώση περί του θέματος και εμπειριούνταν διάφορες έννοιες οι οποίες απέκλιναν από τον τομέα των ενδιαφερόντων μας. Επόμενο βήμα ήταν η ανεύρεση ενός διαφορετικού χώρου προς υλοποίηση. Χρειάστηκε μία αναζήτηση μέσω της Google, με τη λέξη κλειδί ‘top bots’, ώστε να ανακατευθυνθούμε σε ένα σχετικό άρθρο που ανέλυε τα κυριότερα bot σε γενικό πλαίσιο. Έτσι, παρατηρήσαμε ότι τα περισσότερα αφορούσαν ταξίδια με μία κατηγοριοποίηση περί αυτών, κυρίως όσο αφορά ξενοδοχεία και πτήσεις. Σκεπτόμενοι και το γεγονός ότι στη χώρα μας είναι ένας εδραιωμένος σε αρκετά μεγάλο βαθμό τομέας, και από τη στιγμή επίσης που στοχεύουμε και σε ελληνικό κοινό, εκτός φυσικά από το εξωτερικό, καταλήξαμε να εντρυφήσουμε στον τουρισμό και να αναπτύξουμε ένα αυτοματοποιημένο λογισμικό που να εμπεριέχει τεχνητή νοημοσύνη, το οποίο θα προσφέρει μία γενικότερη διευκόλυνση. Σε πρώτο επίπεδο, αναλύσαμε όπως και παρατηρείται φυσικά στο προηγούμενο κεφάλαιο τα κυριότερα bot, και τα δοκιμάσαμε για να κρατήσουμε τα θετικά στοιχεία τους και να τα συνδυάσουμε προς όφελος μας. Έτσι, επειδή προσέφεραν ποικιλία λειτουργιών έπρεπε να κατασταλάξουμε σε ένα πεπερασμένο σύνολο. Σε δεύτερο επίπεδο, ήταν απαραίτητο να δούμε ποια εργαλεία και πλατφόρμες μας βολεύουν καλύτερα για την υλοποίηση. Όσον αφορά την πλατφόρμα που θα θεμελιωθεί το bot, είχαμε πλήθος διαφορετικών επιλογών. Ξεκινήσαμε έχοντας κατά νου, ότι θα μπορούσε να υλοποιηθεί και να χρησιμοποιείται μέσω Skype, ωστόσο δεν αποτελεί την καθημερινή λύση στην επικοινωνία και ενημέρωση από τον μέσο χρήστη, οπότε και φτάσαμε να ασχοληθούμε με το Facebook και συγκεκριμένα το messenger, μιας και συνιστά παγκόσμιας αναγνώρισης. Επιπρόσθετα δεν υπήρξε καμία αμφιβολία για το ποιο framework θα χρησιμοποιούσαμε, το οποίο μας δίνει τη δυνατότητα να επιλέξουμε εμείς το παραπάνω, δηλαδή εάν η εφαρμογή μας στεγάζεται στο Skype, Facebook ή σε κάποιο άλλο κανάλι, μέσο. Η επιλογή αυτή συναντάται με την ορολογία ‘Microsoft Bot Framework’. Στη συνέχεια αναζητήσαμε ποια προγραμματιστική γλώσσα συνδυάζεται με τον βέλτιστο τρόπο με τη συγκεκριμένη τεχνολογία. Στο προηγούμενο bot που αναφέραμε, είχε χρησιμοποιηθεί C#. Ωστόσο παρατηρήσαμε και την ύπαρξη της Javascript, η οποία προσφέρεται μέσα από τον χώρο του Node.js. Θέλοντας να μάθουμε



παραπάνω για τη συγκεκριμένη τεχνολογία, αναζητήσαμε το documentation της Microsoft, για να κατανοήσουμε τη χρήση του. Τελικά, επιλέξαμε τη δεύτερη επιλογή, καθώς αποτελούσε μία

πρόκληση για κάτι το καινούριο, αλλά και επειδή ολοένα και εδραιώνεται στις εφαρμογές/υπηρεσίες διαδικτύου. Επόμενο απαραίτητο στοιχείο της εφαρμογής μας που έπρεπε να αναζητήσουμε ήταν η κατάλληλη επιλογή ενός εργαλείου που θα μας βοηθήσει στη διαμόρφωση της βέλτιστης δυνατής έκδοσης NLP. Αρχικά, εύλογα σκεφτήκαμε να ασχοληθούμε με την υπηρεσία μηχανικής μάθησης Luis της Microsoft, η οποία όμως δεν υποστήριζε την ελληνική γλώσσα. Έπειτα, κατευθυνθήκαμε στο DialogFlow της Google, το οποίο δε μας βοήθησε επίσης και στη συνέχεια ασχοληθήκαμε με το rasa nlu, το οποίο αν και ενδελεχές και αξιόλογο, δεν υποστήριζε ελληνικά, αν και σου προσέδιδε τη δυνατότητα να προσθέσεις εσύ τη δικιά σου γλώσσα. Τελικά, βρήκαμε την πλατφόρμα wit.ai, το οποίο υποστήριζε την ελληνική γλώσσα, αλλά όπως είναι γνωστό με κάποια προβλήματα, λόγω της φύσης της.

## 4.2 Η δυσκολία της ελληνικής γλώσσας – NLP

Σε πρώτο επίπεδο ας ξεκινήσουμε με μία θεωρητική προσέγγιση όσον αφορά την πηγαία υπόσταση της ελληνικής γλώσσας και σε δεύτερο επίπεδο θα αναλύσουμε ένα συγκεκριμένο παράδειγμα ώστε να γίνει με απλό τρόπο κατανοητή η γενικότερη δυσκολία που παρουσιάζει. Αρχικά, η ελληνική γλώσσα, όπως και θα έγινε κατανοητό από τα ήδη προαναφερθέντα, παρουσιάζει κάποια ιδιαίτερα στοιχεία, τα οποία σε διεθνές πλαίσιο δυσχεραίνουν την ομιλία και τη γραφή της από φυσικούς χρήστες, πόσο μάλλον την εκπαίδευση της από κάποια πλατφόρμα μηχανικής μάθησης. Ένα πρώτο στοιχείο συνιστά η πολύπλοκη δομή της, και η ευρύτερη ποικιλομορφία της όσο αναφορά τη σύνθεση και τη σύνταξη της, η οποία και είναι δυνατή λόγω του πολυπληθούς λεξιλογίου της. Έτσι συνήθως συναντώνται δυσκολίες στην κατανόηση σε επίπεδο NLP, καθώς ένα συγκεκριμένο θέμα μπορεί να ειπωθεί με πολλούς διαφορετικούς τρόπους. Ένα ακόμα χαρακτηριστικό το οποίο δε συναντάται στις περισσότερες γλώσσες είναι η ύπαρξη τόνων σε μία λέξη, και όπως είναι γνωστό μπορεί να είναι και παραπάνω από ένας όταν βρίσκεται συνήθως στη γενική πτώση. Ως αποτέλεσμα, μία λέξη μπορεί να γραφτεί είτε χωρίς είτε με τόνο, οπότε και η διαδικασία της κατανόησης της φυσικής γλώσσας δυσχεραίνεται σε έναν σημαντικό βαθμό. Επιπρόσθετα, όπως και στις περισσότερες γλώσσες, η ελληνική διαθέτει εκτός των πεζών γραμμάτων και κεφαλαία, τα οποία προφανώς διαφοροποιούνται μεταξύ τους στις περισσότερες των περιπτώσεων και μία λέξη αλλάζει αισθητά στις δύο περιπτώσεις. Επίσης πολλές φορές συνδυάζεται η ύπαρξη κεφαλαίων και πεζών σε μία λέξη είτε λόγω λάθους από έναν χρήστη, είτε επειδή ξεκινά μία πρόταση, είτε αναφερόμαστε σε ένα όνομα και ούτω καθεξής, στοιχεία τα οποία δικαιολογούν τη δυσκολία που παρουσιάζεται στην ανεύρεση ενός αξιόπιστου εργαλείου NLP. Ένα επίσης χαρακτηριστικό το οποίο έχει επέλθει την τελευταία δεκαετία, είναι τα λεγόμενα ‘greeklish’, τα οποία ουσιαστικά απεικονίζουν την ελληνική γλώσσα, χρησιμοποιώντας τα λατινικά γράμματα. Το γεγονός αυτό έχει επηρεάσει σε έναν σημαντικό βαθμό, μέσω των νεότερων γενιών το υπόβαθρο της ελληνικής γλώσσας, καθώς σταδιακά έχει αλλοιωθεί η ορθογραφία πολλών ελληνικών λέξεων, αντί των οποίων έχουν προκύψει καινούριες λέξεις και εκφράσεις που δεν παραπέμπουν και πολύ στη μητρική μας. Επομένως αφού η γλώσσα μας πλέον μπορεί να απεικονιστεί και με αυτόν τον τρόπο, συνιστά όντως μία ακόμη αισθητή δυσκολία για το λογισμικό που θα αναλάβει την εκπαίδευση της. Έχοντας τελειώσει τη θεωρητική προσέγγιση, ας προχωρήσουμε στο δεύτερο και συντομότερο σκέλος, σε αυτό δηλαδή του παραδείγματος. Παραδείγματος χάριν, η λέξη ‘βοήθεια’ μπορεί να αναγραφεί είτε με τον σωστό τρόπο φυσικά, είτε χωρίς τόνο(‘βοηθεια’), είτε με συνδυασμό κεφαλαίων και πεζών (‘Βοήθεια’), είτε λανθασμένα

λόγω ορθογραφικού λάθους, βιασύνης ('βοιθεια') ή φυσικά με τον ξενόφερτο τρόπο ('voithia'). Φυσικά όποιος και να το έβλεπε το συγκεκριμένο παράδειγμα θα καταλάβαινε το σωστό νόημα, αλλά τι γίνεται στην περίπτωση ενός λογισμικού. Πως γίνεται να επιτευχθεί ένα αξιόλογο απόσταγμα τεχνητής νοημοσύνης γύρω από το συγκεκριμένο δύσκολο ζήτημα, την κατανόηση της ελληνικής γλώσσας. Έχοντας επιμονή και υπομονή καταλήξαμε στην ανεύρεση ενός αρκετά αποτελεσματικού εργαλείου, το οποίο όπως και προαναφέραμε παρουσιάζει δυσκολίες και περιορισμούς, αλλά σε τέτοιο βαθμό που κάνει δυνατή την ύπαρξη του NLP και επαγωγικά προφανώς, την ανάπτυξη αυτοματοποιημένων λογισμικών που διευκολύνουν τη ζωή του χρήστη. Η ανάλυση αυτού θα πραγματοποιηθεί σε επερχόμενη επικεφαλίδα στο τρέχον κεφάλαιο.

### 4.3 Πλατφόρμες και εργαλεία

Παρακάτω θα προχωρήσουμε στην ανάλυση των γλωσσών προγραμματισμού, των εργαλείων και πλατφορμών με μία σχετική αναδρομή, επεξήγηση και γενικότερη χρήση τους. Σε πρώτο επίπεδο θα αναλυθούν τα βασικά τμήματα που θεμελιώνουν την εφαρμογή μας, οι γλώσσες δηλαδή που χρησιμοποιήθηκαν, στη συνέχεια σε ποιες πλατφόρμες, frameworks αναπτύχθηκαν και τέλος με ποια εργαλεία παραμετροποιήθηκαν.

#### 4.3.1 JavaScript

##### 4.3.1.1 Ιστορική αναδρομή

Το 1995, η Netscape ανακοίνωσε την JavaScript, παρουσιάζοντάς την ως μία πολύ εύκολη λύση όσον αφορά τη σχεδίαση live online εφαρμογών, οι οποίες διασυνδέουν αντικείμενα και πόρους και στους clients και στους servers. Ως αποτέλεσμα, αποτέλεσε την πρώτη επιλογή των προγραμματιστών για την ανάπτυξη client-side εφαρμογών μες στο πλαίσιο των προγραμμάτων περιήγησης στο διαδίκτυο [13].

Ωστόσο, το 1995 επίσης με την κυκλοφορία του Netscape Enterprise Server, η JavaScript άρχισε να εφαρμόζεται και στη διαμόρφωση της πλευράς των διακομιστών( server-side). Στη συνέχεια, το 1996 συμπεριλήφθηκε στον IIS(διακομιστής διαδικτύου που υποστηρίζει τα HTTP και FTP πρωτόκολλα [15] ) και παρουσιάστηκε με το όνομα Jscript σε ASP και .NET σελίδες [14].

Τέλος στα μέσα της δεκαετίας του 2000, παρουσιάστηκαν διάφορες καινούριες server-side εφαρμογές της JavaScript, με διασημότερη πλέον το Node.js, το οποίο και θα αναλυθεί σε επόμενη επικεφαλίδα [14].

##### 4.3.1.2 Τι είναι

Η JavaScript η οποία συνήθως συναντάται με τη συντομογραφία JS είναι μία αντικειμενοστραφής, ελαφριά γλώσσα προγραμματισμού, η οποία υποστηρίζει την ύπαρξη συναρτήσεων πρώτης κλάσης( δυνατότητα περάσματος συναρτήσεων ως όρισμα σε άλλη συνάρτηση [8]) και είναι κυρίως γνωστή ως scripting γλώσσα για ιστοσελίδες. Ωστόσο χρησιμοποιείται σε πολλά περιβάλλοντα, τα οποία δε βασίζονται στην ύπαρξη προγράμματος περιήγησης(browser).

Πρόκειται για μια βασισμένη σε πρότυπα(αξιοποιεί την έννοια της κληρονομικότητας [9]) , πολλαπλών παραδειγμάτων scripting γλώσσα προγραμματισμού που είναι δυναμική, και υποστηρίζει αντικειμενοστραφή, επιτακτικά( χρήση εντολών για αξιοποίηση από τον υπολογιστή [10]), και λειτουργικά στυλ προγραμματισμού.( functional programming, δομημένο δηλαδή κώδικα με σειριακή εκτέλεση [11]) [12].

### 4.3.1.3 Που εφαρμόζεται

Η συγκεκριμένη γλώσσα εφαρμόζεται σε πλήθος διαφορετικών εφαρμογών, οι οποίες και αναλύονται παρακάτω με μία σχετική κατηγοριοποίηση [17].

#### 4.3.1.3.1 Παιχνίδια

Συναντάται σε μικρότερης κλίμακας παιχνίδια όπως το διάσημο ‘2048’ [18]. Επίσης, χρησιμοποιείται για την ανάπτυξη παιχνιδιών σε game engines, όπως για παράδειγμα μέσα στην πλατφόρμα της Unity.

#### 4.3.1.3.2 Εφαρμογές παροχής πληροφοριών

Το εύρος εφαρμογών περιλαμβάνει αναγνώστη RSS feed, εφαρμογές καιρού και χαρτών [17], ακόμη και εφαρμογές εκμάθησης ξένων γλωσσών όπως είναι το διάσημο Duolingo [19].

#### 4.3.1.3.3 Εφαρμογές παραγωγικότητας

Η συγκεκριμένη κατηγορία περιλαμβάνει κάποιες υποστηρικτικές εφαρμογές που βοηθούν είτε για παράδειγμα στη διαμόρφωση και οργάνωση ενός προσωπικού καθημερινού προγράμματος [17], είτε ακόμη και στην επεξεργασία φωτογραφίας, όπως το διάσημο Polar [19].

#### 4.3.1.3.4 Βοηθητικά εργαλεία προγραμματισμού

Στη συγκεκριμένη κατηγορία συναντώνται editors, όπως ο διάσημος Atom που είναι εξ’ολοκλήρου γραμμένος σε JavaScript, μέσω του οποίου μπορεί κανείς και να γράφει κώδικα αλλά και να επιδιορθώσει τυχόντα σφάλματα (debugging) [20].

## 4.3.2 Node.js

### 4.3.2.1 Ιστορική αναδρομή

Δημιουργήθηκε το 2009 από τον Ryan Dahl, 13 χρόνια μετά την ανάπτυξη του πρώτου περιβάλλοντος διακομιστή γραμμένο σε JavaScript, τον Enterprise τον οποίο και αναλύσαμε παραπάνω. Η πρώτη έκδοση περιλάμβανε συμβατότητα μόνο με τα λειτουργικά συστήματα των Mac και Linux. Η ευρύτερη ανάπτυξη και συντήρηση πραγματοποιήθηκε από τον προαναφερθέντα και στη συνέχεια χορηγήθηκε από την Joyent [21].

Τον Ιανουάριο του 2010, εισάχθηκε ένας διαχειριστής πακέτων με την ονομασία npm (node package manager) [21], ο οποίος προσδίδει στην κοινότητα των προγραμματιστών διευκόλυνση, καθώς προσφέρει έτοιμα τμήματα κώδικα προς επαναχρησιμοποίηση και επεκτασιμότητα. Η ανάλυση αυτού θα γίνει αργότερα στο ίδιο κεφάλαιο.

Τον Ιούνιο του 2011, η Microsoft σε συνδυασμό με την Joyent προχώρησε στην ενσωμάτωση του Node.js στα Windows, με αποτέλεσμα τον επόμενο μήνα να κυκλοφορήσει η πρώτη υποστηριζόμενη έκδοση [21].

Τον Δεκέμβρη του 2014, ο πηγαίος κώδικας του Node.js χρησιμοποιήθηκε ως βάση για να δημιουργηθεί το io.js (η εν λόγω διαδικασία περιγράφεται με τον όρο fork) [21].

Τον Ιούνιο του 2015 η κοινότητα του io.js αναμείχθηκε με εκείνη του node, και από εκεί και έπειτα, το io.js δεν ανέπτυξε νέες εκδόσεις, αλλά μαζί συνέχισαν να αναπτύσσουν συλλογικά το Node.js [21]. Στην παρούσα χρονική στιγμή, βρισκόμαστε στην έκδοση 10.

#### 4.3.2.2 Τι είναι

Το Node.js είναι με απλά λόγια μία πλατφόρμα η οποία χρησιμοποιείται για ανάπτυξη server-side JavaScript [22]. Ειδικότερα είναι μία μηχανή JavaScript, η οποία είναι θεμελιωμένη επάνω στην Chrome's V8 JavaScript Engine [23]. Για την καλύτερη κατανόηση ας επεξηγήσουμε τι είναι η συγκεκριμένη μηχανή.

Η εν λόγω μηχανή, όπως κάθε JavaScript engine έχει ως στόχο την εκτέλεση του κώδικα που είναι γραμμένος σε JavaScript και η μετατροπή του στην πιο βελτιστοποιημένη μορφή ώστε είτε να μπορεί να ερμηνευτεί από το πρόγραμμα περιήγησης, είτε ακόμα να ενσωματωθεί σε μία εφαρμογή. Το παράδοξο με την V8 είναι το γεγονός ότι ο πηγαίος κώδικας της έχει γραφεί σε C++. Η χρήση της έγκειται στον Google Chrome και φυσικά στο Node.js, αλλά και σε άλλα περιβάλλοντα [24].

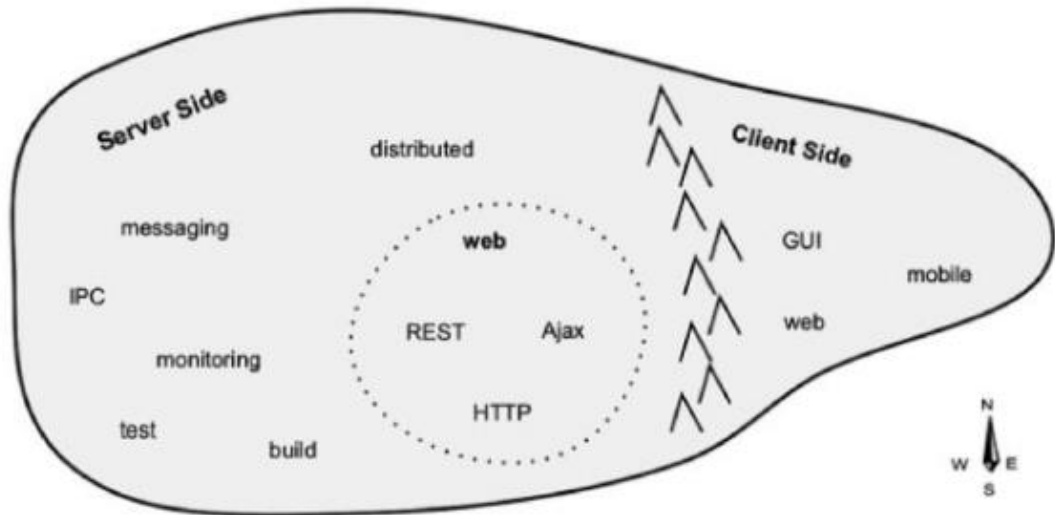
Η πλατφόρμα του Node.js χρησιμοποιεί ένα non-blocking I/O μοντέλο το οποίο επιτρέπει στον κώδικα να έχει ευελιξία και ταχύτητα [23], διότι δεν υπάρχει απλώς ένα thread το οποίο να τρέχει σειριακά και να ελέγχει όλες τις λειτουργίες. Στο υπόβαθρο του Node.js συναντάται μία πολυνηματική αρχιτεκτονική, μέσω της οποίας δεν υπάρχει η πιθανότητα το πρόγραμμα όπως γίνεται κατανοητό κι από την παραπάνω λέξη (non-blocking) να μπλοκαριστεί είτε λόγω ενός λάθους κατά την εκτέλεση είτε λόγω μίας γενικής καθυστέρησης κατά την εκτέλεση ενός request, διότι απλώς θα απασχολείται το συγκεκριμένο τμήμα κι όχι όλο το υπόλοιπο κομμάτι του λογισμικού, μιας και σε κάθε thread έχει ανατεθεί μία λειτουργία, διεργασία. Το εν λόγω μοντέλο εύλογα προσδίδει στην πλατφόρμα τον χαρακτηρισμό ασύγχρονη.

Μέσω του Node.js διατίθεται το μεγαλύτερο οικοσύστημα πακέτων ανοιχτού λογισμικού [23], το οποίο όπως αναφέραμε παραπάνω, προσφέρει βιβλιοθήκες κώδικα προς παραμετροποίηση και εκ νέου επέκταση.

Συμπερασματικά το Node.js αποτελεί μία επέκταση της Chrome's V8 JavaScript Engine, συνιστώντας ένα περιβάλλον διακομιστή, προσδίδοντας στον developer τη δυνατότητα να αναπτύξει server-side JavaScript λογισμικό με ταχύτητα, ευελιξία και ένα υπέρογκο πλήθος πακέτων, μέσω του npm.

#### 4.3.2.3 Απεικόνιση της αρχιτεκτονικής

Στο συγκεκριμένο χωρίο θα προχωρήσουμε σε μία απεικόνιση, η οποία θα μας οδηγήσει να κατανοήσουμε σε ποιο επίπεδο βρίσκεται το Node.js, ανάμεσα στο ευρύτερο πλαίσιο των πραγμάτων μέσα στο πλαίσιο του διαδικτύου. Παρακάτω παρατίθεται μία σχετική εικόνα, την οποία και θα επεξηγήσουμε.

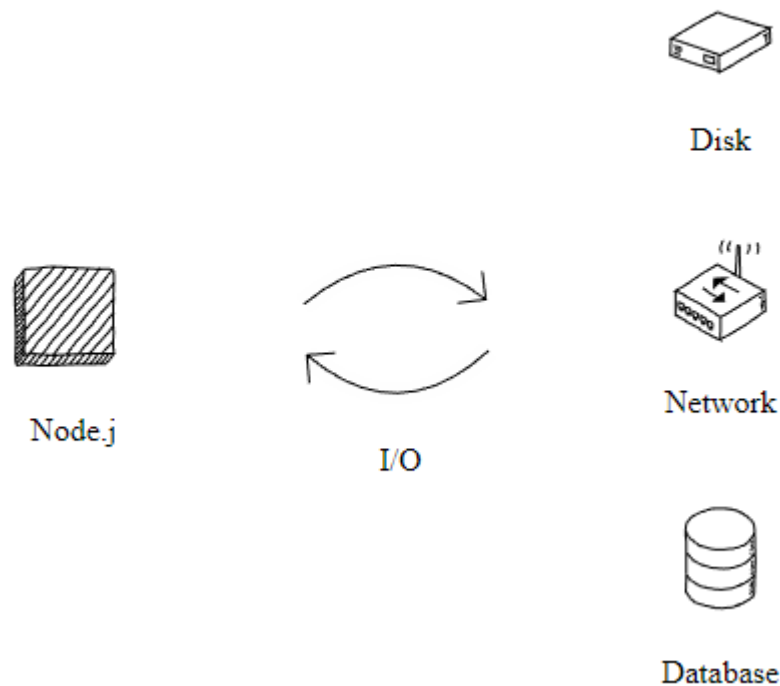


**Εικόνα 42 – Node.js Map - Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2017). Node. js in Action. Manning Publications.**

Αρχικά έχετε κατά νου με βάση τον πάνω χάρτη, ότι όσα στοιχεία βρίσκονται σε εγγύτητα, έχουν κοινό σκοπό και όσα απέχουν διαφέρουν σε αυτόν. Τα στοιχεία αυτά είναι προγράμματα τύπου I/O-bound (Input-εισόδου Output-εξόδου), στα οποία παραπάνω επεξεργαστική ισχύ ή μνήμη RAM δεν πρόκειται να δημιουργήσουν κάποια αισθητή διαφορά [23].

Ας εξηγήσουμε την έννοια I/O-bound για καλύτερη κατανόηση. Στην επιστήμη των υπολογιστών, η συγκεκριμένη έννοια έχει σχέση με μία κατάσταση που ο χρόνος που απαιτείται για να πραγματοποιηθεί ένας υπολογισμός, καθορίζεται κυρίως από την περίοδο που χρειάστηκε για να ολοκληρωθεί [25]. Με άλλα λόγια μία διεργασία, λειτουργία λέγεται I/O-bound ,όταν αξιοποιούνται πηγές εισόδου/ εξόδου κατά την εκτέλεση .Οι περισσότερες εφαρμογές αυτού του είδους καταναλώνουν τον περισσότερο χρόνο αναμένοντας λόγω διαδικτύου, αρχείων ή λόγω της βάσης δεδομένων. Όπως και στην περίπτωση του Node.js ο περισσότερος χρόνος αξιοποιείται σε παρόμοιες εργασίες, οπότε θα ήταν εύλογη για παράδειγμα η αύξηση της ταχύτητας του διαδικτύου για γρηγορότερα αποτελέσματα. Έχοντας περιγράψει την έννοια non-blocking προηγουμένως, μπορεί ίσως πλέον να γίνει πιο κατανοητό ότι όλες οι εισοδοι και εξοδοι (I/O) δε μπλοκάρονται μεταξύ τους κι άρα επιτρέπονται σε διάφορα request να εξυπηρετηθούν, μέχρι να γίνει μια διαδικασία εγγραφής είτε διαβάσματος [26].

## I/O bound



**Εικόνα 43 – I/O bound - Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2017). Node. js in Action. Manning Publications.**

Από την άλλη πλευρά CPU-bound, είναι όταν χρειάζεται τρομερή επεξεργαστική ισχύ, ώστε να γίνει μία πολυσύνθετη διαδικασία, όπως για παράδειγμα το hash ενός κωδικού [26].

Συνεχίζοντας την ανάλυση της εικόνας 42 στα ανατολικά (East), συναντώνται όλα τα προγράμματα που ανήκουν στην πλευρά του client, όπως για παράδειγμα εφαρμογές κινητών και διαδικτύου. Τα συγκεκριμένα αλληλοεπιδρούν με χρήστες, αναμένοντας μία είσοδο εκ αυτών. Στη δυτική πλευρά υπάρχουν όλα τα προγράμματα που έχουν σχέση με την πλευρά του server, όπου και είναι ο χώρος στον οποίο στεγάζεται το Node.js. Τέλος, μέσα στην ίδια ζώνη σε βαθύτερο επίπεδο μπορεί κανείς να βρει εδραιωμένες τεχνολογίες όπως το πρωτόκολλο HTTP, η τεχνολογία AJAX [23].

### 4.3.2.4 Που εφαρμόζεται

Παρακάτω, παρατίθενται δύο χαρακτηριστικά παραδείγματα τα οποία είναι γραμμένα στο server-side περιβάλλον του Node.js.

#### 4.3.2.4.1 Netflix

Αποτελεί τη μεγαλύτερη πλατφόρμα σε διαδικτυακό πλαίσιο όσο αναφορά το video streaming και έτσι για να μείνει σε αυτή τη θέση αξιοποίησαν την τεχνολογία του Node.js, μέσω της οποίας έγινε

δυνατή η παραμετροποίηση από τον χρήστη της σελίδας του και ο χρόνος φόρτισης της σελίδας σημείωσε κάθοδο μέχρι και 70% [27].

#### 4.3.2.4.2 PayPal

Δεν υπάρχει κανείς που να μη γνωρίζει τη συγκεκριμένη πλατφόρμα πληρωμών που εδρεύει στο διαδίκτυο, αλλά ελάχιστοι γνωρίζουν ότι χρησιμοποιεί Node.js. Εξυπηρετώντας βέλτιστα 200 εκατομμύρια χρήστες, υπήρξαν αρχικά θέματα σε σχέση με τη διαφοροποίηση του κώδικα ανάμεσα στα τμήματα του server και του client. Ωστόσο πλέον αξιοποιώντας μόνο JavaScript μέσω Node.js, έχει παρατηρηθεί βελτίωση παραγωγικότητας και μειώθηκε η ποσότητα του κώδικα κατά το εξαιρετικό ποσοστό του 33% [27].

### 4.3.3 Wit.ai

#### 4.3.3.1 Τι είναι NLP/NLU

Το NLP(Natural Language Processing) συνιστά έναν τομέα της επιστήμης των υπολογιστών, ο οποίος ασχολείται με την αλληλεπίδραση των υπολογιστών με τις φυσικές( ανθρώπινες) γλώσσες και ειδικότερα με το πως θα προγραμματίσει τους υπολογιστές να επεξεργάζονται αποτελεσματικά μεγάλα ποσά από δεδομένα φυσικής γλώσσας [38]. Με άλλα λόγια με τη βοήθεια του NLP, άνθρωποι και μηχανές μπορούν πλέον να μιλάνε φυσικά. Επιπρόσθετα είναι σημαντικό να κατανοηθεί, ότι είναι αναπόσπαστο τμήμα σε οποιοδήποτε εγχείρημα ανάπτυξης μίας τεχνητής νοημοσύνης που να αποσκοπεί στην προσομοίωση του ανθρώπινου είδους. [39]

Ένα NLP σύστημα είναι αποτελεσματικό, όταν μπορεί να συλλέξει οτιδήποτε έχει ειπωθεί προς αυτό, να το τμηματοποιήσει, να κατανοήσει τη σημασία και να ανταποκριθεί με τη σωστή επιλογή πράξης και φυσικά αυτή η πράξη να συνιστά φυσική και αντιληπτή από τον χρήστη [39].

Τις περισσότερες φορές προκύπτει παρανόηση μεταξύ της παραπάνω έννοιας και του NLU(Natural Language Understanding). Το τελευταίο αποτελεί απλώς ένα υποσύνολο του ευρύτερου κόσμου του NLP [39]. Για να γίνει κατανοητή η διαφοροποίηση ας προχωρήσουμε στην επεξήγηση της συγκεκριμένης έννοιας .Η έννοια του NLU πραγματεύεται την απόσπαση μίας δομημένης πληροφορίας από την αδόμητη μορφή της φυσικής γλώσσας, μέσω δηλαδή των μηνυμάτων σε μία πλατφόρμα chatting [40].

Στη συνέχεια, ας αναλύσουμε δύο σημαντικές ορολογίες για να γίνει κατανοητή η λειτουργία. Σε πρώτο επίπεδο συναντάται η έννοια intent( πρόθεση, σκοπός), η οποία ουσιαστικά είναι το θέμα για το οποίο μιλά ο χρήστης σε φυσική γλώσσα( π. χ Θα βρέξει αύριο?, εννοώντας τον καιρό). Σε δεύτερο επίπεδο υπάρχει η έννοια της entity(οντότητα) η οποία συνιστά ένα τμήμα πληροφορίας προς ανίχνευση από την είσοδο του χρήστη. Στο παραπάνω παράδειγμα θα πρέπει η οντότητα να είναι ο καιρός με κάποιες τιμές όπως ηλιόλουστος η βροχερός [42].

Υπάρχουν διάφορα είδη intent, τα οποία ας αναφέρουμε. Αρχικά, υπάρχουν trait entities(χαρακτηριστικών) , μέσω του οποίου ανιχνεύεται μια πληροφορία με βάση το γενικότερο νόημα της πρότασης. Δεύτερον με βάση κάποιες λέξεις κλειδιά (keywords) . Επίσης υπάρχει

συνδυασμός χαρακτηριστικών με λέξεις κλειδιά και τέλος υπάρχουν έτοιμες ανάλογα την υπηρεσία. Για παράδειγμα στο Wit.ai υπάρχει έτοιμη οντότητα η οποία κατανοεί την τοποθεσία.

Συμπερασματικά έχουμε την ευρύτερη κατηγορία του NLP, που στοχεύει στη φυσική επικοινωνία μεταξύ μηχανής και ανθρώπου και το NLU, το οποίο όντας υποσύνολο του παραπάνω, αποσκοπεί στη μετατροπή της αδόμητης πληροφορίας (φυσική γλώσσα) σε δομημένη. Για αυτόν τον σκοπό υπάρχουν υπηρεσίες που με τη λογική των intents και entities, τον εξυπηρετούν σε θεμιτό βαθμό.

#### 4.3.3.3 Ιστορική αναδρομή Wit.ai

Αποτελεί ένα χαρακτηριστικό παράδειγμα επιτυχίας μίας startup εταιρείας. Δημιουργήθηκε από τους Alex Lebrun, Laurent Landowski και Willy Blandin το 2014 [36].

Το επόμενο έτος λάνσαρε την εφαρμογή στον ιστότοπο του Hacker News, όπου βρήκε την κατάλληλη απήχηση που χρειαζόταν και εν τέλει έγιναν δεκτοί από την Y Combinator, η οποία συνιστά εταιρεία χρηματοδότησης προς startup εταιρείες, λαμβάνοντας το ποσό των 3 εκατομμυρίων [36].

Τον Ιανουάριο του 2015, η εταιρεία Wit.ai αποκτήθηκε από το Facebook με στόχο να το βοηθήσει, προσφέροντας λύσεις στην παραγωγή ελεγχόμενων με βάση τη φωνή εργαλείων ανάπτυξης και στη μετατροπή φωνής σε κείμενο στην πλατφόρμα του Messenger [37].

#### 4.3.3.3 Τι είναι το Wit.ai

Το Wit.ai είναι μία υπηρεσία NLU [40], το οποίο απλούστατα κατανοεί τι λένε οι χρήστες στην εφαρμογή σου [41] χρησιμοποιώντας τις κατηγορίες οντοτήτων για κατανόηση του περιεχομένου της φυσικής γλώσσας όπως αναλύθηκε παραπάνω. Για να επέλθει καλύτερο αποτέλεσμα, απαραίτητη είναι η εκπαίδευση με ένα εύλογο πλήθος διαφορετικών προτάσεων για το training μίας συγκεκριμένης οντότητας. Τέλος αλλά λιγότερο σημαντικό, μία παράμετρος της υπηρεσίας που πρέπει να παρατηρείται, είναι ο δείκτης αυτοπεποίθησης, ο οποίος κυμαίνεται από το 0 έως το 1 και προειδοποιεί τον διαχειριστή πόσο αποτελεσματικά έχει γίνει η εκπαίδευση.

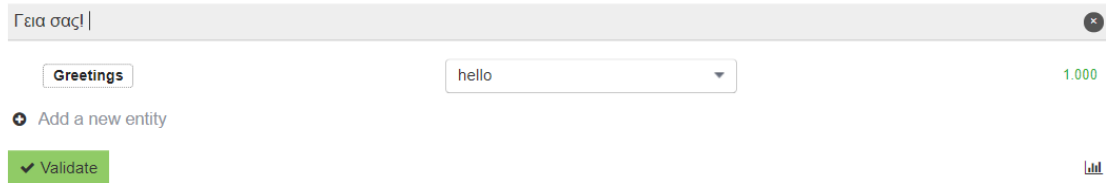
#### 4.3.3.4 Παράδειγμα

Παρακάτω ακολουθεί ένα απλό παράδειγμα με επεξήγηση για μία βασική κατανόηση του εργαλείου. Το σενάριο είναι να στέλνει ο χρήστης προτάσεις σε σχέση με χαιρετισμούς και να καταλαβαίνει ότι πρόκειται για αυτό το συγκεκριμένο θέμα. Δηλαδή όταν θα λέει είτε καλημέρα είτε γεια για παράδειγμα, θα επιστρέφει μία οντότητα Greetings με μία συγκεκριμένη τιμή.



## Test how your app understands a sentence

You can train your app by adding more examples



Γεια σας!

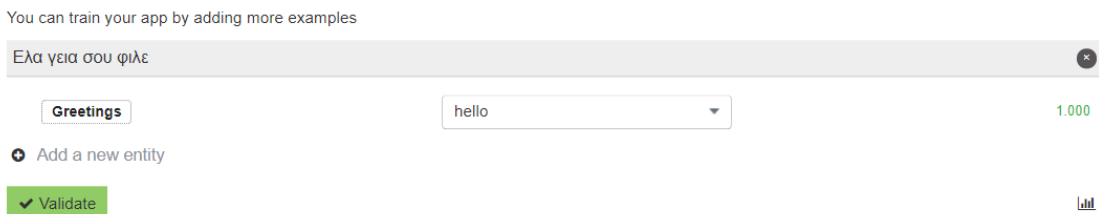
Greetings hello 1.000

⊕ Add a new entity

✓ Validate

### Εικόνα 44 – Wit.ai\_ex1 – Wit.ai( το δικό μας)

Παρατηρούμε ότι επιστρέφει την οντότητα όπως περιμέναμε με μία τιμή hello, καθώς προηγουμένως είχα γράψει την ίδια πρόταση και είχα κάνει validate, επικύρωση δηλαδή ότι πρόκειται για χαιρετισμό. Έτσι ο δείκτης αυτοπεποίθησης είναι 1, διότι είναι ακριβώς το ίδιο κείμενο. Παρακάτω δοκίμασα με μία άλλη πρόταση και ο δείκτης ήταν ξανά 1, χωρίς να έχω γράψει το ίδιο γιατί η οντότητα λειτουργεί με χαρακτηριστικά (trait entity) και συνειδητοποιεί το ευρύτερο νόημα χωρίς να γράψω το ίδιο.



You can train your app by adding more examples

Ελα γεια σου φιλε

Greetings hello 1.000

⊕ Add a new entity

✓ Validate

### Εικόνα 45 – Wit.ai\_ex2 – Wit.ai( το δικό μας)

## 4.3.4 Npm

### 4.3.4.1 Ιστορική αναδρομή

Αναπτύχθηκε από τον Isaac Z. Schlueter, διότι είχε παρατηρήσει ότι η συσκευασία των ενοτήτων, η πιο κατανοητά module packaging (το module αντιστοιχεί σε μία ενότητα κώδικα και packaging στην ενοποίηση αυτών) και λόγω της έμπνευσης από την ύπαρξη ήδη υπαρχόντων εφάμιλλων project, όπως το PEAR στη γλώσσα της PHP και το CPAN σε Perl. Κυκλοφόρησε η πρώτη έκδοση στις 12 Ιανουαρίου του 2010 [43].

Τον Μάρτιο του 2016, λόγω ενός πακέτου που λεγόταν left-pad, το οποίο ήταν εξαρτημένο από διάφορα σημαντικά JavaScript πακέτα και καταργήθηκε η δημοσίευσή του, τράβηξε τη προσοχή του τύπου και έτσι άλλαξε οριστικά η πολιτική του npm για τις καταργήσεις των δημοσιεύσεων [43].

Τον Φεβρουάριο του 2018, στην έκδοση 5.7.1, ανακαλύφθηκε ένα θέμα εκτελώντας την εντολή sudo npm στα συστήματα με λειτουργικό σύστημα Linux. Ειδικότερα, διαφοροποιήθηκε η κατοχή των αρχείων συστήματος, διασπώντας μονίμως το λειτουργικό σύστημα [43].

Στη τρέχουσα χρονική περίοδο, Μάιος του 2018 βρίσκεται στην έκδοση 6.0.1 [43].

#### 4.3.4.2 Τι είναι

Αποτελώντας τον διαχειριστή πακέτων για την JavaScript, διαμορφώνει ένα τεράστιο σύνολο από πακέτα τα οποία προσεγγίζουν τα εξακόσιες χιλιάδες με περίπου τρία δισεκατομμύρια downloads ανά εβδομάδα. Χρησιμοποιείται από προγραμματιστές που αναπτύσσουν ανοιχτό λογισμικό για να δανειστούν κάποιο αλλά και να κοινοποιήσουν ένα δικό τους πακέτο, με αποτέλεσμα όλη η κοινότητα και το σύνολο των πακέτων να αυξάνεται.

Κατανέμεται σε τρία τμήματα και συγκεκριμένα στην ιστοσελίδα, όπου κανείς μπορεί να εξερευνήσει πακέτα, και να διαμορφώσει κάποιο λογαριασμό, στο CLI(Command Line Interface) , ο τρόπος δηλαδή για να αλληλοεπιδράσει ο χρήστης με το npm, και τέλος στο registry, το οποίο ουσιαστικά συνιστά μία δημόσιας εμβέλειας βάση δεδομένων λογισμικού JavaScript μαζί με όλα τα απαραίτητα μεταδεδομένα [44].

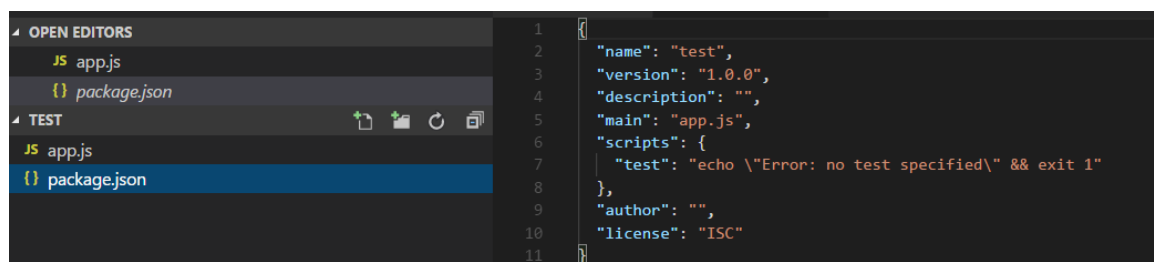
#### 4.3.4.3 Πώς χρησιμοποιείται

Αρχικά, στον IDE που διαθέτει κάποιος χρήστης(ας πάρουμε για παράδειγμα του Visual Studio Code που χρησιμοποιήσαμε εμείς) διαμορφώνεις ένα αρχείο package.json, εκτελώντας την εντολή npm init σε έναν συγκεκριμένο φάκελο , μέσω του οποίου αρχείου θα γίνονται οι αναφορές για ποια πακέτα έχουν χρησιμοποιηθεί και περιλαμβάνει επίσης απαραίτητες πληροφορίες για τον εκδότη, την άδεια, την έκδοση κι άλλα στοιχεία της εφαρμογής στην οποία έγινε το init. Για να εγκαταστήσεις κάποιο πακέτο χρησιμοποιείς στο terminal την παρακάτω εντολή και με το –save παραμένει καταχωρημένο στο package.json [45].

```
PS C:\Users\User\Desktop\test> npm init
```

Εικόνα 46 – npm\_init – Node.js( το δικό μας)

Μετά το npm init παρατηρούμε την ύπαρξη του package.json. Στη συνέχεια ας εγκαταστήσουμε ένα πακέτο για να το δούμε.



```
1 {
2   "name": "test",
3   "version": "1.0.0",
4   "description": "",
5   "main": "app.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "",
10  "license": "ISC"
11 }
```

Εικόνα 47 – package.json – Node.js( το δικό μας)

Βλέπουμε παρακάτω ότι εγκαταστάθηκε με επιτυχία(added 7 packages) και στην επόμενη εικόνα βλέπουμε και την αλλαγή στο package.json αυτόματα(dependencies:chalk).

```
PS C:\Users\User\Desktop\test> npm install chalk --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN test@1.0.0 No description
npm WARN test@1.0.0 No repository field.

+ chalk@2.4.1
added 7 packages in 2.392s
```

**Εικόνα 48 – package install – Node.js( το δικό μας)**

```
{
  "dependencies": {
    "chalk": "^2.4.1"
  }
}
```

**Εικόνα 49 – chalk – Node.js( το δικό μας)**

### 4.3.5 Foursquare (API)

#### 4.3.5.1 Ιστορική Αναδρομή

Το Foursquare ξεκίνησε τον Μάρτιο του 2009 από τους Dennis Crowley και Naveen Selvadurai, εξυπηρετώντας 100 πόλεις στο σύνολο όλου του κόσμου [46].

Τον Σεπτέμβριο του 2010, έγινε αναβάθμιση στην έκδοση 2.0, δίνοντας στους χρήστες τη δυνατότητα όχι απλά να κάνουν check-in, αλλά και να κατευθύνονται σε νέες τοποθεσίες και δραστηριότητες [46].

Τον Ιανουάριο του 2010, επέτρεψε να γίνονται check-in από οποιαδήποτε τοποθεσία στον κόσμο [46].

Τον Φεβρουάριο του 2011, έφτασε τους επτά εκατομμύρια διαφορετικούς χρήστες και τον Αύγουστο του ίδιου έτους το αξιοποίηση μέχρι και ο πλανητάρχης δημόσια [46].

Τον Οκτώβριο του 2012, η έκδοση 4.0 ήταν διαθέσιμη για τα iPhone και ακολούθησαν οι υπόλοιπες πλατφόρμες [46].

Τον Αύγουστο του 2014, διαμορφώθηκε η έκδοση 8.0, στην οποία αφαιρέθηκε πλήρως η δυνατότητα δημιουργίας check-in και επικεντρώθηκε στην τοπική αναζήτηση [46].

Το 2017, προσέθεσε πάνω από πενήντα διαφορετικούς ρόλους. Στα προηγούμενα τρία χρόνια η εταιρεία κλιμακώθηκε σε ποσοστό 50% [46].

#### 4.3.5.2 Τι είναι

Ας προχωρήσουμε στον διαχωρισμό ανάμεσα στην ίδια την εταιρεία και το API της. Το Foursquare είναι με εταιρεία τεχνολογίας, η οποία χρησιμοποιεί την τοποθεσία ώστε να θεμελιώσει εμπειρίες γεμάτες νόημα για τους πελάτες και επίσης σε δεύτερο επίπεδο να προσφέρει επιχειρηματικές λύσεις.

Για τους καταναλωτές(πελάτες/consumers), διαθέτει δύο εφαρμογές , το Foursquare City Guide και το Foursquare Swarm. Η πρώτη δίνει τη δυνατότητα εξερεύνησης νέων τοποθεσιών και προσφέρει προτάσεις από μία αξιόπιστη κοινότητα. Ιδιαίτερο ενδιαφέρον παρουσιάζει η εφαρμογή Swarm, η οποία διατηρεί τα ίχνη από όλα τα μέρη που έχει επισκεφθεί και όντας συνεχώς μαζί σου, σε κινητοποιεί για νέες και πιο ενδιαφέρουσες εμπειρίες μέσα από την πραγματοποίηση ταξιδιών [47].

Όσον αφορά τις επιχειρήσεις, τους δίνεται η δυνατότητα να επικοινωνήσουν με τους δικούς τους πελάτες και να συνειδητοποιήσουν το μέγεθος της απήχησης που διαθέτουν προς αυτούς. Διατίθεται πλήθος λύσεων, οι οποίες περιλαμβάνουν το Pinpoint by Foursquare, Attribution by Foursquare, το Place Insights by Foursquare και το Pilgrim SDK by Foursquare, μέσω των οποίων οι ίδιες οι επιχειρήσεις μπορούν να κατανοήσουν και να συνδεθούν με τα κοινά-στόχους (targeted audiences), να μετρήσουν την κίνηση και την επιτυχία που ανακύπτει από τις διαφημίσεις [47].

Ας προχωρήσουμε στη συνεισφορά του Foursquare προς τους developers, το API δηλαδή που προσφέρει, μιας και είναι αυτό που μας ενδιαφέρει για την υλοποίηση της εφαρμογής μας. Το API του αποτελεί την διεπαφή η οποία επιτρέπει την ανταλλαγή δεδομένων είτε μέσα από μία εφαρμογή είτε μέσω μίας βιβλιοθήκης, είτε μέσω ενός απλού request σε ένα url. [48] Πρώτο βήμα για τη σύνδεση και την αξιοποίηση της διεπαφής από έναν προγραμματιστή, είναι η δημιουργία ενός λογαριασμού developer, η δημιουργία μίας εφαρμογής με βάση το url, η επιλογή πακέτου και τέλος η απόκτηση του Id και του Secret Key, απαραίτητα για να υπάρξει μια αντιστοίχιση των υπηρεσιών που προσφέρει το Foursquare με έναν φυσικό χρήστη, ώστε να υπάρξει φυσικά η ανάλογη τιμολόγηση. Επόμενο βήμα πλέον αποτελεί η πραγματοποίηση της πρώτης κλήσης προς το API για την ανάκτηση των επιθυμητών δεδομένων.

#### 4.3.5.3 Πώς λαμβάνω δεδομένα από το API

[48] Ο developer μπορεί να λάβει τα δεδομένα με τους εξής τρόπους( διαφορετικές γλώσσες):

- Curl, πραγματοποίησης δηλαδή request διαμέσου της γραμμής εντολών (terminal)
- Μέσω Python
- Μέσω Node.js, το οποίο μπορεί να γίνει με δύο τρόπους 1)με κανονικό request, 2) με κάποια βιβλιοθήκη/ πακέτο από το npm( Εμείς χρησιμοποιήσαμε και τους δύο τρόπους).

Το αποτέλεσμα (response) που λαμβάνεται είναι το ίδιο στις παραπάνω περιπτώσεις με μορφή JSON, όπως και φαίνεται παρακάτω .

```
// Response shortened to display relevant information
{
  "meta": {
    "code": 200,
    "requestId": "5991c2dbdd57972dfdf5831b"
  },
  "response": {
    "headerLocation": "SoHo",
    "headerFullLocation": "SoHo, New York",
```

#### Εικόνα 50 – Foursquare response - <https://developer.foursquare.com/docs/api/venues/details>

Μετά μένει η επιλογή ποιου στοιχείο χρειάζεται να ανακτηθεί. Για παράδειγμα στο συγκεκριμένο παράδειγμα εάν επιθυμούσαμε να πάρουμε το όνομα 'Soho', θα γράφαμε response.headerLocation, διότι response είναι το πατρικό κλαδί από εκεί δηλαδή που ξεκινάνε εμφωλευμένα τα υπόλοιπα δεδομένα και μέσω της τελείας φτάνουμε στο μέσα επίπεδο και παίρνουμε τελικά το όνομα.

#### 4.3.5.4 Ένα απλό παράδειγμα

Ας διαμορφώσουμε σταδιακά ένα απλό αίτημα ώστε να πάρουμε τα μέρη που είναι καφετέριες και είναι κοντά στην Αθήνα.

- Βάση request <https://api.foursquare.com/v2/venues/explore> . Με τη λέξη κλειδί explore εξερευνάς τις προτάσεις για μέρη/πόλεις. Υπάρχουν πολλές άλλες επιλογές όπως για παράδειγμα το search( αναζητεί πληροφορίες για ένα συγκεκριμένο id μέρους) και το photos( μέσω του οποίου λαμβάνονται αποκλειστικά φωτογραφίες για ένα συγκεκριμένο μέρος)
- Επιπρόσθετοι παράμετροι του request, όπως το near, σε ποια πόλη βρίσκεται δηλαδή κοντά, το query το οποίο είναι οποιαδήποτε κατηγορία(food, shopping,sushi κλπ.).
- Απαραίτητοι παράμετροι είναι το id και το secret key για την αυθεντικοποίηση του request.
- Τελική μορφή του request

[https://api.foursquare.com/v2/venues/explore?near=Athens,GR  
&query=coffee&client\\_id=ID &client\\_secret=SECRET&v=20180516](https://api.foursquare.com/v2/venues/explore?near=Athens,GR&query=coffee&client_id=ID &client_secret=SECRET&v=20180516)

Όπως φαίνεται η τιμή της παραμέτρου near είναι η Αθήνα, το query είναι ο καφές, το id και το secret είναι του κάθε χρήστη διαφορετικά (προσωπικά μας δεδομένα) και στο v αναθέτεις τη σημερινή ημερομηνία.

#### 4.3.6 Google Geocoding API

##### 4.3.6.1 Τι είναι

Geocoding είναι η διαδικασία κατά την οποία φυσικές διευθύνσεις όπως για παράδειγμα Αιγάλεω, Ευριπίδου 24 μετατρέπεται σε συντεταγμένες. Όταν πραγματοποιείται το αντίστροφο, δηλαδή από

συντεταγμένες να ανακτήσουμε τη φυσική διεύθυνση χρησιμοποιούμε τον όρο reverse geocoding [49] .

#### 4.3.6.2 Πώς και γιατί το χρησιμοποιούμε

Όπως και σε όλα τα API, και σε αυτό χρειαζόμαστε μία μορφή αυθεντικοποίησης και συγκεκριμένα ένα key κι αφού το λάβουμε με την παρακάτω διαδικασία είμαστε έτοιμοι να το χρησιμοποιήσουμε.

1. Go to the [Google API Console](#).
2. Create or select a project.
3. Click **Continue** to enable the API.
4. On the **Credentials** page, get an **API key**.  
Note: If you have an existing unrestricted API key, or a key with server restrictions, you may use that key.
5. From the dialog displaying the API key, select **Restrict key** to set a server restriction on the API key.
6. In the **Key restriction** section, select **'IP addresses (web servers, cron jobs, etc.)**, follow the on-screen instructions to add server IP addresses, then click **Save**. Read more about [restricting API keys](#).
7. (Optional) Enable billing. See [Usage Limits](#) for more information.

#### Εικόνα 51 – geocoding steps

- <https://developers.google.com/maps/documentation/geocoding/get-api-key>

Το χρησιμοποιούμε, διότι αφού λάβουμε την τοποθεσία του χρήστη, η οποία ανάγεται σε συντεταγμένες, πρέπει να καταλάβουμε την ευρύτερη περιοχή που βρίσκεται (π.χ. Αιγάλεω), ώστε να την αξιοποιήσουμε στο request προς το Foursquare και συγκεκριμένα να την αναθέσουμε στη μεταβλητή near, όπως αναλύσαμε στην προηγούμενη επικεφαλίδα. Τέλος η χρήση γίνεται μέσω ενός προσφερόμενου πακέτου (reverse-geocoding-google) διαμέσου του node packet manager.

### 4.3.7 Messenger Platform

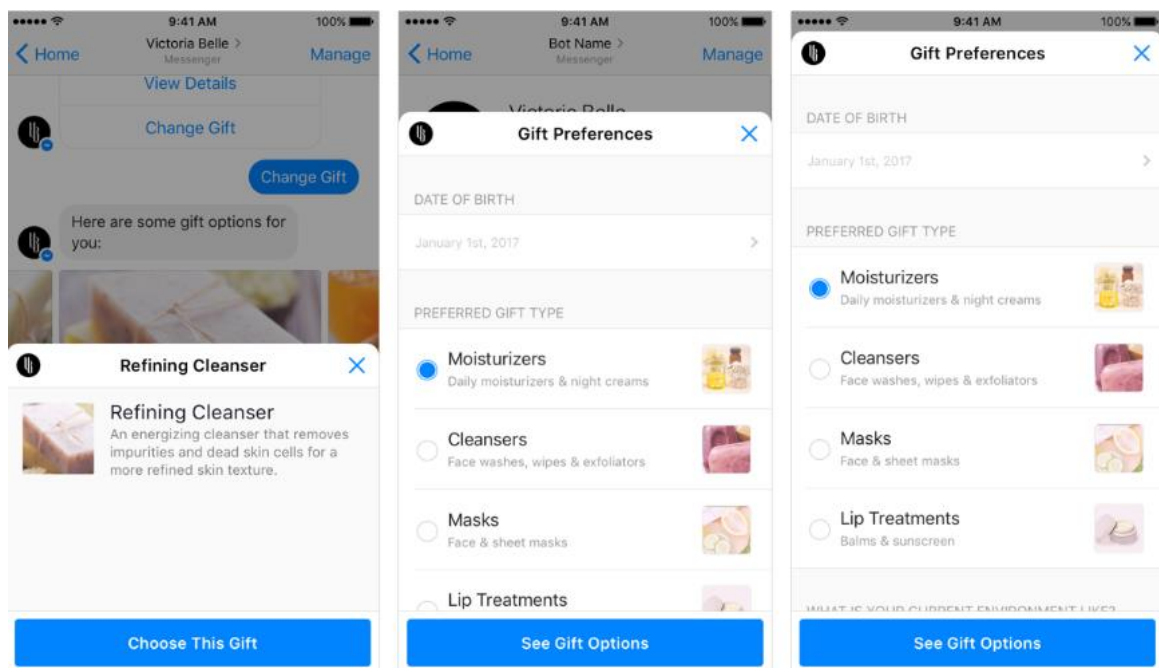
[50]Συνιστά την εργαλειοθήκη ενός developer για τη δημιουργία bots και αποτελείται από διάφορα χαρακτηριστικά, τα οποία και θα αναλύσουμε παρακάτω.

#### 4.3.7.1 Messaging

Μέσω του συγκεκριμένου χαρακτηριστικού, γίνεται δυνατή η λήψη και αποστολή μηνυμάτων και πολυμέσων [50].

#### 4.3.7.2 Webview

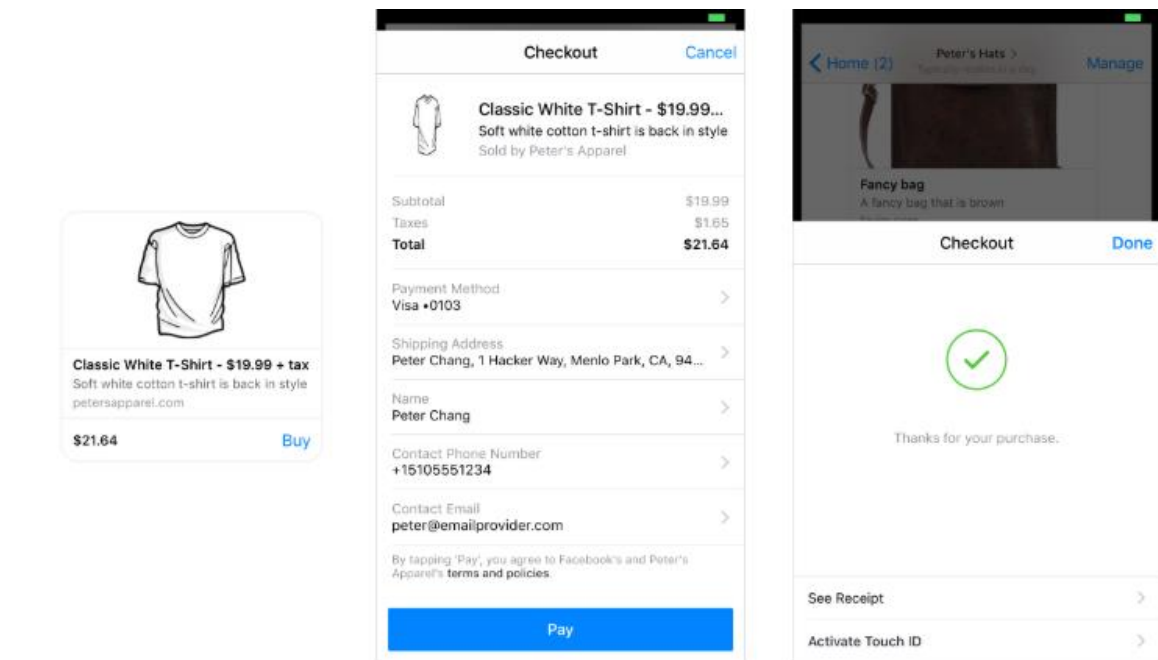
Αξιοποιείται για να παρουσιαστεί στον χρήστη μία όμορφη γραφική απεικόνιση που θα τον βοηθήσει γενικότερα, όταν μία επιλογή είναι πολυσύνθετη( π.χ. η επιλογή για subscription σε πολλά διαφορετικά θέματα σε ένα bot ενημέρωσης) [51]. Παρακάτω φαίνεται μία εικόνα για να επέλθει καλύτερη κατανόηση.



Εικόνα 52 - Messenger Platform Webview - <https://developers.facebook.com/docs/messenger-platform/webview>

#### 4.3.7.3 Payments

Μπορούν πλέον να γίνουν αγορές μέσα από τη πλατφόρμα του messenger με τον πιο εύκολο κι άμεσο τρόπο( στο πλαίσιο μιας σελίδας/εφαρμογής/bot) [52].



**Εικόνα 53 - Messenger Platform Payments –**

<https://developers.facebook.com/docs/messenger-platform/payments>

#### 4.3.7.4 Discovery

Μέσω αυτού του χαρακτηριστικού δίνεται η δυνατότητα να προωθηθεί το bot μέσα από ένα πλαίσιο διαφημίσεων προς τον χώρο του Facebook, Instagram, Messenger και επίσης η δυνατότητα αναζήτησης του bot μέσα από το Messenger, όπως ισχύει και στη δικιά μας περίπτωση. ( σαν να αναζητείς ένα φίλο σου) [53].

#### 4.3.7.5 IDs & Profile Information

Προσδίδει προσωπικό χαρακτήρα στις συνομιλίες, καθώς επιτυγχάνεται η διασύνδεση με τον χρήστη με την ταύτιση του ID του και διαμορφώνεται το ιδιαίτερο προφίλ του [54].

#### 4.3.7.6 NLP

Αντί της ύπαρξης ενός NLP API, η messenger πλατφόρμα δίνει τη δυνατότητα στον developer να χρησιμοποιήσει έναν εμφωλευμένο NLP, μέσω του οποίου γίνεται αυτόματα η ανίχνευση των οντοτήτων, χωρίς να χρειάζεται επιπλέον αίτημα προς την υπηρεσία με το οποίο επέρχεται πιθανή καθυστέρηση και ασύγχρονα προβλήματα κατά την επικοινωνία [55].



#### 4.3.7.7 Analytics

Από τη στιγμή που έχει δημοσιευθεί το bot, μπορεί ο διαχειριστής να δει κάποια στατιστικά δεδομένα είτε μέσω της σελίδας στο facebook, είτε μέσα από το διαχειριστικό περιβάλλον στο facebook developers [56].

#### 4.3.7.8 Webhooks

Είναι το βασικό σημείο της πλατφόρμας, διότι εκεί υπάρχει ο κώδικας και εκεί επίσης θα ληφθούν τα επεξεργασμένα και θα αποσταλούν τα μηνύματα από τον χρήστη. [57] Τα Webhooks είναι ουσιαστικά ορισμένα από τον χρήστη HTTP callbacks, τα οποία πυροδοτούνται από ένα συγκεκριμένο γεγονός. Αναλόγως σε τι webhook θα γίνει συνδρομή (δηλαδή θα χρησιμοποιηθεί), θα μπορεί η λειτουργικότητα του bot να αναβαθμιστεί και να γίνει πιο σύνθετη. Για παράδειγμα μπορεί να χρησιμοποιεί το webhook messages για να μπορούν να υπάρξουν συνομιλίες ανάμεσα σε χρήστες και στο bot, αλλά επίσης μπορεί να έχει γίνει συνδρομή στο webhook messaging\_payments, το οποίο είναι ένα απαραίτητο άκρο για να πραγματοποιούνται οι πληρωμές από την πλευρά του χρήστη μέσα στο πλαίσιο λειτουργίας του bot. Παρακάτω φαίνονται τα υπαρκτά webhooks που είναι διαθέσιμα μέσω της πλατφόρμας [58].

| Webhook Event                                  | Description  |
|--|--|
| <code>messages</code>                          | Subscribes to Message Received events                  |
| <code>messaging_account_linking</code>         | Subscribes to Account Linking events                   |
| <code>messaging_checkout_updates</code> (beta) | Subscribes to Checkout Update events                   |
| <code>message_deliveries</code>                | Subscribes to Message Delivered events                 |
| <code>message_echoes</code>                    | Subscribes to Message Echo events                      |
| <code>messaging_game_plays</code>              | Subscribes to Instant Game events                      |
| <code>messaging_handovers</code> (beta)        | Subscribes to Handover Protocol events                 |
| <code>messaging_optins</code>                  | Subscribes to Plugin Opt-in events                     |
| <code>messaging_payments</code> (beta)         | Subscribes to Payment events                           |
| <code>messaging_policy_enforcement</code>      | Subscribes to Policy Enforcement events                |
| <code>messaging_postbacks</code>               | Subscribes to Postback Received events                 |
| <code>messaging_pre_checkouts</code> (beta)    | Subscribes to Payment Pre-checkout events              |
| <code>message_reads</code>                     | Subscribes to Message Read events                      |
| <code>messaging_referrals</code>               | Subscribes to Referral events                          |
| <code>standby</code> (beta)                    | Subscribes to Handover Protocol Standby Channel events |

Εικόνα 54 - Messenger Platform Webhooks –

<https://developers.facebook.com/docs/messenger-platform/webhook>

## 4.3.8 Visual Studio Code

### 4.3.8.1 Ιστορική Αναδρομή

Ανακοινώθηκε στις 29 Απρίλη του 2015 από τη Microsoft στο συνέδριο των developers της Microsoft, Build.

Στις 18 Νοέμβρη του ίδιου έτους δημοσιεύτηκε, έχοντας την άδεια MIT και ο πηγαίος κώδικας τοποθετήθηκε στο Github, που εδρεύει το ανοιχτό λογισμικό .

Στις 14 Απριλίου του 2016, δημοσιεύτηκε στο διαδίκτυο.

### 4.3.8.2 Τι είναι

Αποτελεί όπως είναι και το Visual Studio, έναν editor κώδικα, ο οποίος διαθέτει ένα πλήθος γλωσσών προγραμματισμού

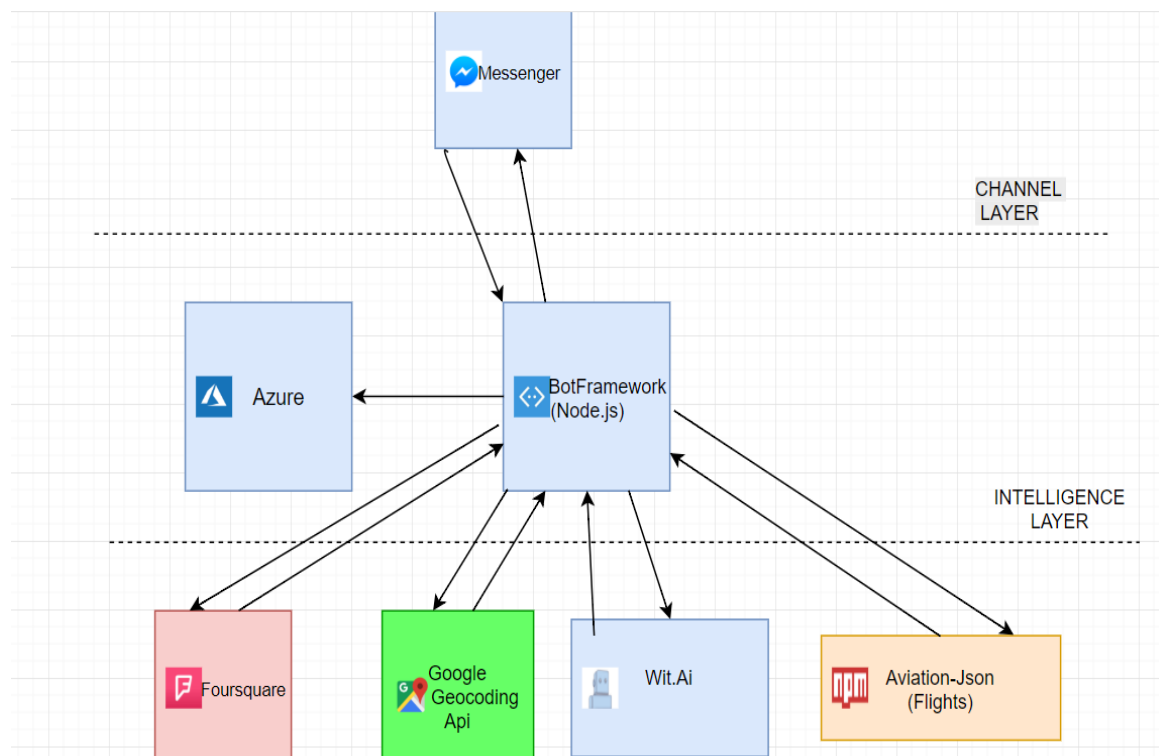
## 4.4 Προβλήματα και περιορισμοί εργαλείων

Τα προβλήματα και οι περιορισμοί των εργαλείων που χρησιμοποιήσαμε αφορούν μόνο την Javascript και την πλατφόρμα NLP, το Wit.ai. Σε πρώτο επίπεδο, στοχεύαμε να παίρνουμε τα αποτελέσματα από το Foursquare API και να τα προβάλλουμε στον χρήστη, μόνο με το όνομα του κάθε μέρους, χωρίς περισσότερες λεπτομέρειες, ώστε να σιγουρευτούμε ότι μπορούμε να επεκτείνουμε την εφαρμογή με μία βασική σωστή λειτουργικότητα. Η γλώσσα της Javascript δε μας δημιούργησε κάποιο περιορισμό στην συγκεκριμένη περίπτωση, και είχαμε ως response όλα τα απαραίτητα στοιχεία που χρειαζόμασταν σε μορφή JSON. Ωστόσο, στη συνέχεια θελήσαμε να προσθέσουμε και μία φωτογραφία για το κάθε μέρος/μαγαζί, όπως και είναι στην τρέχουσα έκδοση. Στέλνοντας το ίδιο request προς το Foursquare, λαμβάναμε ως απάντηση φωτογραφία για το κάθε μέρος, αλλά όταν προσπαθούσαμε να την προβάλλουμε, δεν παρατηρούσαμε κάποιο αποτέλεσμα. Εν τέλει, διακρίναμε μέσω από την βοηθητική κονσόλα του Node.js (τερματικό), ότι η φωτογραφία ήταν undefined, δεν είχε οριστεί δηλαδή, επιδεικνύοντας μας ότι δεν υπήρχε σαν αποτέλεσμα εξ' αρχής. Το θέμα ωστόσο επιλύθηκε, χρησιμοποιώντας ένα ακόμη επιπρόσθετο request, το οποίο αποσκοπούσε μόνο στην λήψη των φωτογραφιών και όχι περισσότερων πληροφοριών. Το πρόβλημα αυτό συνέβαινε λόγω της ασύγχρονης φύσης της διαδικασίας που προσπαθούσαμε να επιτύχουμε( να στείλουμε δηλαδή το request, να παραλάβουμε τις γενικές πληροφορίες, τις φωτογραφίες, να τα ορίσουμε ως μεταβλητές και να τα προβάλλουμε στον χρήστη). Έτσι, χρησιμοποιώντας παράλληλα δύο διαφορετικά request ( το ένα για το όνομα, τηλέφωνο, διεύθυνση και το άλλο για την φωτογραφία του κάθε μέρους) και συνδυάζοντας τα με ένα πακέτο με την ονομασία async, μέσω του Npm, καταφέραμε να προβάλλουμε στον χρήστη τα πέντε μέρη που επιθυμούσαμε με μία φωτογραφία που χαρακτηρίζει το καθένα. Σε δεύτερο επίπεδο, στην πλατφόρμα του Wit.ai συναντήσαμε αρχικά τον εξής περιορισμό. Χρησιμοποιώντας οντότητες ( entities) που λειτουργούσαν με βάση κάποια χαρακτηριστικά ( traits), και έχοντας φυσικά επιλέξει τη ρύθμιση για την κατανόηση της ελληνικής γλώσσας, παρατηρήσαμε μία δυσκολία της πλατφόρμας στο να συνειδητοποιεί για ποιο θέμα ο χρήστης ομιλούσε ( intent). Για

παράδειγμα δηλαδή, όταν έστειλε ‘θέλω να φάω κάτι’ και καταλάβαινε ότι το θέμα της συζήτησης είναι φαγητό, εάν άλλαζαν λίγο οι τόνοι ή η σειρά της πρότασης, δεν ήταν σίγουρο ότι ο χρήστης μιλούσε για φαγητό και υπήρχαν φορές που εντόπιζε διαφορετική, μη σχετική οντότητα. Φυσικά, αυτό το θέμα επιλύθηκε αλλάζοντας την λειτουργία των οντοτήτων από trait σε keywords. Έτσι με τις λέξεις κλειδιά μας δινόταν η δυνατότητα να τοποθετήσουμε όλες τις λέξεις που θεωρούσαμε εμείς κατάλληλες με τόνους και χωρίς, ώστε να καλυφθούν όλες οι πιθανότητες. Επιπρόσθετα, άλλος ένας περιορισμός που συναντήσαμε είναι ο εξής. Στην εφαρμογή μας υπάρχουν για παράδειγμα οι οντότητες Φαγητό και υποκατηγορίες των οντοτήτων (π.χ. burger). Όταν ο χρήστης γράψει ‘θέλω να φάω’ θα καταλάβει το λογισμικό NLP, ότι μιλά γενικότερα για φαγητό κι όλα κυλάνε σωστά. Στην περίπτωση που θα γράψει κάτι σχετικό με ‘θέλω να έχω ένα burger’, θα κατανοήσει ότι η συνομιλία έχει σχέση με την υποκατηγορία burger κι όχι την γενικότερη του φαγητού, γεγονός που και πάλι είναι σωστότατο. Ωστόσο, εάν στείλει ένα μήνυμα που να περιλαμβάνει τις λέξεις κλειδιά και των δύο οντοτήτων, όπως για παράδειγμα θέλω να φάω ( πυροδοτεί την οντότητα Φαγητό) burger ( πυροδοτεί την οντότητα burger), το Wit.ai δεν γνωρίζει ποια από τις δύο οντότητες είναι το σωστό θέμα, οπότε πυροδοτεί και τις δύο, με αποτέλεσμα το bot ( ο κώδικας μας) να μην μπορεί να αναγνωρίσει το σωστό. Συμπερασματικά, από τα περισσότερα εργαλεία δεν προέκυψε κάποιος περιορισμός που να μας δημιουργήσει πρόβλημα, αλλά σε αυτά που αναλύθηκαν παραπάνω τα προβλήματα στον περισσότερο βαθμό καταπολεμήθηκαν.

#### 4.5 Διάγραμμα αρχιτεκτονικής

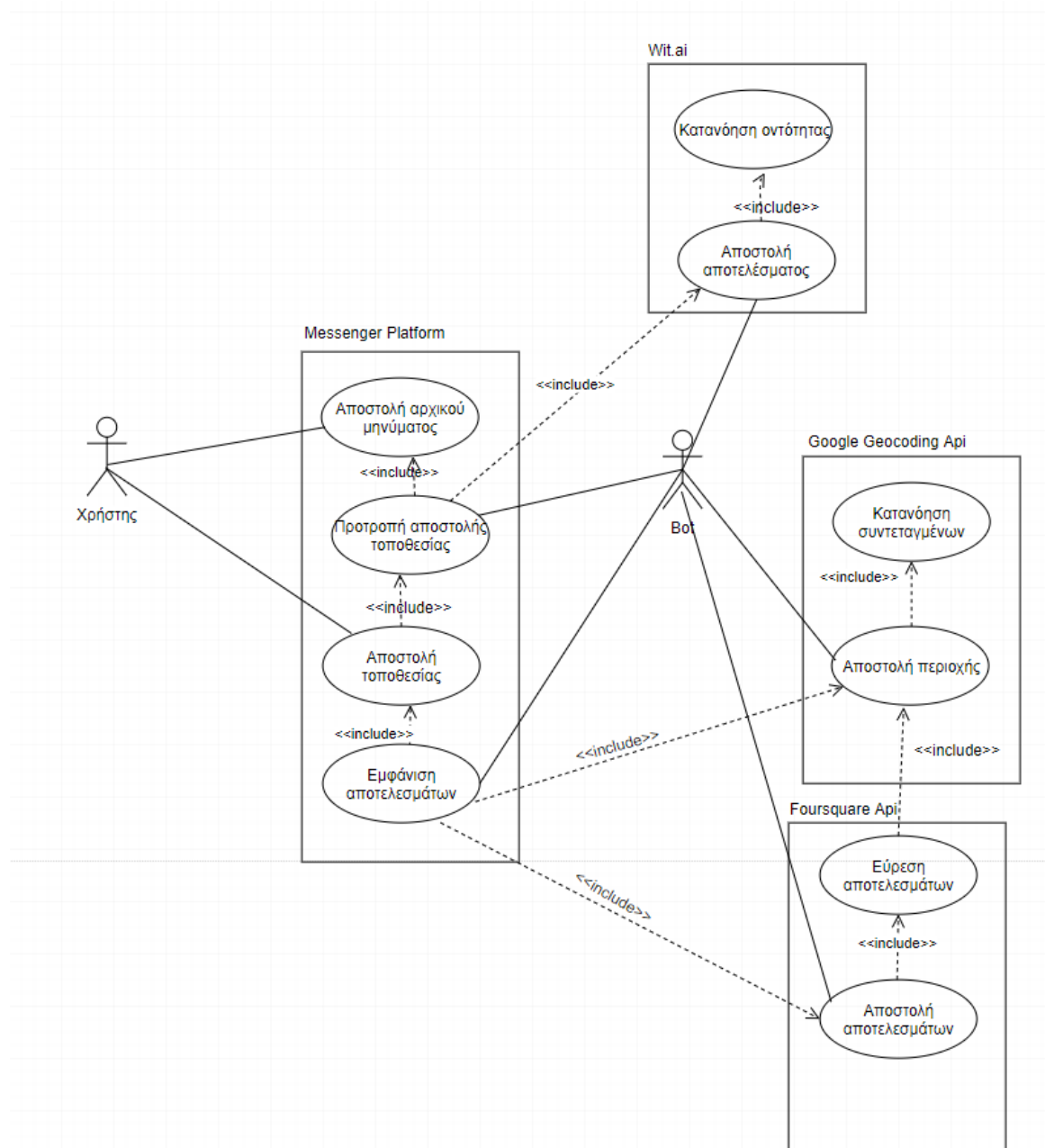
- Channel Layer: Τα κανάλια που διασυνδέεται το Bot Framework(Messenger).
- Intelligence Layer: Η νοημοσύνη του bot μας. Ειδικότερα το Wit.ai είναι υπεύθυνο για την κατανόηση της γλώσσας, το Google Geocoding Api για την τοποθεσία και τα υπόλοιπα δύο συνιστούν την πηγή των δεδομένων μας. (Aviation-json πακέτο του npm με δεδομένα πτήσεων και αεροδρομίων).



# ΚΕΦΑΛΑΙΟ 5-ΑΠΑΙΤΗΣΕΙΣ ΒΟΤΝΟΥΑΓΕ

## 5.1 Διαγράμματα

### 5.1.1 Use Case Diagrams



Ο χρήστης κάνει αποστολή αρχικού μηνύματος μέσω της πλατφόρμας του Messenger. Στη συνέχεια το bot προχωρά σε προτροπή αποστολή τοποθεσίας ('send location'), αλλά για να γίνει αυτό λόγω της συσχέτισης include, χρειάζεται να γίνει η αποστολή του αποτελέσματος μέσω της πλατφόρμας του Wit.ai. Ωστόσο για να γίνει η αποστολή μέσω συσχέτισης include, πρέπει πρώτα να γίνει η κατανόηση οντότητας. Για παράδειγμα δηλαδή, εάν ο χρήστης γράψει πεινάω θα μεταφραστεί σε Φαγητό και θα αποσταλθεί, οπότε θα είναι έτοιμο το bot να στείλει την προτροπή για τοποθεσία. Στη συνέχεια ο χρήστης έχει το δικαίωμα να στείλει την τοποθεσία η οποία και έχει πάλι συσχέτιση include με την παραπάνω, αφού έπεται αυτής. Επιπρόσθετα, το bot για να κάνει την τελική εμφάνιση των αποτελεσμάτων, πρέπει να γίνουν πρώτα δύο διαφορετικά γεγονότα. Αρχικά πρέπει μέσω Google Geocoding Api να μετατραπούν οι συντεταγμένες σε μία ονομασία περιοχής και να αποσταλθεί στο bot. Σε δεύτερο επίπεδο, μέσω του Foursquare Api, γίνεται η αποστολή αποτελεσμάτων, αφού βέβαια γίνει πρώτα η εύρεση αυτών. Φυσικά η εύρεση των αποτελεσμάτων συσχετίζεται με include με την αποστολή περιοχής, αφού χρειάζεται να υπάρχει πρώτα η τοποθεσία ως δεδομένο για να εμπεριεχθεί στο request προς το Foursquare. Τέλος, συνδέεται με την αποστολή αποτελεσμάτων και με την εμφάνιση οπότε και μέσω της πλατφόρμας του Messenger, απεικονίζονται στον χρήστη.

## 5.2 Απαιτήσεις συστήματος

### 5.2.1 Σύνολο απαιτήσεων

1. Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.
2. Ο χρήστης θα πρέπει να είναι είτε χρήστης του Facebook, είτε του Messenger.
3. Δύναται η πρόσβαση σε τοποθεσία( GPS Receiver), αναλόγως τι θα επιλέξει ο χρήστης κάθε φορά.
4. Ο χρήστης θα πρέπει να προχωρήσει στην επιλογή της γλώσσας μία φορά.
5. Ο χρήστης θα πρέπει να έχει τη δυνατότητα να δει εξαρχής όλες τις λειτουργικές επιλογές της εφαρμογής.
6. Ο χρήστης θα πρέπει να επιλέγει μία δυνατότητα κάθε φορά.
7. Θα πρέπει να δίνεται η δυνατότητα στον χρήστη να αλλάξει την επιλογή του πριν την ολοκλήρωση της ήδη υπάρχουσας.
8. Θα πρέπει να υπάρχει δυνατότητα στον χρήστη reset του διαλόγου και των στοιχείων του για περίπτωση λάθους.
9. Το σύστημα θα πρέπει να ανακατευθύνει τον χρήστη στους ιστότοπους για πραγματοποίηση συναλλαγών και ενημέρωσης.
10. Το σύστημα θα πρέπει να διαμορφώνει ένα profile για τον κάθε χρήστη για τη διευκόλυνση της αλληλεπίδρασής του.
11. Το σύστημα θα πρέπει να παρέχει στον χρήστη μία κατανοητή ροή, μέσα από το γραφικό του περιβάλλον.
12. Το σύστημα θα πρέπει να κατανοεί τη φυσική γλώσσα του χρήστη.
13. Το σύστημα θα πρέπει να συνδέεται σε ένα από τα api και να επιστρέφει τα δεδομένα που ζητήθηκαν από τον χρήστη.
14. Το σύστημα θα πρέπει ανά διαστήματα να ενημερώνει τον χρήστη για γεγονότα με βάση το ιστορικό του profile του.
15. Το σύστημα θα πρέπει ανάλογα την τοποθεσία του χρήστη να επισκέπτεται και διαφορετικό api ( π.χ. Yelp, Αμερική / Foursquare, Ευρώπη).
16. Το σύστημά μας θα πρέπει να αποκρίνεται το συντομότερο δυνατό, γεγονός που κρίνεται από την επιλογή διακομιστή.
17. Ο διακομιστής μας θα ανήκει στο Azure της Microsoft.
18. Η διασύνδεση χρήστη θα πρέπει να υλοποιηθεί μέσω της JavaScript, σε Node.js περιβάλλον.
19. Το ιστορικό του διαλόγου του χρήστη θα πρέπει να αποθηκεύεται με ασφάλεια στη βάση δεδομένων μας.

20. Τα δεδομένα που θα παραχωρούνται στον χρήστη θα πρέπει να έχουν προέλευση από αξιόπιστες πηγές.
21. Το σύστημα θα πρέπει να ενημερώνει τον χρήστη σε περιπτώσεις λάθους, ώστε να υπάρξει η κατάλληλη χρηστικότητα.
22. Το σύστημα δε θα πρέπει να παραβιάζει το privacy policy των api, όσο αναφορά την κατακράτηση δεδομένων.
23. Το σύστημα θα πρέπει να ακολουθεί τη λογική ροή ήδη υπαρχόντων bot, ώστε να είναι πιο εύκολη η εμπειρία χρήσης.
24. Το σύστημα θα δίνει τη δυνατότητα στον χρήστη να παρακολουθήσει μία παραδειγματική ροή της εφαρμογής.
25. Το σύστημα δε θα περιλαμβάνει κάποιο χρονικό περιορισμό όσο αναφορά την είσοδο του χρήστη.

## 5.2.2 Πίνακες απαιτήσεων

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 1   |
| Όνομα απαίτησης   | Σύνδεση στο διαδίκτυο                     |
| Περιγραφή         | Ο χρήστης πρέπει να είναι διασυνδεδεμένος |
| Προτεραιότητα     | V1  |
| Χρήστες           | Διαχειριστής, Χρήστες                     |
| Σχόλια            |   |

**Πίνακας 2 - Απαίτηση 1**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 2  |
| Όνομα απαίτησης   | Κατοχή Messenger/Facebook                                      |
| Περιγραφή         | Ο χρήστης πρέπει να είναι χρήστης ενός από τα δύο τουλάχιστον. |
| Προτεραιότητα     | V1   |
| Χρήστες           | Διαχειριστής, Χρήστες  |
| Σχόλια            |  |

**Πίνακας 3 - Απαίτηση 2**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 3 |
|-------------------|---|

|                 |   |
|-----------------|---|
| Όνομα απαίτησης | Έγκριση αποστολής τοποθεσίας                        |
| Περιγραφή       | Πρέπει να αποστέλλεται η τοποθεσία για αποτελέσματα |
| Προτεραιότητα   | V2  |
| Χρήστες         | Χρήστες   |
| Σχόλια          | Στις πτήσεις δε θα υπάρχει                          |

**Πίνακας 4 - Απαίτηση 3**

|                   |                             |
|-------------------|-----------------------------|
| Κωδικός απαίτησης | 4                           |
| Όνομα απαίτησης   | Επιλογή γλώσσας             |
| Περιγραφή         | Ο χρήστης επιλέγει μία φορά |
| Προτεραιότητα     | V3                          |
| Χρήστες           | Χρήστες                     |
| Σχόλια            |                             |

**Πίνακας 5 - Απαίτηση 4**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 5  |
| Όνομα απαίτησης   | Εμφάνιση επιλογών                            |
| Περιγραφή         | Ο χρήστης πρέπει να βλέπει όλες τις επιλογές |
| Προτεραιότητα     | V2   |
| Χρήστες           | Χρήστες                                      |
| Σχόλια            | Μέσω 'Help'                                  |

**Πίνακας 6 - Απαίτηση 5**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 6   |
| Όνομα απαίτησης   | Επιλογή λειτουργίας                         |
| Περιγραφή         | Ο χρήστης πρέπει να διαλέγει μία τη φορά    |
| Προτεραιότητα     | V1  |
| Χρήστες           | Χρήστες                                     |
| Σχόλια            | Να γίνεται 'triggered' μόνο μία κάθε στιγμή |



**Πίνακας 7 - Απαίτηση 6**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 7   |
| Όνομα απαίτησης   | Αλλαγή λειτουργίας  |
| Περιγραφή         | Ο χρήστης να αλλάζει πριν τελειώσει κάποια άλλη           |
| Προτεραιότητα     | V1  |
| Χρήστες           | Χρήστες   |
| Σχόλια            | Να πυροδοτείται η καινούρια λειτουργία οποιαδήποτε στιγμή |

**Πίνακας 8 - Απαίτηση 7**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 8  |
| Όνομα απαίτησης   | Reset Διαλόγου                             |
| Περιγραφή         | Start Over σε περίπτωση λάθους/μπερδέματος |
| Προτεραιότητα     | V2   |
| Χρήστες           | Χρήστες                                    |
| Σχόλια            | Με ύπαρξη κουμπιού                         |

**Πίνακας 9 - Απαίτηση 8**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 9   |
| Όνομα απαίτησης   | Ανακατεύθυνση σε κύριο ιστότοπο                               |
| Περιγραφή         | Redirect με στόχο την ενημέρωση του χρήστη και για συναλλαγές |
| Προτεραιότητα     | V2  |
| Χρήστες           | Διαχειριστής, Χρήστες   |
| Σχόλια            | Για παράδειγμα, να δει τις πτήσεις                            |

**Πίνακας 10 - Απαίτηση 9**

|                   |                   |
|-------------------|-------------------|
| Κωδικός απαίτησης | 10                |
| Όνομα απαίτησης   | Διαμόρφωση προφίλ |

|               |                          |
|---------------|--------------------------|
| Περιγραφή     | Ο χρήστης να έχει προφίλ |
| Προτεραιότητα | V4                       |
| Χρήστες       | Διαχειριστής, Χρήστες    |
| Σχόλια        | Για ενημέρωση προσφορών  |

**Πίνακας 11 - Απαίτηση 10**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 11   |
| Όνομα απαίτησης   | Καλό γραφικό περιβάλλον                              |
| Περιγραφή         | Διευκόλυνση του χρήστη μέσα από εύκολη ροή εφαρμογής |
| Προτεραιότητα     | V 1  |
| Χρήστες           | Χρήστες  |
| Σχόλια            | Όχι πολλά περιττά κουμπιά                            |

**Πίνακας 12 - Απαίτηση 11**

|                   |                           |
|-------------------|---------------------------|
| Κωδικός απαίτησης | 12                        |
| Όνομα απαίτησης   | Ύπαρξη NLP/NLU            |
| Περιγραφή         | Κατανόηση φυσικής γλώσσας |
| Προτεραιότητα     | V1                        |
| Χρήστες           | Διαχειριστής, Χρήστες     |
| Σχόλια            |                           |

**Πίνακας 13 - Απαίτηση 12**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 13   |
| Όνομα απαίτησης   | Σύνδεση σε api                                     |
| Περιγραφή         | Σύνδεση και αποστολή των δεδομένων προς τον χρήστη |
| Προτεραιότητα     | V1   |
| Χρήστες           | Διαχειριστής, Χρήστες                              |
| Σχόλια            |  |

#### Πίνακας 14 - Απαίτηση 13

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 14  |
| Όνομα απαίτησης   | Ενημέρωση χρήστη                                    |
| Περιγραφή         | Αποστολή πληροφοριών με βάση προηγούμενη λειτουργία |
| Προτεραιότητα     | V4  |
| Χρήστες           | Διαχειριστής, Χρήστες                               |
| Σχόλια            |   |

#### Πίνακας 15 - Απαίτηση 14

|                   |                                       |
|-------------------|---------------------------------------|
| Κωδικός απαίτησης | 15                                    |
| Όνομα απαίτησης   | Ύπαρξη διαφορετικών api               |
| Περιγραφή         | Ανάλογα τη χώρα, διαφορετικά δεδομένα |
| Προτεραιότητα     | V3                                    |
| Χρήστες           | Διαχειριστής, Χρήστες                 |
| Σχόλια            | Foursquare, Yelp                      |

#### Πίνακας 16 - Απαίτηση 15

|                   |                                   |
|-------------------|-----------------------------------|
| Κωδικός απαίτησης | 16                                |
| Όνομα απαίτησης   | Βέλτιστη ανταπόκριση              |
| Περιγραφή         | Αποστολή μηνυμάτων σε άμεσο χρόνο |
| Προτεραιότητα     | V1                                |
| Χρήστες           | Χρήστες                           |
| Σχόλια            |                                   |

#### Πίνακας 17 - Απαίτηση 16

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 17  |
| Όνομα απαίτησης   | Επιλογή διακομιστή  |
| Περιγραφή         | Θα πρέπει να ανήκει στο Azure για να γίνονται όλες οι παραμετροποιήσεις |
| Προτεραιότητα     | V1  |
| Χρήστες           | Διαχειριστής, Χρήστες   |
| Σχόλια            | Προτεραιότητα V1, γιατί το Azure είναι αναπόσπαστο του Bot Framework    |

**Πίνακας 18 - Απαίτηση 17**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 18  |
| Όνομα απαίτησης   | Επιλογή server-side περιβάλλοντος                         |
| Περιγραφή         | Ύπαρξη Node.js και JavaScript                             |
| Προτεραιότητα     | V1  |
| Χρήστες           | Διαχειριστής  |
| Σχόλια            | Απαραίτητες τεχνολογίες για το bot (άλλη επιλογή .NET C#) |

**Πίνακας 19 - Απαίτηση 18**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 19  |
| Όνομα απαίτησης   | Ασφάλεια στοιχείων χρήστη                               |
| Περιγραφή         | Τα στοιχεία να διατηρούνται χωρίς πιθανότητες υποκλοπής |

|               |   |
|---------------|---|
| Προτεραιότητα | V2  |
| Χρήστες       | Διαχειριστής, Χρήστες   |
| Σχόλια        | Η προτεραιότητα δεν είναι 1, γιατί το προφίλ επίσης έχει χαμηλή (4) |

**Πίνακας 20 - Απαίτηση 19**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 20  |
| Όνομα απαίτησης   | Αξιόπιστα δεδομένα                        |
| Περιγραφή         | Παροχή δεδομένων από api που είναι ασφαλή |
| Προτεραιότητα     | V1  |
| Χρήστες           | Διαχειριστής                              |
| Σχόλια            | Για παράδειγμα Foursquare                 |

**Πίνακας 21 - Απαίτηση 20**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 21   |
| Όνομα απαίτησης   | Ενημέρωση για σφάλματα                                 |
| Περιγραφή         | Κατανόηση λάθους από το σύστημα και πληροφόρηση χρήστη |
| Προτεραιότητα     | V1   |
| Χρήστες           | Διαχειριστής, Χρήστες                                  |
| Σχόλια            | Απαραίτητο για την αποφυγή παρανοήσεων                 |
|                   |  |

**Πίνακας 22 - Απαίτηση 21**

|                   |                  |
|-------------------|------------------|
| Κωδικός απαίτησης | 22               |
| Όνομα απαίτησης   | Νόμιμη χρήση api |

|               |   |
|---------------|---|
| Περιγραφή     | Αποφυγή κατακράτησης στοιχείων, μη παραβίαση privacy policy |
| Προτεραιότητα | V1  |
| Χρήστες       | Διαχειριστής  |
| Σχόλια        |   |

**Πίνακας 23 - Απαίτηση 22**

|                   |   |
|-------------------|---|
| Κωδικός απαίτησης | 23  |
| Όνομα απαίτησης   | Εφάμιλλη λειτουργία                                 |
| Περιγραφή         | Απαραίτητη η παρόμοια ροή με άλλα ήδη υπάρχοντα bot |
| Προτεραιότητα     | V1  |
| Χρήστες           | Διαχειριστής, Χρήστες                               |
| Σχόλια            | Παρόμοιο UI και λειτουργίες(π. χ help)              |

**Πίνακας 24 - Απαίτηση 23**

|                   |  |
|-------------------|--|
| Κωδικός απαίτησης | 24   |
| Όνομα απαίτησης   | Ύπαρξη demo  |
| Περιγραφή         | Υποστήριξη χρήστη, δίνοντας του υλικό εκμάθησης                |
| Προτεραιότητα     | V3   |
| Χρήστες           | Χρήστες  |
| Σχόλια            | Μικρής προτεραιότητας, διότι θα είναι ξεκάθαρες οι λειτουργίες |

**Πίνακας 25 - Απαίτηση 24**

|                   |                              |
|-------------------|------------------------------|
| Κωδικός απαίτησης | 25                           |
| Όνομα απαίτησης   | Απουσία χρονικού περιορισμού |

|               |  |
|---------------|--|
| Περιγραφή     | Ο χρήστης μπορεί να στείλει όποτε θέλει μήνυμα, χωρίς περιορισμούς |
| Προτεραιότητα | V2   |
| Χρήστες       | Χρήστες  |
| Σχόλια        | Απαραίτητο για την αποφυγή παρανοήσεων                             |

**Πίνακας 26 - Απαίτηση 25**

## ΚΕΦΑΛΑΙΟ 6-ΤΕΧΝΙΚΗ ΑΝΑΛΥΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

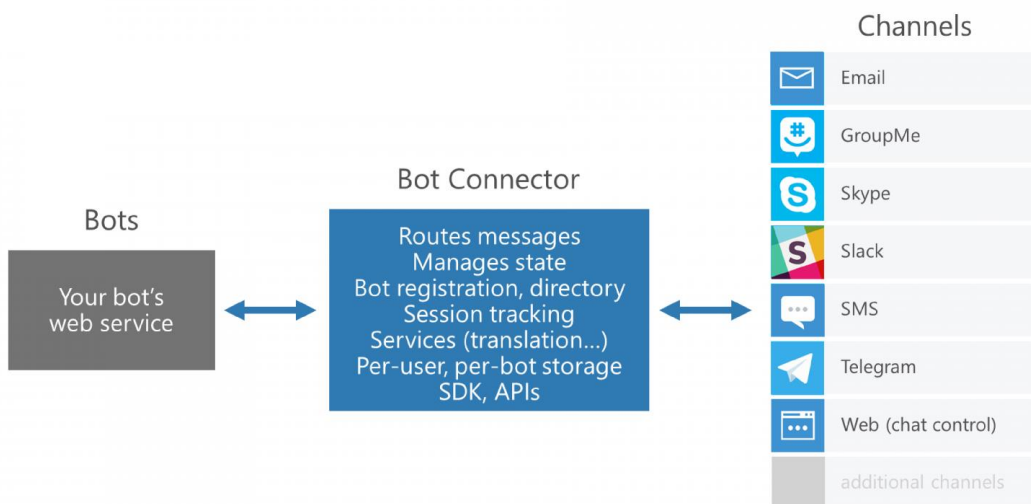


Εικόνα 55 - Bot Framework - <http://www.garypretty.co.uk/2018/02/21/whats-next-for-the-microsoft-bot-framework/>

### 6.1 Τι είναι το Bot Framework

Το Microsoft Bot Framework [28] παρέχει ακριβώς αυτό που χρειαζόμαστε για να δημιουργήσουμε και να συνδέσουμε έξυπνα bots που αλληλοεπιδρούν φυσικά για την ύπαρξη ομιλίας χρηστών, από απλά μηνύματα /sms μέχρι τις πιο διάσημες διαδικτυακές εφαρμογές ομιλίας (Skype, Messenger, Viber etc). Για τη δημιουργία ενός ολοκληρωμένου συστήματος υπάρχουν τρία κομμάτια που συμπληρώνουν ουσιαστικά το παζλ. Το bot , ο bot connector , και τα κανάλια. Τα bots επιτρέπουν στους χρήστες να αλληλοεπιδρούν με έξυπνες λύσεις σαν να συνομιλούν με κάποιο άλλο φυσικό πρόσωπο, και οι αλληλεπιδράσεις αυτές μπορούν να πάρουν πολλές μορφές. Το Framework παρέχει στους προγραμματιστές μία «πύλη» για προγραμματιστές και ένα σύστημα ανάπτυξης λογισμικού (SDK) για να δημιουργήσει bots, και έναν bot connector για να μπορέσει να συνδεθεί σε διάφορα κανάλια όπως είναι το Twitter και το Slack, όπως επίσης και ένα σημειωματάριο Bot για να μπορέσει να δει τι bot υπάρχουν και ποια απο αυτά μπορεί να δοκιμάσει.

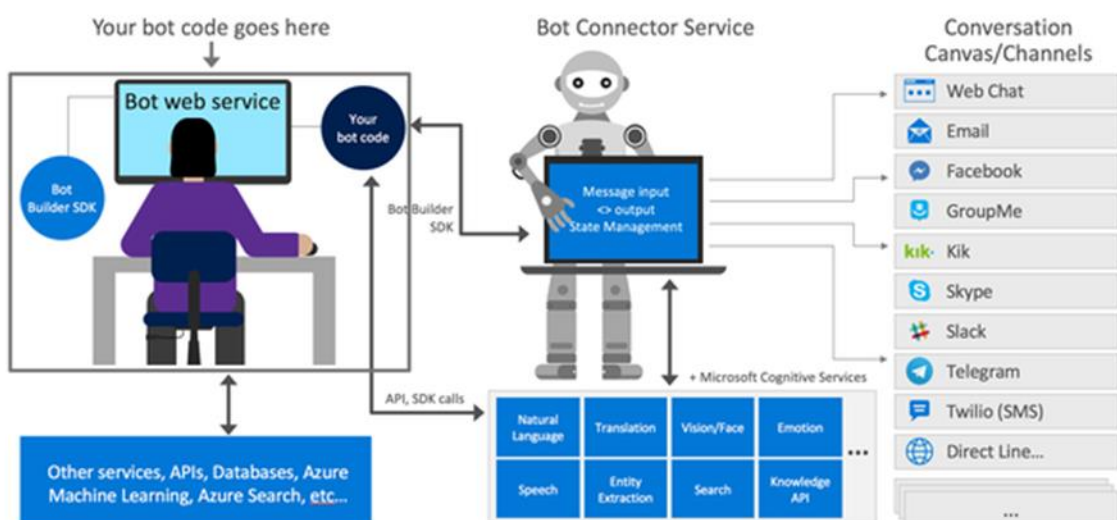




Εικόνα 56 - Bot Framework with Channels –

[https://blogs.msdn.microsoft.com/uk\\_faculty\\_connection/2016/04/05/what-is-microsoft-bot-framework-overview/](https://blogs.msdn.microsoft.com/uk_faculty_connection/2016/04/05/what-is-microsoft-bot-framework-overview/)

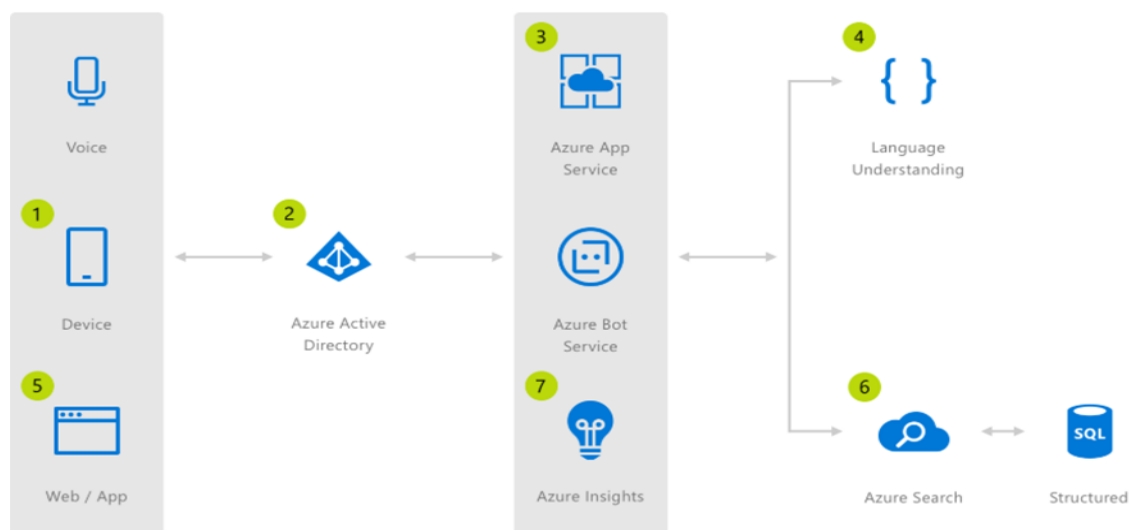
Στην παραπάνω απεικόνιση βλέπουμε τη διαδικασία που μας παρέχει ουσιαστικά το Bot Framework. Ο Bot Connector μας επιτρέπει να συνδέσουμε τα bot μας χωρίς προβλήματα στα διαθέσιμα κανάλια που μπορούμε να επιλέξουμε.



Εικόνα 57 Bot Framework\_Whole - <https://www.quora.com/What-is-Microsofts-Bot-Framework>

Ένα πλήρες παράδειγμα των δυνατοτήτων που μας παρέχει το Bot Framework είναι το παραπάνω. Κάθε σειρά συνομιλιών ονομάζεται Διάλογος(Dialog), όπως ο διάλογος που μπορεί να υπάρξει μεταξύ ενός bot με τον χρήστη. Για παράδειγμα ένας χρήστη μπορεί να γράψει σε κάποιο από τα διαθέσιμα κανάλια ‘θέλω να κλείσω μία πτήση από την Αθήνα’ και το bot να το απαντήσει με κάποια μορφή διαλόγου τύπου: ‘θα ήθελα να μάθω που θα ήθελες να επιβιβαστείς και ποια μέρα’.

Παράδειγμα Bot Παροχής πληροφοριών:



Εικόνα 58 - Info Bot - <https://dev.botframework.com/>

- 1)Ο χρήστης ξεκινά την εφαρμογή.
- 2)Ο Azure Active Directory επαληθεύει την ταυτότητα του χρήστη.
- 3)Ο χρήστης μπορεί να ρωτήσει τι είδους ερωτήματα μπορούν να απαντηθούν.
- 4)Η υπηρεσία επιστρέφει ένα FAQ το οποίο έχει δημιουργηθεί με QnA.
- 5)Ο χρήστης ρωτά έγκυρο ερώτημα.
- 6)Το bot στέλνει ερώτημα στο Azure Search, το οποίο επιστρέφει πληροφορίες σχετικά με τα δεδομένα της εφαρμογής.
- 7)Οι πληροφορίες σχετικά με την εφαρμογή συγκεντρώνει τηλεμετρία χρόνου εκτέλεσης για να βοηθήσει στην ανάπτυξη με την απόδοση και τη χρήση το bot.

## 6.2 Azure Bot Service

Το azure bot service παρέχει εργαλεία για να δημιουργήσει , να ελέγχει ,να αναπτύσσει και να διαχειρίζεται έξυπνα bot σε έναν χώρο. Μέσω του αρθρωτού και επεκτάσιμου πλαισίου που παρέχεται από το SDK, οι προγραμματιστές μπορούν να αξιοποιήσουν τα πρότυπα για να

δημιουργήσουν bots που παρέχουν ομιλία, κατανόηση γλώσσας, ερωτήσεις και απαντήσεις και πολλά άλλα.

### 6.2.1 Η ιδέα του χτίζω ένα bot

Υπάρχουν τρεις τρόποι με τους οποίους μπορούμε να δημιουργήσουμε και να 'χτίσουμε' ένα bot. Ο πρώτος τρόπος είναι η δημιουργία ενός bot μέσω του Bot Service, ο δεύτερος η δημιουργία ενός bot μέσω του bot builder sdk χρησιμοποιώντας .net και ο τρίτος είναι η δημιουργία ενός bot μέσω του bot builder sdk χρησιμοποιώντας node.js. Όμως, η βασική σύλληψη ιδέας και δημιουργίας ενός bot βασίζεται σε μια σειρά από συγκεκριμένες λειτουργίες οι οποίες αποφέρουν τη σωστή διαδικασία εφαρμογής ενός bot. Η διαδικασία [29] αυτή αποτελείται από 6 στάδια: 1) Το πλάνο , 2) Τη δημιουργία , 3) Τη δοκιμή , 4) Τη δημοσίευση, 5) Τη σύνδεση , 6) Τη αξιολόγηση.



Εικόνα 59 - Bot steps - <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-3.0>

#### 6.2.1.1 Πλάνο

Πριν ξεκινήσουμε το προγραμματιστικό κομμάτι, θα πρέπει να έχουμε ξεκαθαρίσει τις ανάγκες του bot που σκοπεύουμε να δημιουργήσουμε. Θα ήταν φρόνιμο να ανατρέξουμε στις οδηγίες σχεδιασμού bot για τις βέλτιστες πρακτικές σχεδίασης που υπάρχουν. Μπορούμε να δημιουργήσουμε ένα απλό bot ή να συμπεριλάβουμε πιο εξελιγμένες δυνατότητες, όπως είναι η ομιλία , η κατανόηση γλώσσας , οι ερωτοαπαντήσεις (QnA), ή τη δυνατότητα εξαγωγής γνώσεων από διαφορετικές πηγές και την παροχή έξυπνων απαντήσεων.

#### 6.2.1.2 Δημιουργία Bot

Το Bot το οποίο θέλουμε να δημιουργήσουμε είναι μια υπηρεσία ιστού(web service) που υλοποιεί μια διεπαφή συνομιλίας και επικοινωνεί με το Bot Service. Γίνεται να δημιουργήσουμε αυτή τη λύση σε οποιοδήποτε περιβάλλον γλώσσας (.net , node.js) και μέσω της υπηρεσίας αυτής να πάρουμε εργαλεία εύκολης εκκίνησης για το Visual Studio ή απευθείας μέσω του Azure Portal. Μπορούμε να δημιουργήσουμε ένα bot χρησιμοποιώντας SDK, Azure Portal ή CLI εργαλεία[30](έτοιμα templates μέσω των οποίων μπορούμε να δημιουργήσουμε bot είτε στην πλατφόρμα της node.js, είτε της Python, είτε της Java, είτε της .NET).

### 6.2.1.3 Δοκιμή Bot

Τα bots είναι περίπλοκες εφαρμογές , με μεγάλα και πολλά διαφορετικά μέρη που πρέπει να συνεργάζονται ταυτόχρονα. Όπως με οποιαδήποτε άλλη περίπλοκη εφαρμογή έτσι και στα bot υπάρχει πιθανότητα η εύρεση διάφορων σημαντικών bugs ή συμπεριφορά του bot με διαφορετικό τρόπο με τον οποίο θα περιμέναμε να λειτουργεί. Όταν πριν τη δημοσίευση ενός bot τίθεται αναγκαίο το testing του , με τους δύο παρακάτω εξής τρόπους. Είτε δοκιμάζοντας του μέσω της εφαρμογής που υπάρχει του bot emulator είτε μέσω είτε μέσω web chat το οποίο παρέχεται μέσω του Azure Portal.

### 6.2.1.4 Δημοσίευση

Έφωσον έχουμε περάσει τα προηγούμενα στάδια με επιτυχή τρόπο είμαστε έτοιμοι να δημοσιεύσουμε το bot μας , μέσω του Azure Portal ή μέσω του δικού μας web service.Είναι δυνατό να ρυθμίσουμε τη συνεχή ανάπτυξη που μας επιτρέπει να αναπτύσουμε το bot μας τοπικά και θα ήταν χρήσιμο να ανεβάσουμε τον κώδικα μας σε πλατφόρμες ανοιχτού κώδικα όπως είναι το GitHub ή το Visual Studio Team services για τη βοήθεια ή την βελτίωση που μπορούμε να δεχθούμε απο έναν δεύτερο προγραμματιστή.

### 6.2.1.5 Σύνδεση

Έφωσον έχουμε δημοσιεύσει το bot μας είμαστε έτοιμοι να το συνδέσουμε σε οποιοδήποτε απο τα διαθέσιμα κανάλια που μας παρέχονται(Facebook, Messenger, Kik, Skype, Slack, Microsoft Terms, Telegram , SMS, Twitter) για να αυξηθούν οι αλληλεπιδράσεις και να προσεγγίσουμε με αυτόν τον τρόπο περισσότερους χρήστες.

### 6.2.1.6 Αξιολόγηση

Απο τη στιγμή που έχουμε κάνει τη σύνδεση με το κανάλι που επιθυμούμε το bot είναι έτοιμο να παρέχει τις συγκεκριμένες λειτουργίες που μπορεί να δώσει. Απο τη στιγμή που υπάρχει αλληλεπιδράσει και οι χρήστες το χρησιμοποιούν θα ήταν φρόνιμο να χρησιμοποιήσουμε τα δεδομένα αυτά που έχουμε συλλέξει απο τους χρήστες απο το Azure Portal για να μπορέσουμε να δούμε τα θετικά στοιχεία του bot μας , αλλά και τα αρνητικά αυτού ούτως ώστε να εστιάσουμε σε επόμενο στάδιο στη βελτίωση των προβλημάτων αυτών για την καλύτερη επίδοση του bot μας. Μερικά απο τα δεδομένα που μπορούμε να λάβουμε είναι επιπέδου υπηρεσίας (service-level) και οργάνων όπως κυκλοφορία ,λανθάνουσα κατάσταση. Τα Analytics παρέχουν επίσης περιέχουν αναφορές σε επίπεδο συζητήσεων σχετικά με τα δεδομένα των χρηστών ,μηνυμάτων και καναλιών.

## 6.2.2 Azure Bot Service Documentation

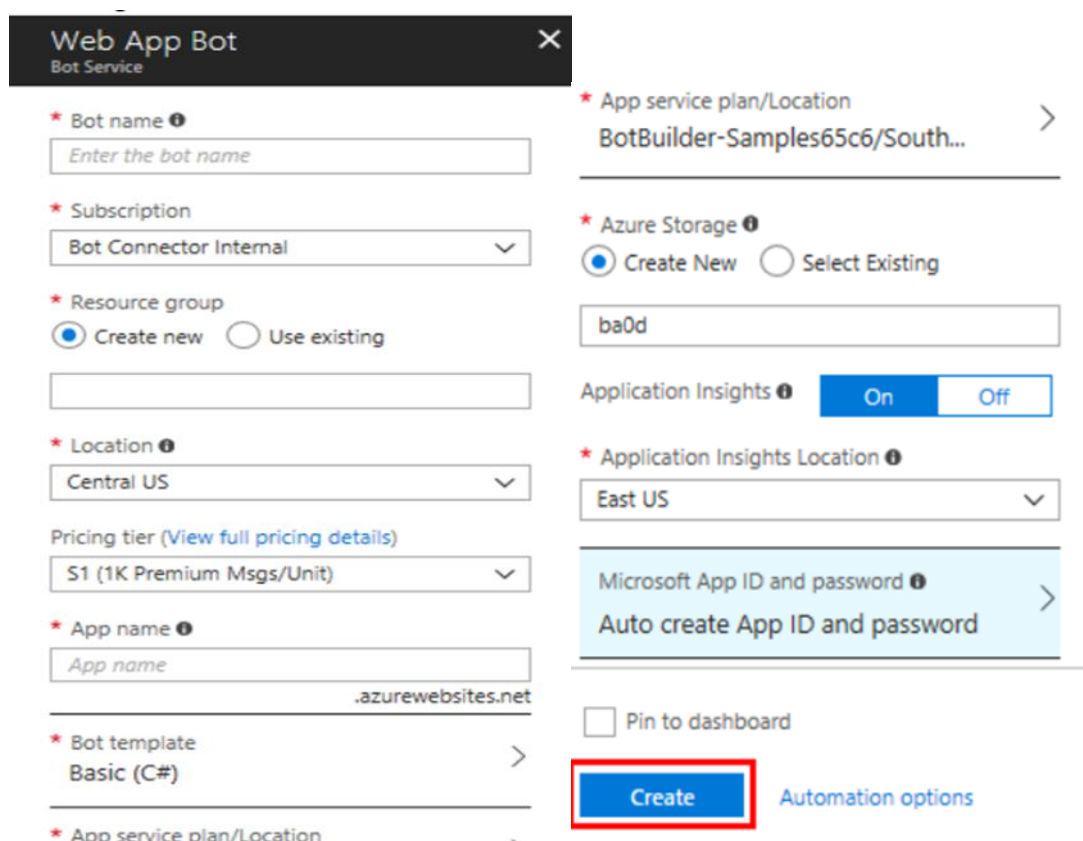
Το Bot Service παρέχει ένα ενσωματωμένο περιβάλλον που έχει σχεδιαστεί ειδικά για την ανάπτυξη των bot, δίνοντας μας τη δυνατότητα, να δημιουργήσουμε να συνδέσουμε , να δοκιμάσουμε και να διαχειριζόμαστε έξυπνα bot. Το Bot Service αξιοποιεί το Bot Builder SDK με υποστήριξη σε .NET και σε Node.js.Όπως αναφέραμε και πιο πάνω υπάρχουν τρεις τρόποι με τους οποίους μπορούμε να δημιουργήσουμε ένα bot. Ας ξεκινήσουμε αναλύοντας τον πρώτο, ο οποίος είναι η δημιουργία ενός bot μέσω bot service αναφέροντας λίγα λόγια και τα βασικά στάδια που χρειάζονται να υλοποιηθούν για την επίτευξη του τρόπου αυτού.

### 6.2.2.1 Bot Service

Το Bot service [31] παρέχει τα βασικά συστατικά για τη δημιουργία ενός bot, συμπεριλαμβανομένου του Bot Builder SDK για την ανάπτυξη των bot και του Bot Framework που χάρη σ' αυτό μπορούμε να κάνουμε τη σύνδεση των bot με τα κανάλια όπως έχουμε αναφέρει και παραπάνω. Το Bot Service παρέχει πέντε πρότυπα από τα οποία μπορούμε να επιλέξουμε κατά τη δημιουργία του bot με υποστήριξη για .NET και για Node.js. Παρακάτω φαίνεται ποια ροή θα έπρεπε να ακολουθήσουμε για τη δημιουργία ενός bot μέσω του bot service.

#### Εικόνα 60 - Bot Service-

<https://docs.microsoft.com/en-us/azure/bot-service/bot-service-quickstart?view=azure-bot-service-3.0>



Αρχικά, θα πρέπει να δημιουργήσουμε έναν λογαριασμό στο Azure και να συνδεθούμε σε αυτό. Έπειτα δημιουργούμε ένα καινούργιο bot service ακολουθώντας τις παρακάτω 3 διαδικασίες:

1) Κάνουμε 'click' στη δημιουργία νέο συνδέσμου πόρων (Create New Resource) που βρίσκεται στην πάνω αριστερή γωνία του Azure Portal, και έπειτα επιλέγουμε AI+γνωσιακές υπηρεσίες (Cognitive Services) > Web App Bot.

2) Θα ανοίξει μια νέα mini σελίδα η οποία θα περιέχει πληροφορίες με το web app bot το οποίο είμαστε πρόθυμοι να δημιουργήσουμε.

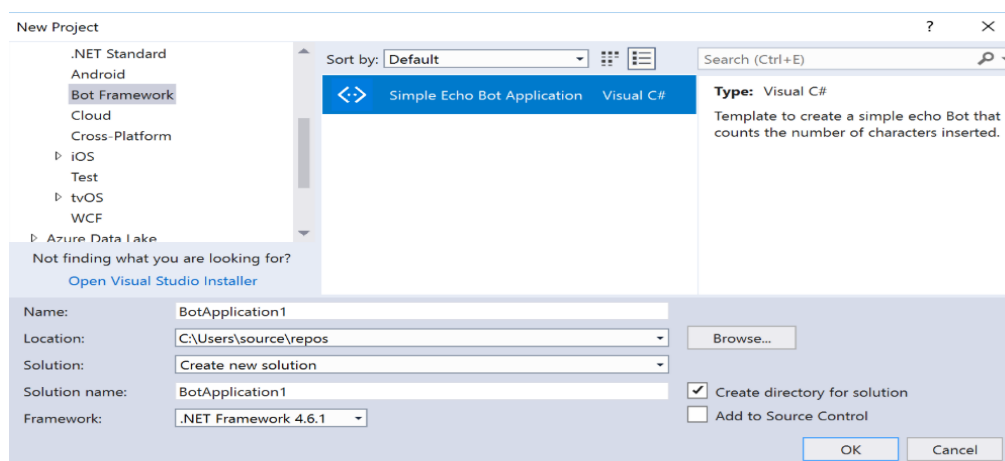
3) Στη σελίδα που θα μας ανοίξει που θα περιέχει πληροφορίες σχετικά με το Bot Service, θα πρέπει να επιλέξουμε τις δικές μας ανάγκες που χρειαζόμαστε για το bot μας, όπως φαίνεται στη παραπάνω εικόνα(1).

### 6.2.2.2 Bot Builder SDK

Ο Bot Builder SDK [32] μας δίνει τη δυνατότητα να δημιουργήσουμε bot που υποστηρίζουν διαφορετικούς τύπους αλληλεπιδράσεων με τους χρήστες. Είναι δυνατό να σχεδιάσουμε συνομιλίες στο bot για να είμαστε ελεύθεροι, όπως επίσης μπορεί να έχει πιο καθοδηγούμενες αλληλοεπιδράσεις όπου παρέχει τις επιλογές ή τις ενέργειες του χρήστη. Η συνομιλία μπορεί να είναι είτε απλό κείμενο είτε πιο περίπλοκες κάρτες που περιέχουν κείμενο, εικόνες και κουμπιά δράσης(action buttons). Μπορούμε να προσθέσουμε αλληλεπιδράσεις φυσικής γλώσσας και ερωτήσεις-απαντήσεις, οι οποίες επιτρέπουν στους χρήστες να αλληλοεπιδρούν με το bot με φυσικό τρόπο. Μπορούμε να δημιουργήσουμε bot χρησιμοποιώντας το Bot Builder SDK με υποστήριξη .NET και με υποστήριξη Node.js.

#### 6.2.2.2.1 Για .NET

[33]Είναι ένα χρήσιμο και εύχρηστο πλαίσιο για την ανάπτυξη bot χρησιμοποιώντας Visual Studio και Windows. Το SDK στο συγκεκριμένο τρόπο χρησιμοποιεί τη γλώσσα της C# για να παρέχει ένα οικείο περιβάλλον στους προγραμματιστές .NET για τη δημιουργία ισχυρών bot. Προϋποθέσεις για τη δημιουργία bot μέσω .NET: 1) Visual Studio 2017, 2) Μέσα στο Visual Studio, ενημέρωση όλων των επεκτάσεων στη τελευταία τους έκδοση, 3) Bot πρότυπο για C#. Παρακάτω φαίνεται ένα στιγμιότυπο από το visual studio στο οποίο φαίνονται οι απαραίτητες διαδικασίες που χρειάζονται για τη δημιουργία ενός bot σε .net περιβάλλον.



**Εικόνα 61 - .NET - <https://docs.microsoft.com/en-us/azure/bot-service/dotnet/bot-builder-dotnet-quickstart?view=azure-bot-service-3.0>**

#### 6.2.2.2.2 Για Node.js

[34]Είναι ένα ακόμη πλαίσιο για τη δημιουργία και ανάπτυξη των bot. Είναι εύκολο να χρησιμοποιήσουμε και να μοντελοποιήσουμε πλαίσια όπως το Express & Restify για να παρέχει

έναν οικείο περιβάλλον για τους προγραμματιστές JavaScript. Προϋποθέσεις για τη δημιουργία bot σε περιβάλλον Node.js: 1) Εισαγωγή της Node.js, 2) Δημιουργία ενός φακέλου για το bot μας, 3) Από μία γραμμή εντολών ή ένα τερματικό, μεταβείτε στο φάκελο που μόλις δημιουργήσατε, 4) Τρέξτε την παρακάτω npm εντολή – npm init. Με την παραπάνω εντολή καταχωρούμε πληροφορίες σχετικά με το bot μας και το npm θα δημιουργήσει ένα αρχείο package.json που περιέχει της πληροφορίες που δώσαμε. Η εισαγωγή της Node.js γίνεται αντίστοιχα τρέχοντας την εντολή -npm install σε ένα τερματικό.

## 6.3 Σενάρια Bot

[35] Σε αυτήν την ενότητα θα μελετήσουμε τα βασικά σενάρια bot που υπάρχουν που είναι ισχυρά και επιτυχημένα και έχουν κατασκευαστεί με τη βοήθεια του Bot Service. Τα πέντε βασικά σενάρια bot που υπάρχουν είναι : Το εμπορικό σενάριο bot (Commerce Bot Scenario), το Bot πληροφοριών (Information Bot Scenario), το Cortana Skill Bot Scenario, το επιχειρηματικό σενάριο για την παραγωγικότητα επιχειρήσεων (Enterprise Productivity bot scenario), και το IoT (Internet of things) bot Scenario.

### 6.3.1 Commerce Bot Scenario

Στο συγκεκριμένο σενάριο περιγράφεται ένα bot το οποίο είναι έτοιμο να αντικαταστήσει τις παραδοσιακές αλληλεπιδράσεις ηλεκτρονικού ταχυδρομείου και τηλεφωνικών κλήσεων που υπήρχε μέχρι τώρα μεταξύ πελάτη και ιδιοκτήτη. Το Bot εκμεταλλεύεται τις γνωσιακές του υπηρεσίες που έχει εκπαιδευτεί κατά τη δημιουργία του για την καλύτερη επεξεργασία των αιτημάτων των πελατών μέσω κειμένου και φωνής με το περιεχόμενο που συγκεντρώνεται από την ενσωμάτωση με τις υπηρεσίες που τρέχουν από πίσω (backend services).

### 6.3.2 Cortana Skill Bot Scenario

Το σενάριο Cortana Skill bot εκμεταλλεύεται την Cortana (εικονικός βοηθός που υπάρχει σε συσκευές που έχουν δημιουργηθεί από την Microsoft). Χρησιμοποιώντας το φυσικό περιβάλλον της φωνής σας και ένα προσαρμοσμένο Cortana Skill Bot, μπορείτε να ζητήσετε από την Cortana να μιλήσει σε έναν οργανισμό, όπως π.χ. σε μια εταιρία κινητής τηλεφωνίας, για να σας βοηθήσει να κλείσετε ένα ραντεβού με βάση το που βρίσκεστε όταν πραγματοποιείτε την κλήση.

### 6.3.3 Enterprise Productivity Bot Scenario

Στο συγκεκριμένο σενάριο μπορούμε να δούμε πως να ενσωματώσουμε ένα bot στο ημερολόγιο του Office 365 και άλλες υπηρεσίες για να αυξήσουμε την παραγωγικότητα μας. Το Bot ενσωματώνεται στο Office 365 για να υπάρξει πιο γρήγορη και πιο εύκολη η δημιουργία μιας συνάντησης με ένα άλλο άτομο. Κατά τη διαδικασία αυτή μπορούμε να έχουμε επίσης πρόσβαση σε πρόσθετες υπηρεσίες όπως η Dynamics CRM.

### 6.3.4 Information Bot Scenario

Το Bot πληροφοριών μπορεί να απαντήσει σε ερωτήσεις που ορίζονται σε ένα μεγάλο σύνολο των συνήθη ερωτήσεων (FAQs) χρησιμοποιώντας την τεχνική των ερωτοαπαντήσεων (QnA) και να απαντήσει σε πιο ανοιχτού τύπου ερωτήσεις του Azure Search. Συχνά οι πληροφορίες είναι

«θαμμένες» σε δομημένα data stores όπως είναι για παράδειγμα ο Sql Server που μπορούν να εμφανιστούν εύκολα μέσω αναζήτησης.

### **6.3.5 Internet of Things Bot Scenario**

Τέλος το συγκεκριμένο σενάριο μια διευκολύνει να ελέγξουμε για παράδειγμα «έξυπνες» συσκευές που έχουμε στο σπίτι μας όπως θα μπορούσε να είναι η ρύθμιση της φωτεινότητας των λαμπών ανάλογα με την ένταση της φωνής μας.



## ΚΕΦΑΛΑΙΟ 7- BOTVOYAGE

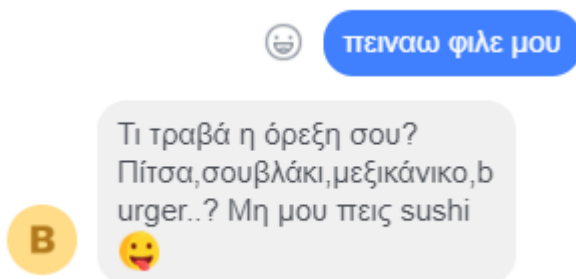
### 7.1 Απεικόνιση και ανάλυση των λειτουργιών

Η εφαρμογή BotVoyage, όπως έχει αναφερθεί, αξιοποιεί ένα εύρος λειτουργιών με κάποιες κύριες κατηγορίες. Υπάρχει η κατηγορία φαγητού η οποία περιλαμβάνει τις υποκατηγορίες sushi, pizza, burger, κρέπα, μεξικάνικο, σουβλάκι. Επιπρόσθετα, υπάρχει η κατηγορία του ποτού με υποκατηγορίες τις εξής : κρασί, καφές, cocktail, μπίρα. Άλλη μία κύρια κατηγορία είναι οι Υπηρεσίες και ειδικότερα πτήσεις, ξενοδοχεία, αξιοθέατα, νοσοκομεία, ψώνια, μουσεία, θέατρα. Παρακάτω θα αναλύσουμε με στιγμιότυπα τη ροή των δύο πρώτων κύριων λειτουργιών και από την κατηγορία Υπηρεσίες τις πτήσεις, τα ξενοδοχεία και τα αξιοθέατα. Τέλος θα αναλυθούν επίσης δευτερεύουσες λειτουργίες όπως είναι ο χαιρετισμός, η βοήθεια, η επανεκκίνηση και η αλλαγή της γλώσσας.

Όσον αφορά τη γενικότερη ροή της εφαρμογής, στις λειτουργίες που περιλαμβάνουν αποστολή τοποθεσίας γίνονται οι εξής διαδικασίες. Ο χρήστης αποστέλλει ένα μήνυμα σε φυσική γλώσσα, το οποίο αποστέλλεται στην πλατφόρμα Wit.ai και μέσω NLU γίνεται κατανοητό ποια οντότητα αφορά το φυσικό μήνυμα, για ποιο θέμα δηλαδή μιλά ο χρήστης. Επόμενο βήμα είναι η αποστολή τοποθεσίας, μέσω της οποίας ο χρήστης μπορεί να στείλει την τρέχουσα είτε οποιαδήποτε άλλη πληκτρολογήσει στην αναζήτηση εάν χρησιμοποιεί messenger. Στην περίπτωση που χρησιμοποιεί την ιστοσελίδα του Facebook, μπορεί μόνο να στείλει τη τρέχουσα ή να μετακινήσει την πινέζα που δείχνει την τοποθεσία, μην έχοντας τη δυνατότητα να αναζητήσει για κάποιον πιο απόμακρο προορισμό. Η τοποθεσία αποστέλλεται με τη μορφή συντεταγμένων, οι οποίες μετατρέπονται σε μία ονομασία περιοχής, χρησιμοποιώντας το reverse geocoding της Google. Η τοποθεσία με την ονομασία της και η οντότητα που έχει επιλεγεί από το Wit.ai συνδυάζονται σε ένα request προς το Foursquare, λαμβάνεται από την εφαρμογή και παρουσιάζεται στον χρήστη οπτικά με μορφή carousel των πέντε πιο προτεινόμενων μερών της πλατφόρμας του Foursquare.

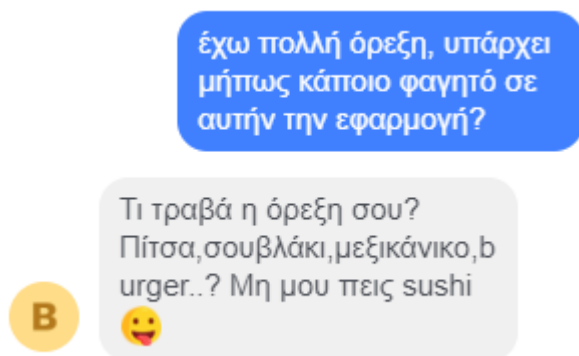
#### 7.1.1 Φαγητό

Για να δει ο χρήστης τι μπορεί να φάει αρκεί να γράψει κάποια σχετική πρόταση με το φαγητό σε γενικό πλαίσιο, είτε ότι πεινάω, θέλει να φάει, είτε τι φαγητό υπάρχει κ.λπ. Αυτό συμβαίνει διότι η κατηγορία αυτή είναι με οντότητα βασισμένη σε λέξεις κλειδιά.



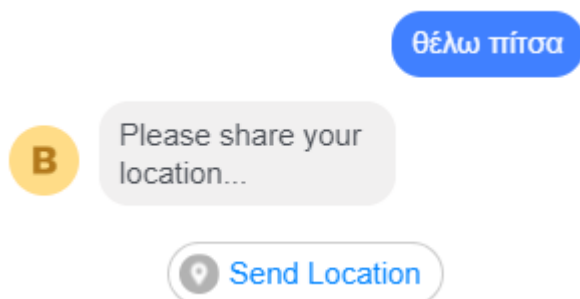
Εικόνα 62 - Φαγητό\_1 - Messenger

Βλέπουμε και ότι σε πιο σύνθετη πρόταση γίνεται κατανοητό μέσω NLU, ότι ο χρήστης ομιλεί για το θέμα του φαγητού.



**Εικόνα 63 - Φαγητό\_2 - Messenger**

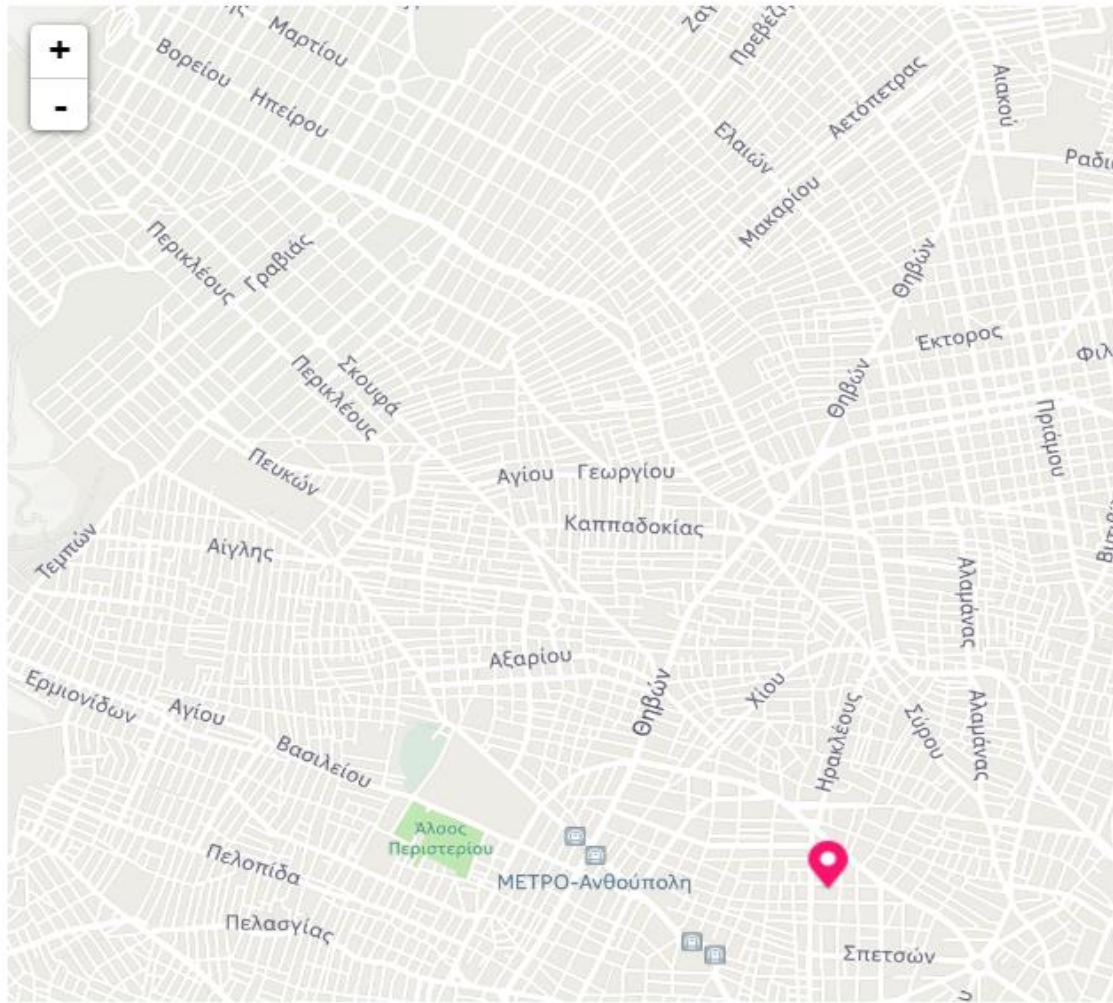
Επόμενο βήμα με τη λογική αποτελεί η αποστολή ενός μηνύματος σε σχέση με αυτά τα φαγητά που μου έστειλε το bot.



**Εικόνα 64 - Φαγητό\_3 - Messenger**

Όπως φαίνεται παραπάνω αφού γράψω ότι θέλω πίτσα, με προτρέπει να αποστείλω την τοποθεσία μου. Οπότε το επόμενο βήμα στη ροή είναι αυτό και παρατίθεται στην κάτω εικόνα.

## Η τοποθεσία σας



Αποστολή τοποθεσίας

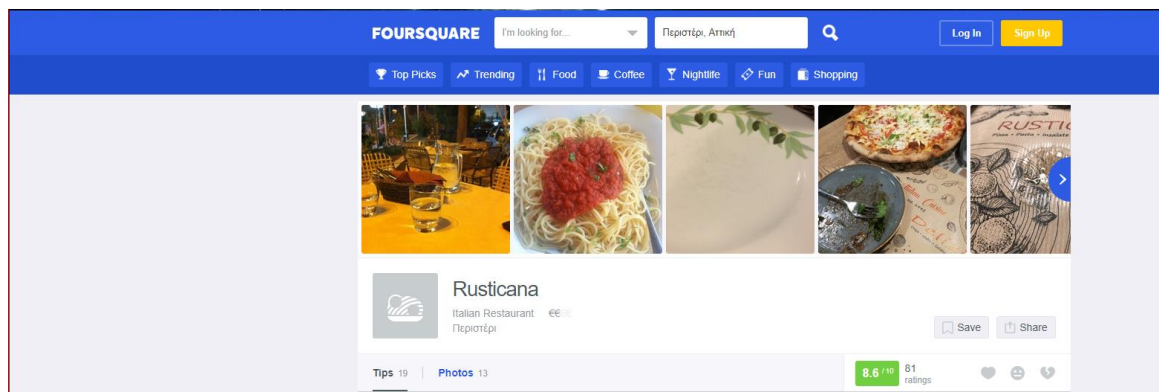
**Εικόνα 65 – Αποστολή τοποθεσίας - Messenger**

Μετά την επιλογή τοποθεσίας, πατώντας αποστολή, λαμβάνει ο χρήστης τα πέντε πιο διάσημα μέρη που έχουν την ετικέτα πιτσαρία.



**Εικόνα 66 - Προβολή Φαγητών - Messenger**

Επόμενο βήμα ίσως αποτελεί μία εκτενέστερη ενημέρωση για ένα από τα εικονιζόμενα μέρη πατώντας το κουμπί Μια καλύτερη ματιά. Μπορεί φυσικά ο χρήστης εάν θελήσει να μην προχωρήσει σε αυτή την ενέργεια, αλλά να συνεχίσει τη συνομιλία διαφορετικά.



**Εικόνα 67 - Ανακατεύθυνση σε Foursquare - Messenger**


Έγινε ανακατεύθυνση στην πρώτη επιλογή Rusticana στο site της Foursquare. Για άλλη μια φορά φαίνεται ο πηγαίος χαρακτήρας της εφαρμογής μας που είναι αυτός του μεσάζοντα.

Ένα σημαντικό χαρακτηριστικό που πρέπει να σημειωθεί είναι ότι υπάρχει η δυνατότητα ο χρήστης να επιλέξει αμέσως τι φαγητό θέλει χωρίς να περάσει από την κατηγορία του φαγητού. Μπορεί δηλαδή να γράψει αμέσως ότι θέλει να έχει ένα burger, χωρίς να στείλει ότι πεινάει ή οτιδήποτε συναφές.

## Εικόνα 68 - Κατανόηση οντότητας burger - Messenger

πόσο θα ήθελα να έχω ένα burger

Please share your location...



|   |   |  |
|---|---|--|
| <b>The Wooden Box</b><br>Κάλλχου 54-Ιλιον - +302102617777 | <b>Goody's Burger House</b><br>Ιδσημενέως 118-Ιλιον - +302102629150 | <b>Beaten Faten</b><br>undefined-Ιλιον - undefir |
| Μια καλύτερη ματιά?                                       | Μια καλύτερη ματιά?   | Μι   |

## Εικόνα 69 - Προβολή burger - Messenger

Έτσι παρατηρούμε ότι ακριβώς με τον ίδιο τρόπο με ένα λιγότερο βήμα( την αποστολή μηνύματος σε σχέση με το φαγητό), έχουμε το ίδιο αποτέλεσμα. Όμοια λειτουργούν κι όλες οι υπόλοιπες κατηγορίες φαγητού.

### 7.1.2 Ποτό , Ξενοδοχεία, Αξιοθέατα κ.λπ.

Ομοίως με το φαγητό, το ίδιο ισχύει και για το ποτό. Δηλαδή ο χρήστης μπορεί να γράψει ότι διψάει, η θέλει να πει, η αμέσως ότι θέλει για παράδειγμα κρασί.

## Εικόνα 70 - Κατανόηση ποτού - Messenger

θέλω να πιω. θέλω να πιω...

Τι θέλεις να πεις?Κοκτέιλ.Μπιρίτσα.Κρασάκι..

Στο παραπάνω στιγμιότυπο, πρέπει απλώς να στείλεις τι θέλεις να πεις.

ένα κρασάκι και λίγη θάλασσα

Please share your location...

Send Location

### Εικόνα 71 - Κατανόηση κρασιού - Messenger

Στο συγκεκριμένο στιγμιότυπο στέλνεις απλώς τη τοποθεσία σου κι όλα λαμβάνονται άμεσα.

Παρακάτω δοκιμάζουμε τις λειτουργίες των ξενοδοχείων και των αξιοθέατων, όπου όπως είπαμε λειτουργούν ακριβώς με τον ίδιο τρόπο.

### Εικόνα 72 - Κατανόηση ξενοδοχείου - Messenger

που θα μπορούσα να μείνω?

Please share your location...



**Fresh Hotel**  
Σοφοκλέους 26-Αθήνα - +302105248511

Μια καλύτερη ματιά?



**Grecotel Pallas Athena**  
Αθηνάς 65-Αθήνα - +302103250900

Μια καλύτερη ματιά?

### Εικόνα 73 - Προβολή ξενοδοχείων - Messenger

### Εικόνα 74 - Κατανόηση αξιοθέατων - Messenger

αξιοθέατα

Please share your location...





**Εικόνα 75 - Προβολή αξιοθέατων - Messenger**

### 7.1.3 Πτήσεις

Η συγκεκριμένη λειτουργία διαφέρει από τις άλλες, καθώς δεν υλοποιήθηκε με τη λογική της τοποθεσίας, διότι συναντήθηκαν πολλές δυσκολίες και αποφασίσαμε απλώς να είναι τα ονόματα των πόλεων που διαθέτουν αεροδρόμια σε αγγλικά γράμματα με το πρώτο μόνο γράμμα να είναι κεφαλαίο π.χ. Athens, London κ.λπ.

Ας επεξηγήσουμε ειδικότερα την ολική ροή των πτήσεων. Αρχικά ο χρήστης στέλνοντας όπως και στις παραπάνω λειτουργίες ένα μήνυμα σε φυσική γλώσσα (π.χ. θέλω να πετάξω, πτήσεις ή οτιδήποτε σχετικό) θα κληθεί να αποστείλει αρχικά το αεροδρόμιο από το οποίο θα ξεκινήσει με βάση τον προαναφερθέντα περιορισμό. Εάν δεν υπάρχει στη βάση των πόλεων που έχουν αεροδρόμια, θα αποσταλθεί στον χρήστη σχετικό μήνυμα να προσπαθήσει ξανά. Όταν το συμπληρώσει επιτυχώς, επόμενο βήμα είναι η αποστολή του αεροδρομίου προορισμού, με τον ίδιο τρόπο. Αφού εκπληρωθεί κι αυτό το βήμα, καλείται να συμπληρώσει την ημερομηνία με συγκεκριμένο μοτίβο( π.χ. 2018-04-15, Έτος-Μήνας-Μέρα), η οποία ελέγχεται αλγοριθμικά εάν είναι μετά η ίση της τρέχουσας μέρας. Εάν υπάρξει σφάλμα, ο χρήστης προχωρά ξανά σε αποστολή μέχρι να γίνει σωστά. Τελευταίο βήμα είναι η αποστολή της ημερομηνίας που θα γυρίσει, η οποία ελέγχεται στον κώδικα εάν είναι μετά της ημερομηνίας που θα ξεκινήσει το ταξίδι του. Πληρώνοντας το συγκεκριμένο κριτήριο εμφανίζεται στο χρήστη ένα κουμπί ενέργειας, το οποίο τον ανακατευθύνει στην ιστοσελίδα του Kayak για να δει περισσότερες λεπτομέρειες ή ακόμα αν θελήσει να προχωρήσει σε κάποια κράτηση. Ειδικότερα, η ανακατεύθυνση στον ιστότοπο γίνεται διότι σε ένα request προς το Kayak έχει συσσωρευτεί όλη η απαραίτητη πληροφορία με το εξής συγκεκριμένο μοτίβο.(From= αεροδρόμιο από, To= αεροδρόμιο προς, From\_Year, Month, Day διάσπαση της ημερομηνίας αναχώρησης, τα αντίστοιχα To, αποχώρησης).

[https://www.gr.kayak.com/flights/From-To/From Year-From Month-From Day/To Year-To Month-To Day?sort=bestflight\\_a](https://www.gr.kayak.com/flights/From-To/From Year-From Month-From Day/To Year-To Month-To Day?sort=bestflight_a)

Παρακάτω, συναντώνται τα διάφορα στιγμιότυπα χρήσης για να απεικονιστεί γραφικά η ροή της συγκεκριμένης λειτουργίας.

πεταζω θελω

B Επιλέξτε αεροδρόμιο αναχώρησης

Athens

B Επιλέξτε αεροδρόμιο άφιξης

London

B Επιλέξτε ημερομηνία αναχώρησης M-D(π.χ 2018-04-15)

2018-05-24

B Επιλέξτε ημερομηνία αποχώρησης Y-M-D(π.χ 2018-04-20)

2018-05-27

B Υπαρκτές πτήσεις

Δες τις πτήσεις

### Εικόνα 76 - Ροή πτήσεων - Messenger

Φαίνεται όλη η ροή με τη συγκεκριμένη εικόνα, χωρίς να έχει γίνει κάποιο λάθος και πλέον μπορεί ο χρήστης να ανακατευθυνθεί στο κεντρικό site. Το url που περιλαμβάνει το κουμπί Δες τις πτήσεις που εμφανίζει και το παρακάτω αποτέλεσμα είναι το εξής: [https://www.gr.kayak.com/flights/ATH-LTN/2018-05-24/2018-05-27?sort=bestflight\\_a](https://www.gr.kayak.com/flights/ATH-LTN/2018-05-24/2018-05-27?sort=bestflight_a)

### Εικόνα 77 - Αποτελέσματα Kayak - Messenger

KAYAK Πτήσεις Ξενοδοχεία Αυτοκίνητα Περισσότερα ▾

Με επιστροφή ▾ 1 ενήλικας ▾ Οικονομική ▾

✈ Αθήνα (ATH) ↔ ✈ Λονδίνο (LTN) 📅 Περμ 24.5 < > | 📅 Κυρ 27.5 < > ✎

9 από 9 πτήσεις Ταξινόμηση ανά Προτεινόμενα ▾

ΣΥΜΒΟΥΛΗ: Φόρτωση...  
 Παρακολουθήστε τις τιμές  ΟΧΙ

Στάσεις  
 απευθείας  
 +2 στάσεις 565€

Ωρες  
 Απογείωση από Αθήνα (ATH)  
 Περμ 14:30 19:00

| Οδηγός Τιμών | Αποσκευή  | 0     | +            | Μέθοδος πληρωμής |
|--------------|-----------|-------|--------------|------------------|
| TAROM        | 18:45 ATH | — — — | 15:50 +1 LTN | 23ώ 05λ          |
| TAROM        | 16:35 LTN | — — — | 18:05 +1 ATH | 23ώ 30λ          |
| TAROM        | 18:45 ATH | — — — | 15:50 +1 LTN | 23ώ 05λ          |
| TAROM        | 16:35 LTN | — — — | 18:05 +1 ATH | 23ώ 30λ          |

565€ TAROM **Επιλέξτε**

569€ TAROM **Επιλέξτε**



Όπως γίνεται κατανοητό, οι ημερομηνίες που έγραψε ο χρήστης είναι ίδιες με αυτές της εικόνας, όπως και οι προορισμοί.

Ας παρατηρήσουμε στην παρακάτω εικόνα και την περίπτωση λάθους κατά την εισαγωγή και των δύο αεροδρομίων αλλά και των ημερομηνιών.

### Εικόνα 78 - Λάθη στις πτήσεις - Messenger

θέλω να δω τις πτήσεις

B Επιλέξτε αεροδρόμιο αναχώρησης

athena

B Δεν βρέθηκε!Επιλέξτε ξανά σωστό αεροδρόμιο αναχώρησης

Athens

B Επιλέξτε αεροδρόμιο άφιξης

Parisara

B Δεν βρέθηκε!Επιλέξτε ξανά σωστό αεροδρόμιο άφιξης

Paris

B Επιλέξτε ημερομηνία αναχώρησης M-D(π.χ 2018-04-15)

2018-04-15

B Επιλέξτε ημερομηνία αποχώρησης Y-M-D(π.χ 2018-04-20)

2018-04-18

B Λανθασμένες ημερομηνίες!Επιλέξτε ξανά ημερομηνία αναχώρησης M-D(π.χ 2018-04-15)

2018-05-25

B Επιλέξτε ημερομηνία αποχώρησης Y-M-D(π.χ 2018-04-20)

2018-05-29

B Υπαρκτές πτήσεις

Δες τις πτήσεις

Συνειδητοποίησε ο κώδικας ότι η ημερομηνία αναχώρησης είναι πριν την τωρινή μέρα κι σου αποστέλλει το σφάλμα σου, με αποτέλεσμα να ξανακλήθεις να το στείλεις σωστά.

### 7.1.4 Αλλαγή γλώσσας

### Εικόνα 79 - Αλλαγή γλώσσας - Messenger

θελω να αλλαξω τη γλωσσα

B Hey there!How can i help you?

ελληνικά

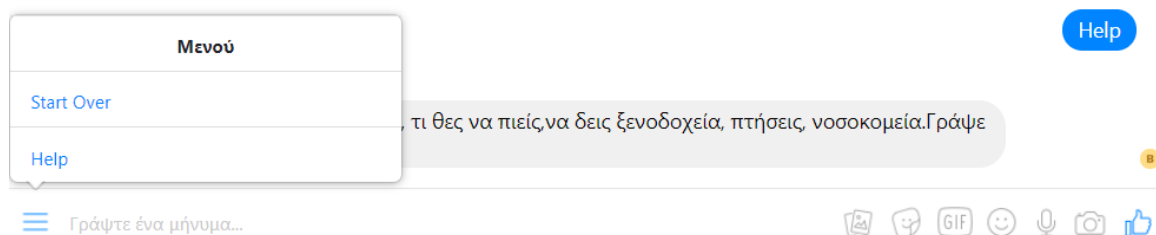
B Χαίρομαι που είστε εδώ!Τι θέλετε να βρω για σας?

Η αλλαγή της γλώσσας είναι μία οντότητα στο Wit.ai, που περιέχει δύο τιμές, ελληνικά και αγγλικά και είναι τύπου λέξεων κλειδιών. Επιπρόσθετα στον κώδικα διατηρείται σε μία μεταβλητή η τρέχουσα γλώσσα, όποτε όταν ο χρήστης γράψει σε φυσική γλώσσα κάτι σχετικό με το θέμα της αλλαγής, αλλάζει η υπάρχουσα γλώσσα στην άλλη.

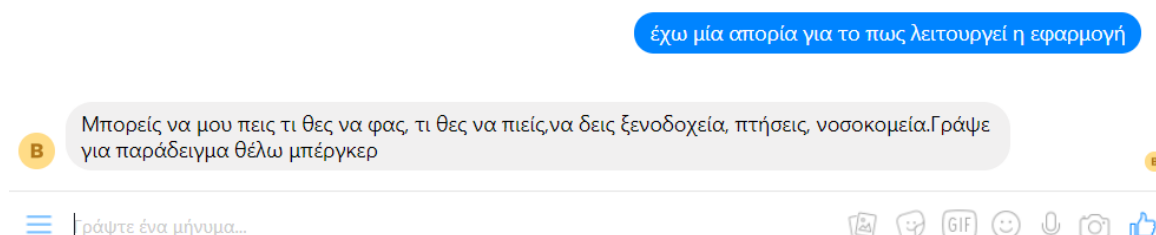
### 7.1.5 Help

Η λειτουργία του Help μπορεί να αξιοποιηθεί με δύο τρόπους. Είτε με το κουμπί στο μενού Help, είτε γράφοντας σε φυσική γλώσσα, η οποία θα κατανοηθεί πάλι μέσω NLU, διότι έχουμε μία οντότητα που λειτουργεί με βάση τα keywords που αναγνωρίζει οποιοδήποτε περιεχόμενο σε σχέση με απορίες, προβληματισμούς και βοήθεια.

#### Εικόνα 80 - Help\_1 - Messenger



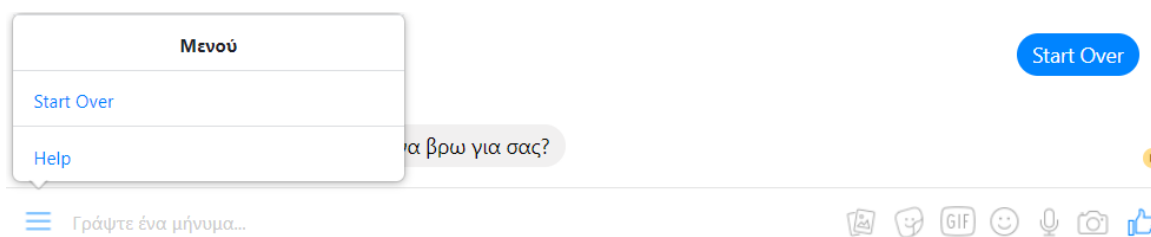
#### Εικόνα 81 - Help\_2 - Messenger



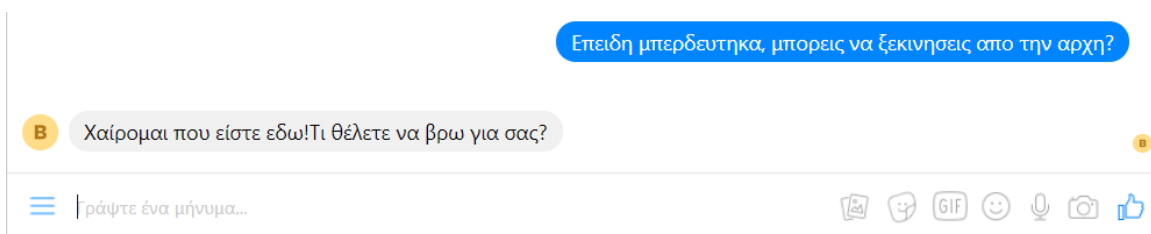
### 7.1.6 Start Over

Η συγκεκριμένη λειτουργία υπάρχει απλώς για να ξεμπερδέψει τον χρήστη από την τρέχουσα κατάσταση και να μετατεθεί η συνομιλία σε αρχικό επίπεδο και συγκεκριμένα στη φάση του χαιρετισμού. Η συγκεκριμένη λειτουργία μπορεί να αξιοποιηθεί ακριβώς με τον ίδιο τρόπο με το Help, διότι ακολουθεί την ίδια λογική σε επίπεδο NLU.

**Εικόνα 82 - StartOver\_1 - Messenger**

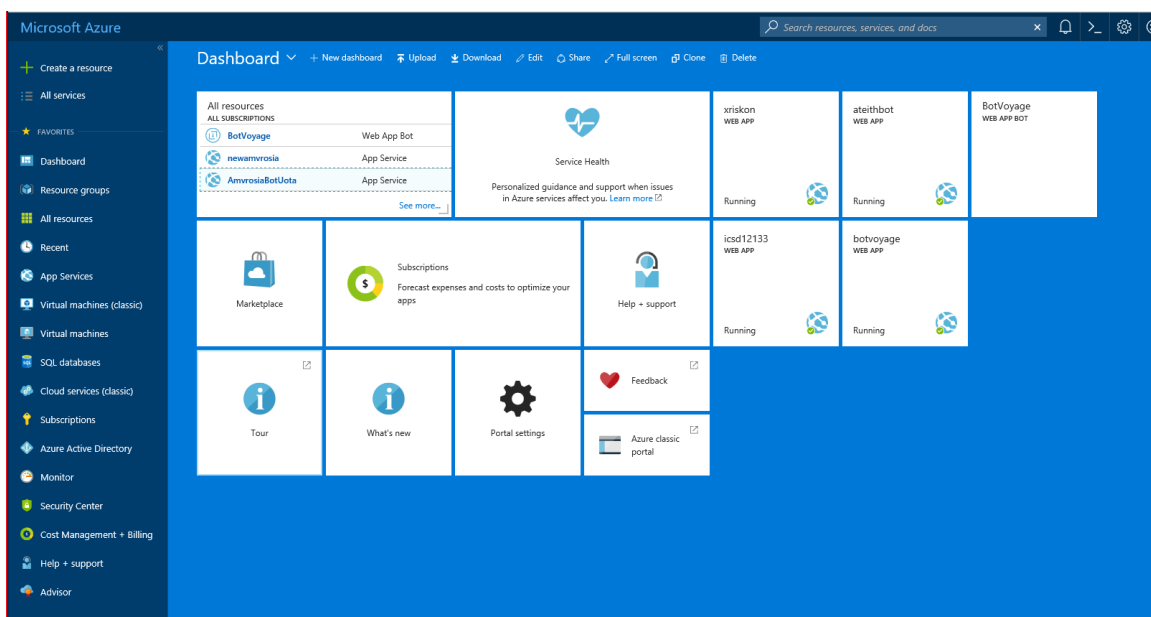


**Εικόνα 83 - StartOver\_2 - Messenger**



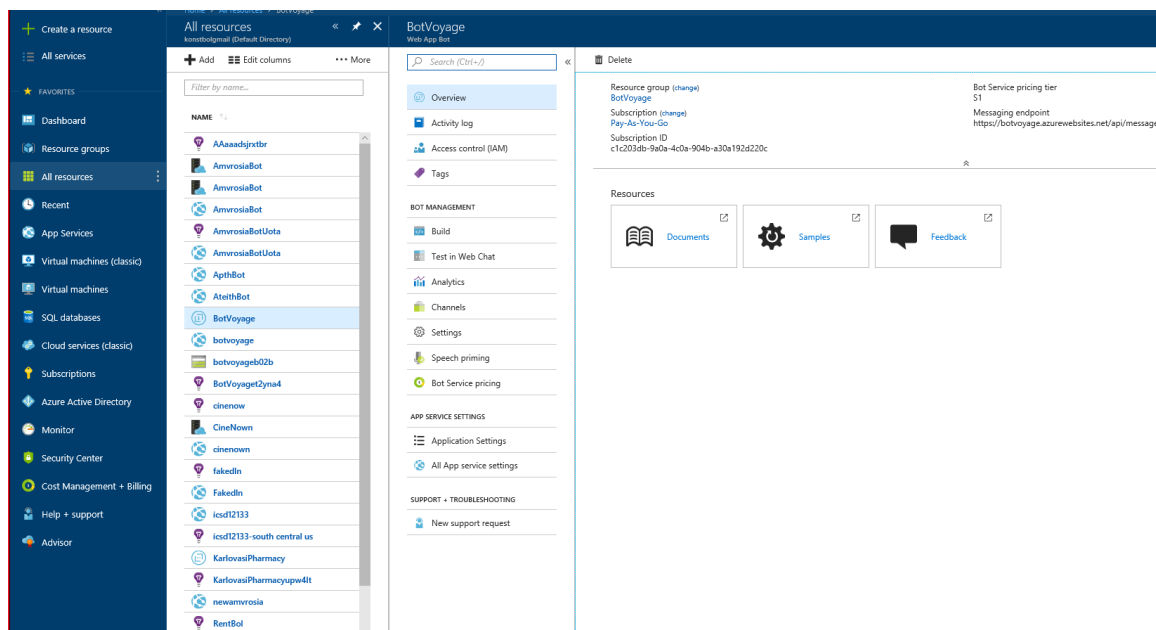
## 7.2 Απεικόνιση και ανάλυση διαχειριστικών περιβαλλόντων

### 7.2.1 Azure Portal



**Εικόνα 84 - Azure Dashboard – Azure Portal( δικό μας)**

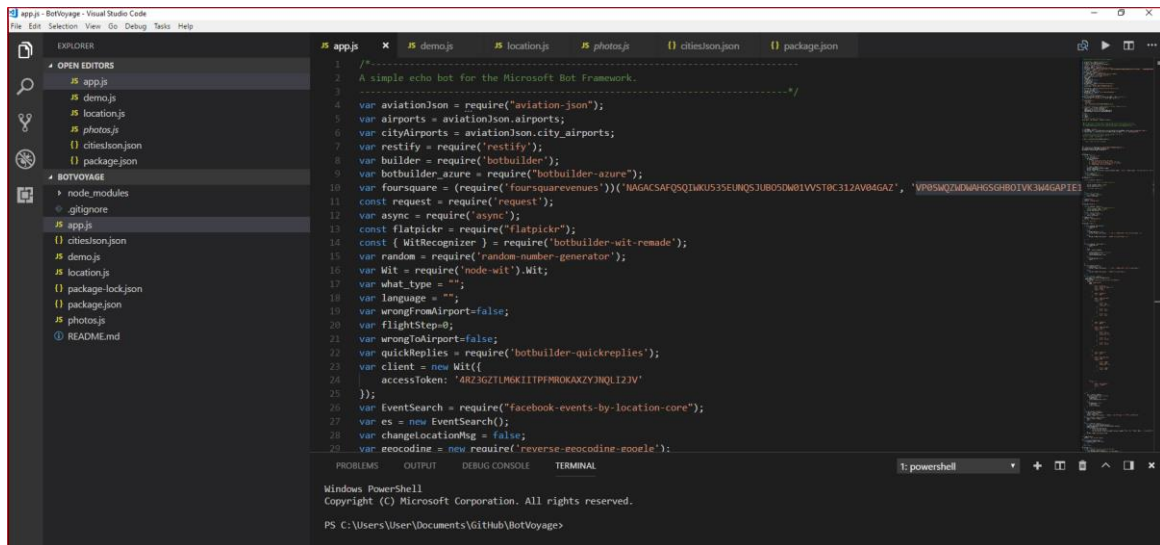
Παρατηρείται στην κάτω δεξιά γωνία το resource botvoyage. Ας το ανοίξουμε για να δούμε το περιβάλλον του συγκεκριμένου bot web app. Ειδικότερα βλέπουμε το πλήθος των επιλογών που έχουμε ως διαχειριστές, όπως τον online editor και η διασύνδεση με διάφορα κανάλια, επιλογές τις οποίες τις έχουμε αναλύσει εξονυχιστικά στο κεφάλαιο 6.



**Εικόνα 85 - App Dashboard – Azure Portal( δικό μας)**

## 7.2.2 Visual Studio Code

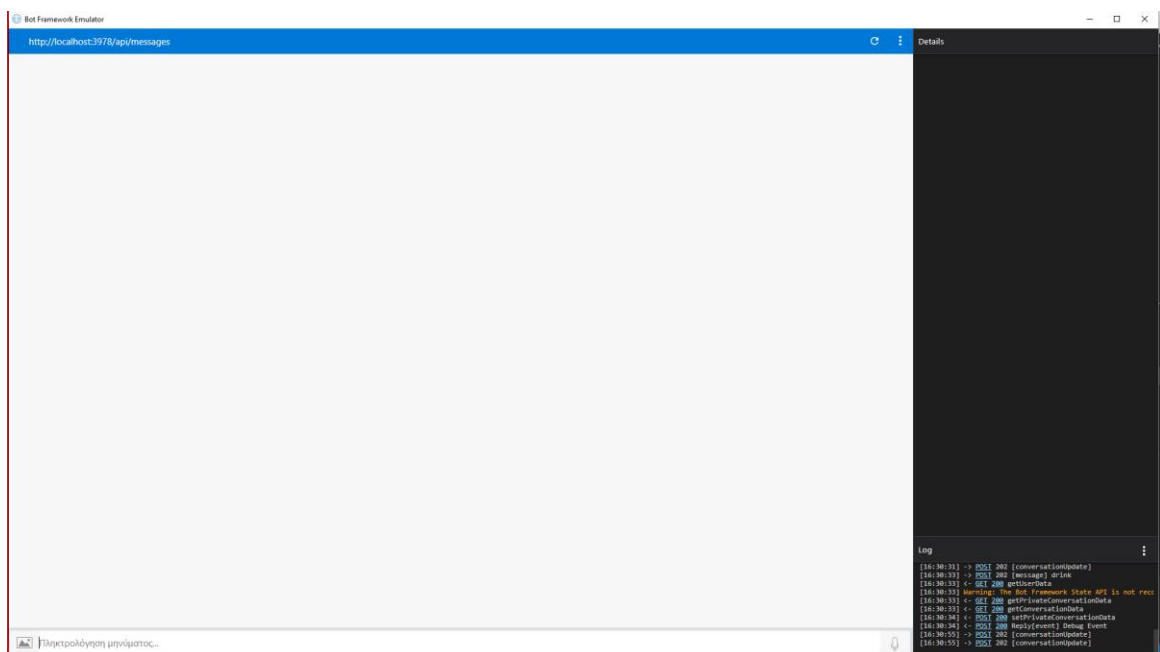
Βλέπουμε παρακάτω τον compiler/ editor μας , όπου αριστερά διακρίνεται η ταξινόμηση των φακέλων της εφαρμογής μας, στο κάτω άκρο το τερματικό για να ξεκινάμε την εφαρμογή και για να κατεβάζουμε διάφορα πακέτα και τέλος στην εργαλειοθήκη αριστερά μας δίνεται η δυνατότητα να κάνουμε διάφορες αναβαθμίσεις, να αναζητήσουμε αρχεία και να αξιοποιήσουμε το debugging mode.



**Εικόνα 86 - Visual Studio Code – (δικό μας)**

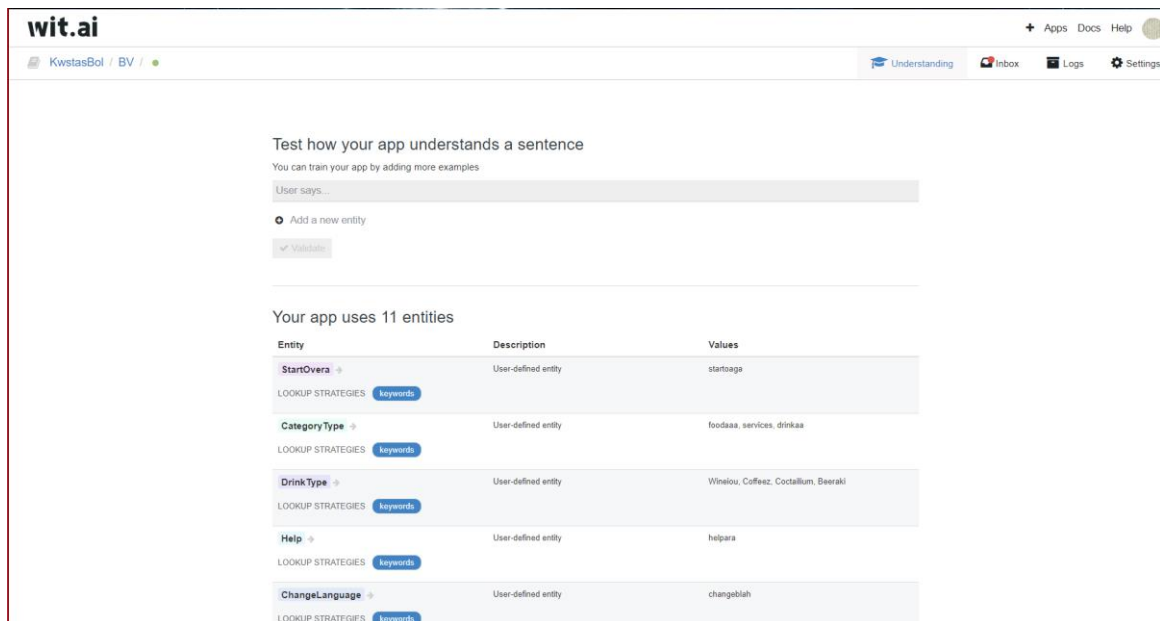
### 7.2.3 Bot Framework Emulator

Στην είσοδο που βάζουμε το url, διαμορφώνουμε τη σύνδεση με το node.js, ώστε να μπορούμε να τεστάρουμε την εφαρμογή μας πριν δημοσιευτεί στο διαδίκτυο. Κάτω δεξιά παρατηρούμε τα logs, που μας διευκρινίζουν αν κάτι απέκλινε από τη φυσιολογική λειτουργία του.



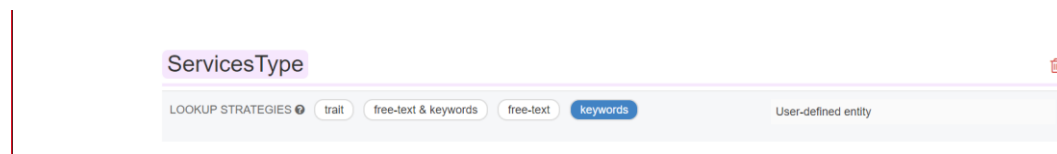
**Εικόνα 87 - Bot Framework Emulator – (δικό μας)**

## 7.2.4 Wit.ai

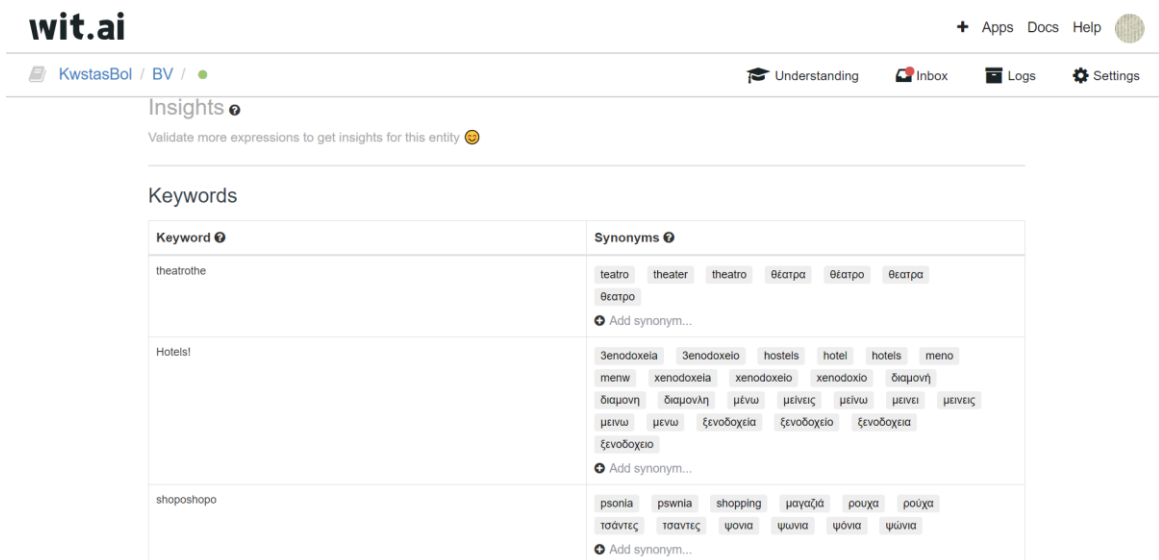


Εικόνα 88 - Wit Main Panel – Wit.ai( δικό μας)

Στο input (User says...) γράφουμε ως διαχειριστές μία πρόταση για επικύρωση σε σχέση με μία οντότητα. Στο κάτω μέρος βλέπουμε τις οντότητες που χρησιμοποιεί η εφαρμογή μας. Ας δούμε και το γραφικό περιβάλλον μίας οντότητας στην παρακάτω εικόνα.



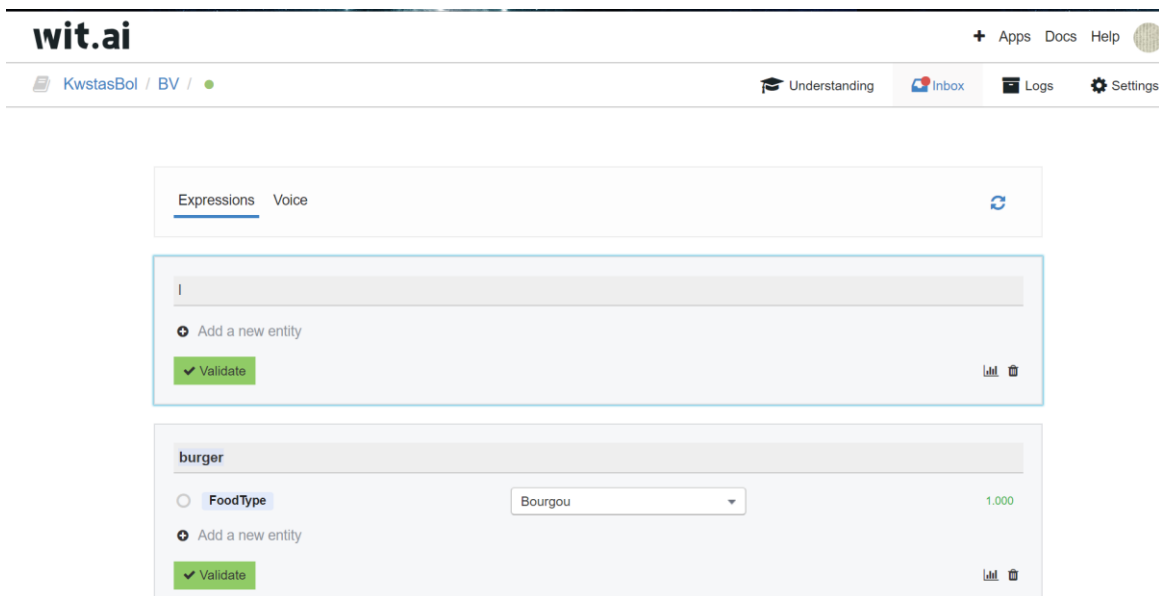
Εικόνα 89 – Wit Entity Panel – Wit.ai( δικό μας)



**Εικόνα 90 – Wit Entity Panel\_2 – Wit.ai( δικό μας)**

Παραπάνω φαίνεται μία συγκεκριμένη οντότητα (ServicesType), η οποία είναι τύπου keywords. Επίσης, βλέπουμε στην αριστερή στήλη τις μεταβλητές (Θέατρο, Ξενοδοχεία, Ψώνια), τα οποία αντιστοιχούν στα συνώνυμα της δεξιάς στήλης.

Παρακάτω στην καρτέλα inbox, επικυρώνουμε την αντιστοίχιση μηνυμάτων από το messenger με τις κατάλληλες οντότητες.



**Εικόνα 91 – Wit Inbox Panel – Wit.ai( δικό μας)**

Στην παρακάτω εικόνα φαίνεται η καρτέλα ρυθμίσεις, στις οποίες μπορούμε να παραμετροποιήσουμε διάφορες παραμέτρους, όπως για παράδειγμα τη γλώσσα της εφαρμογής, βάση της οποίας πραγματοποιείται το NLU(Greek).

Type something to cURL

```
curl \
-H 'Authorization: Bearer 4RZ3GZTLM6KIIIPFMROKAXZYJNQLI2JV' \
'https://api.wit.ai/message?v=20180516&q='
```

[Learn more](#)

### Change App Details

App name

Default Timezone

Language

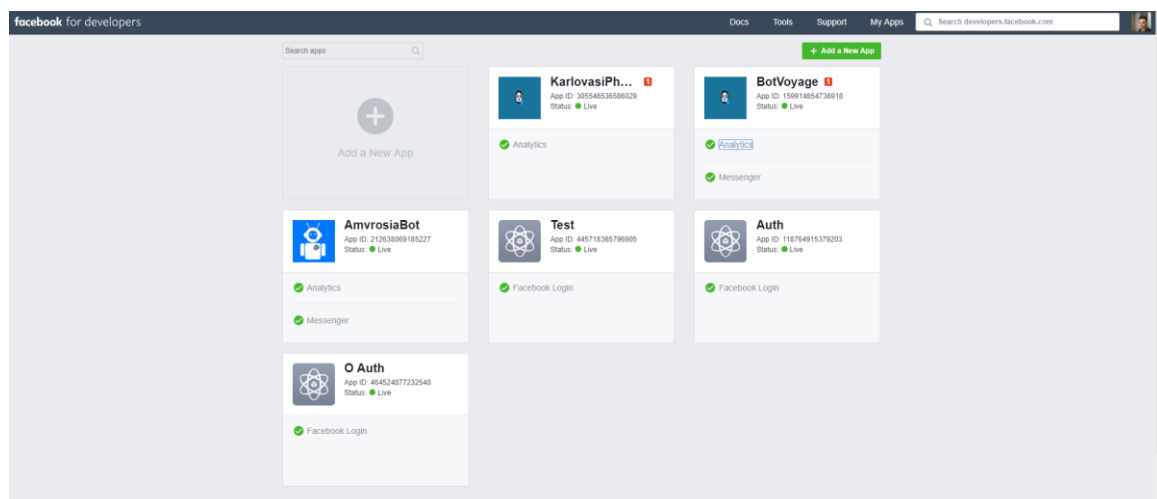
**BETA** Some entities might not work as expected. [Help us extend it!](#)

[Change App Details](#)

Εικόνα 92 - Wit Settings Panel – Wit.ai( δικό μας)

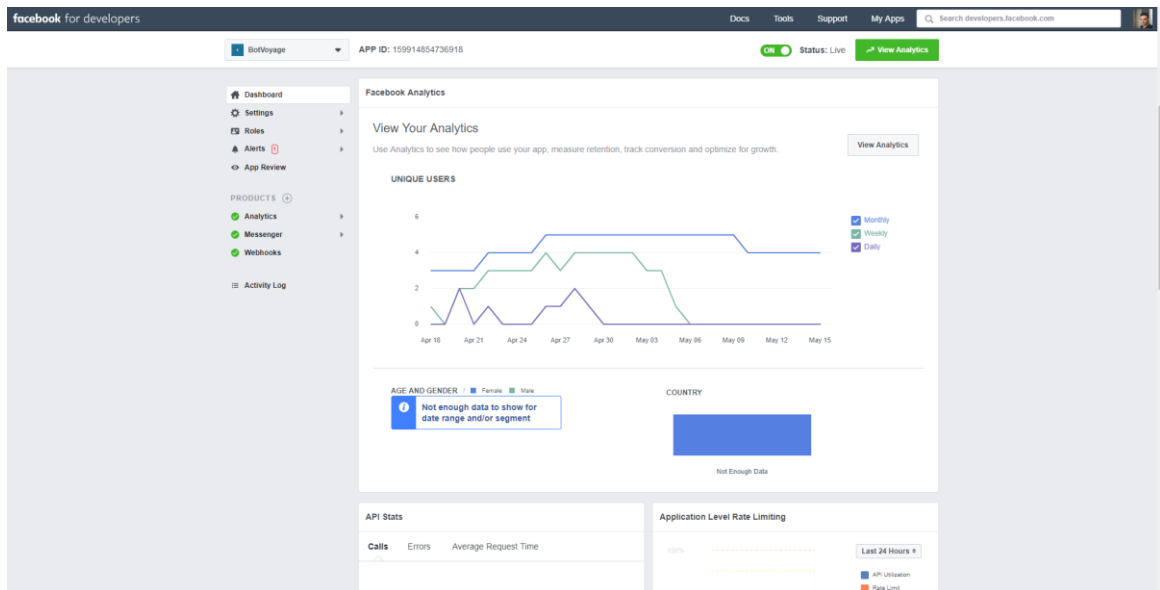
## 7.2.5 Facebook for Developers Platform

Σε πρώτο πλαίσιο ας διακρίνουμε το γενικότερο dashboard ενός developer, κι ειδικότερα στο συγκεκριμένο στιγμιότυπο φαίνονται οι εφαρμογές που μπορείς να διαχειριστείς.



Εικόνα 93 - Facebook Dev Panel – ( δικό μας)





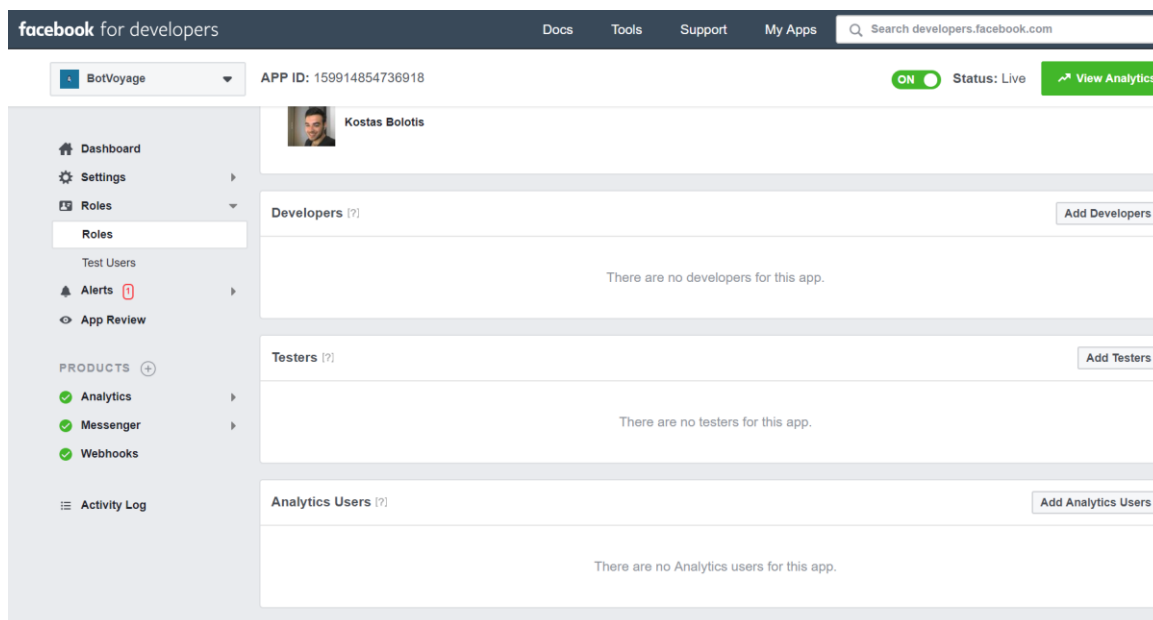
**Εικόνα 94 - Facebook Dev App Panel – ( δικό μας)**

Στην παραπάνω εικόνα βλέπουμε το dashboard της εφαρμογής μας, που όπως φαίνεται αριστερά έχουμε διάφορες επιλογές. Ας δούμε κάποιες επιλογές ενδεικτικά.

Παρακάτω μπορούμε να επιλέξουμε το λογότυπο, το όνομα, τον κωδικό, ακόμη και το έγγραφο που περιλαμβάνει το privacy policy

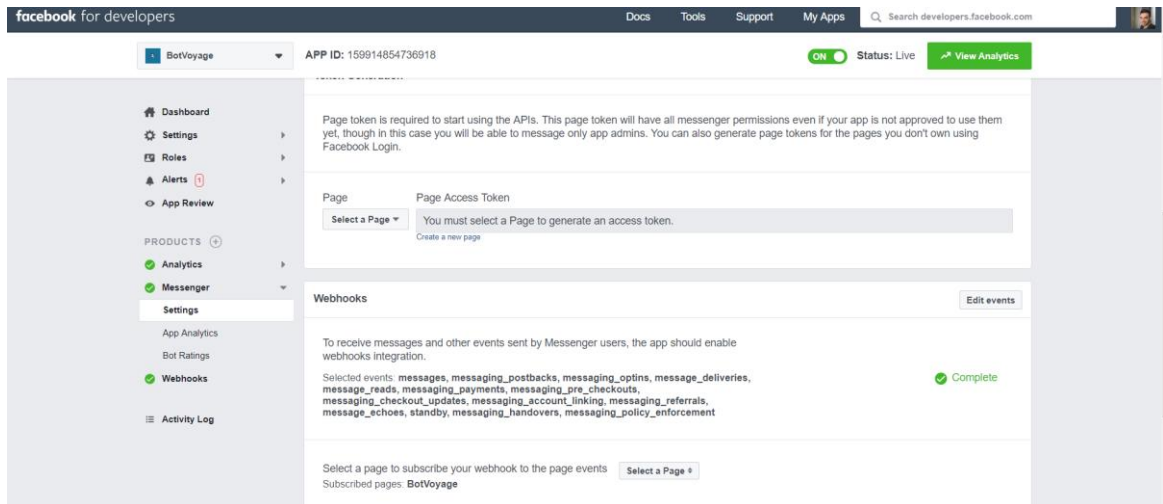
**Εικόνα 95 - Facebook Dev Settings – ( δικό μας)**

Στην παρακάτω εικόνα βλέπουμε το γραφικό περιβάλλον, μέσω του οποίου μπορούμε να ορίσουμε τα άτομα που θα ασχολούνται με την εφαρμογή είτε αυτά είναι διαχειριστές, προγραμματιστές και ούτω καθεξής.



**Εικόνα 96 - Facebook Dev Roles – ( δικό μας)**

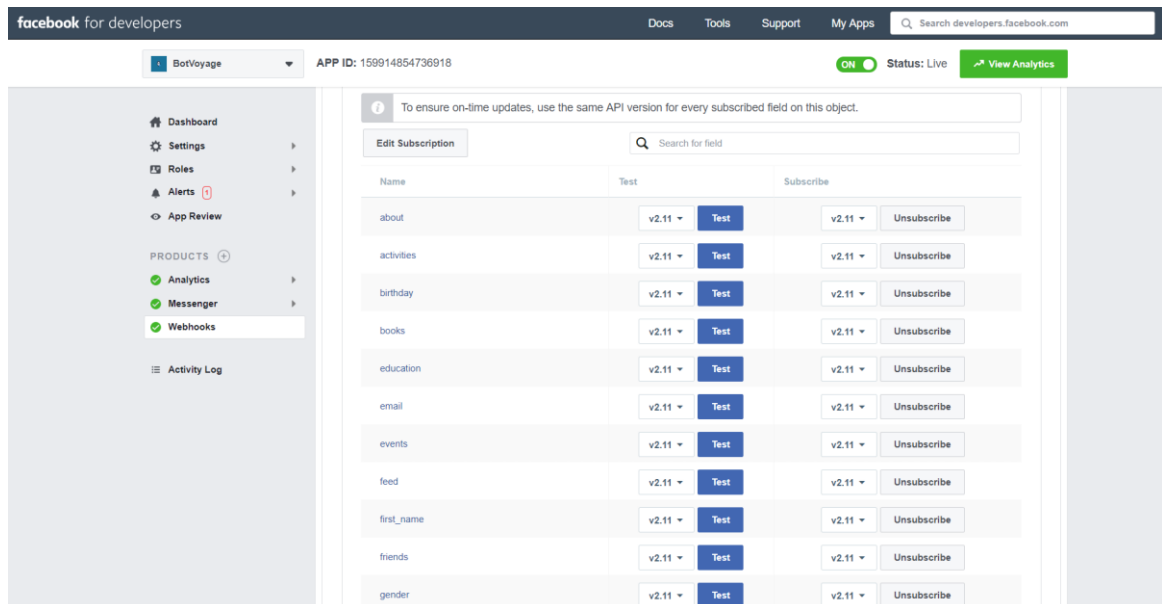
Αξίζει επίσης να σημειωθεί η καρτέλα products και συγκεκριμένα η Messenger -> Settings, στην οποία ουσιαστικά μπορείς να επιτελέσεις διάφορες λειτουργίες. Αρχικά, απαραίτητο βήμα είναι στο Token Generation να επιλέξεις τη σελίδα που θες για να πάρεις το απαραίτητο token και να το συνδέσεις με την πλευρά του azure portal. Επιπρόσθετα στο σημείο που διαφαίνεται η έννοια Webhooks, μπορείς να δεις τι ακριβώς μπορεί να κάνει. Για παράδειγμα με το Selected events: messages μπορεί πλέον ο χρήστης να στείλει μήνυμα στη σελίδα.



**Εικόνα 97 - Facebook Dev Products – (δικό μας)**

Τέλος αξίζει να δούμε και την σελίδα των Webhooks που εκεί του ορίζει ο διαχειριστής τι μπορεί να κάνει ακριβώς η εφαρμογή διαμέσου της πλατφόρμας του Messenger ( δηλαδή ότι βλέπει στα Settings, αλλάζει από την παρακάτω σελίδα)

**Εικόνα 98 - Facebook Dev Webhooks – (δικό μας)**



Όπως φαίνεται, έχουν γίνει subscribe τα περισσότερα από τα webhooks

## ΚΕΦΑΛΑΙΟ 8 -ΣΥΜΠΕΡΑΣΜΑΤΑ-ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ

### 8.1 Συμπεράσματα επί των εργαλείων

Αρχικά, η επιλογή της γλώσσας προγραμματισμού της εφαρμογής μας ήταν κατά τη γνώμη μας η βέλτιστη διότι εκτός από τον περιορισμό που αναλύσαμε στο κεφάλαιο 4, δεν δημιουργήθηκε κάποιο επιπρόσθετο πρόβλημα. Αυτό συμβαίνει διότι, λόγω προηγούμενης ενασχόλησης στο παρελθόν αλλά και επειδή διαθέτει μεγάλη ευκολία στην χρήση της. Ειδικότερα, η λογική με την οποία συνδυάζεται με το Bot Framework, προσδίδει στον developer μία ευχρηστία, καθώς προσφέρει το λεγόμενο ‘waterfall’, το οποίο ουσιαστικά είναι εφάμιλλο με τη λειτουργία ενός καταρράκτη, κάθε βήμα του οποίου αναπαριστά ένα συγκεκριμένο μέρος της συνομιλίας και μόλις ολοκληρωθεί το ‘νερό’ κυλά προς το επόμενο βήμα, έχοντας ως τελευταίο στάδιο της συνομιλίας εκεί που καταλήγουν όλα τα ‘νερά’ (όλα δηλαδή τα μηνύματα της συνομιλίας του bot με τον φυσικό χρήστη). Επίσης η Javascript συνδυάζεται σε ένα πολύ λειτουργικό περιβάλλον, σε αυτό του Visual Studio Code, το οποίο προσφέρει τόσα εργαλεία και αναβαθμίσεις και φυσικά intellisense( έξυπνη συμπλήρωση κώδικα για μεγαλύτερη ταχύτητα και ευκολία). Ας μην ξεχνάμε και το Node.js, το οποίο ως server-side περιβάλλον διαθέτει το Npm, μέσω του οποίου ως developers έχουμε υπέρογκες δυνατότητες με τις αναρίθμητες βιβλιοθήκες που προσφέρει. Οπότε η Javascript, αποτέλεσε σαφώς την καλύτερη επιλογή. Σε δεύτερο επίπεδο, όσον αφορά τα δεδομένα της εφαρμογής μας, το API του Foursquare μας έδωσε τη δυνατότητα να έχουμε άμεση πρόσβαση στα δεδομένα που επιθυμούσαμε χωρίς κάποιο περιορισμό στη χρήση του που θα μας εμπόδιζε( ο περιορισμός ανάγεται σε πάρα πολλά calls/requests ανά χρονικό διάστημα, τον οποίο και δεν προσεγγίζουμε στο σημείο που είμαστε). Επίσης, μιας και είναι τόσο εδραιωμένη η χρήση του Foursquare, τα δεδομένα ήταν αξιόπιστα και αφορούσαν όλη την υδρόγειο, οπότε και η εφαρμογή μας μπορούσε να επεκταθεί, κλιμακωθεί ευκολότερα. Τέλος προσφέρει τεράστια ποικιλία από διαφορετικά δεδομένα, με αποτέλεσμα να έχουμε λειτουργίες από φαγητό μέχρι και νοσοκομεία χάρη αυτού. Επίσης, μένοντας στα API που χρησιμοποιήσαμε, το Google Geocoding Api, ήταν τόσο απλό που από τη στιγμή που λάμβανες τα απαραίτητα διαπιστευτήρια, με κυριολεκτικά τρεις γραμμές κώδικα, η τοποθεσία σε συντεταγμένες μετατρέποταν σε ονομαστική τοποθεσία πολύ εύκολα. Στη συνέχεια ας προβάλλουμε τα συμπεράσματα μας για την πλατφόρμα NLP που χρησιμοποιήσαμε, το Wit.ai. Με την πρώτη ματιά, το περιβάλλον χρήσης ήταν αρκετά εύκολο ακόμη και για κάποιον που δεν είχε ιδέα περί του θέματος, όπως ακριβώς ήμαστε και εμείς εκείνη τη χρονική περίοδο. Επίσης ήταν αρκετά εύκολο να κατανοήσει κανείς τους διαφορετικούς τύπους οντοτήτων, καθώς είχε αρκετά σαφές documentation γεγονός που συνιστά μεγίστης σημασίας, διότι χωρίς αυτή τη γνώση δεν μπορεί να αναπτυχθεί ένα λειτουργικό σύστημα που να περιλαμβάνει NLP και να εξάγει τα σωστά αποτελέσματα. Ωστόσο ως εργαλείο περιείχε τους περιορισμούς που αναλύσαμε στο κεφάλαιο τέσσερα, όπως ήταν η αδυναμία πλήρης κατανόησης της ελληνικής γλώσσας σε τυχούσες αλλαγές που συμβαίνουν μεταξύ εφάμιλλων διαφορετικών προτάσεων. Τέλος η πλατφόρμα του Messenger, συνιστά ένα εργαλείο με απίστευτες δυνατότητες, μιας και μπορεί να συνθέσει κανείς το bot του όπως ακριβώς το επιθυμεί με λειτουργίες όπως η πραγματοποίηση πληρωμών, ένα πολυσύνθετο γραφικό περιβάλλον και ακόμη και ενσωματωμένο NLP για μεγαλύτερη ευκολία και αποφυγή καθυστερήσεων δικτύου και ας μην ξεχνάμε και το διαχειριστικό περιβάλλον που προσδίδεται με αποτέλεσμα ο developer να μπορεί να παραμετροποιήσει ότι θελήσει, όπως για παράδειγμα να αλλάξει το όνομα της εφαρμογής( bot), να την απενεργοποιήσει, είτε να δημιουργήσει καινούρια webhooks.

## 8.2 Συμπεράσματα επί της εφαρμογής

Η εφαρμογή μας BotVoyage στο στάδιο που βρίσκεται τώρα, μπορούμε να πούμε με σιγουριά ότι δεν απέχει πολύ από αυτό που σχεδιάζαμε σε πρώτο επίπεδο. Ειδικότερα θέλαμε να έχουμε ένα έξυπνο και εύχρηστο λογισμικό, το οποίο θα μπορούσε να κατανοεί οποιαδήποτε πρόταση έστελνε ο χρήστης, είτε αυτή αφορούσε έναν χαιρετισμό, ένα ‘ευχαριστώ’ ή την ενημέρωση του για κάποιο συγκεκριμένο θέμα, όπως για παράδειγμα για τα καλύτερα εστιατόρια που εξειδικεύονται στην παρασκευή πίτσας. Επιπρόσθετα, στοχεύαμε στη διατήρηση των στοιχείων και του ιστορικού του χρήστη, ώστε να του αποστέλλουμε διάφορες προσφορές, ανάλογες με τα ενδιαφέροντά του και τις συνήθειες του. Ωστόσο η λειτουργία αυτή δεν πραγματοποιήθηκε πιο πολύ επειδή δεν το βρήκαμε τελικά σκόπιμο και επειδή δεν εξυπηρετούσε τελικά την εμβέλεια εφαρμογής του BotVoyage. Στοχεύαμε επίσης στην ύπαρξη διαφόρων API, ώστε για παράδειγμα ένας χρήστης της εφαρμογής μας που θα έστελνε τοποθεσία από τις Η.Π.Α να χρησιμοποιούσε το API του Yelp και όχι του Foursquare, μιας και είναι πιο διαδεδομένο το πρώτο στην συγκεκριμένη περιοχή. Ωστόσο, επειδή ήταν χαμηλότερης προτεραιότητας σε σχέση με άλλες απαιτήσεις δεν υλοποιήθηκε. Επίσης, η εφαρμογή μας διαθέτει ένα πολύ απλό γραφικό περιβάλλον, όπως ακριβώς το είχαμε σκεφτεί, χωρίς πολλές κάρτες και κουμπιά, τα οποία υπάρχουν μόνο όταν χρειάζεται και συγκεκριμένα στη διαδικασία απεικόνισης των αποτελεσμάτων προς τον χρήστη. Κατά τα άλλα, η εφαρμογή μας διαθέτει ένα απλό περιβάλλον, όπου πραγματοποιείται μια φυσική συζήτηση ανάμεσα σε bot και χρήστη με την προσθήκη οπτικού υλικού (εικόνες και κάρτες) μόνο την κατάλληλη χρονική στιγμή. Παρατηρώντας την εφαρμογή μας, διακρίναμε κάποιο μικρό χρονικό διάστημα δέκα περίπου δευτερολέπτων που χρειάζεται ο server που βρίσκεται στο Azure να ανταποκριθεί, εάν δεν του έχει σταλθεί αίτημα το τελευταίο χρονικό διάστημα(ας πούμε προσεγγιστικά μεγαλύτερο της μισής ώρας). Το γεγονός αυτό όμως δεν αποτελεί πρόβλημα αρκεί ο χρήστης που θα συνομιλήσει για πρώτη φορά με τον διακομιστή να μην προβληματιστεί και να νομίζει ότι έχει προκύψει κάποια δυσλειτουργία. Ένα ακόμη χαρακτηριστικό που δεν ενσωματώσαμε τελικά στην εφαρμογή μας είναι η ύπαρξη feedback, η ενημέρωση δηλαδή του χρήστη από το λογισμικό μας. Ειδικότερα, όταν προχωρεί στην λειτουργία των πτήσεων ανακατευθύνεται μετά από την είσοδο των απαραίτητων δεδομένων στην κεντρική ιστοσελίδα για την κράτηση/αγορά θέσεων για την συγκεκριμένη πτήση. Μετά από αυτό το βήμα στην αρχική μας σχεδίαση στοχεύαμε να ενημερώνουμε τον χρήστη ότι αφού ολοκληρώσει την αγορά θα του εμφανίζαμε κάποιο ενημερωτικό μήνυμα και στη συνέχεια θα διατηρούσαμε αυτή τη συναλλαγή, ώστε στο μέλλον να τον ενημερώσουμε για κάποια παρόμοια πτήση που πιθανώς θα τον ενδιέφερε. Επίσης ο χρήστης δεν έχει την δυνατότητα να ενημερώσει το bot, εάν οι πληροφορίες του φάνηκαν χρήσιμες, και σε περίπτωση που δε του φάνηκαν, να του έδινε την ευκαιρία να ψάξει μέσω ενός διαφορετικού API, όπως είχαμε σκεφτεί σε πρώιμο στάδιο. Ωστόσο η πολυσύνθετη φύση αυτής της λειτουργίας και η ενασχόληση μας με βασικότερες απαιτήσεις δεν μας οδήγησε στην επίτευξη της. Συμπερασματικά, το BotVoyage αποτελεί μία εύκολη μορφή συνομιλίας σε φυσική γλώσσα ανάμεσα σε χρήστη και bot, όπως ακριβώς σχεδιάζαμε εξ’αρχής, μη έχοντας όμως ενσωματωμένες κάποιες λειτουργίες που θέλαμε να εμπεριέχει, οι οποίες όμως κατείχαν χαμηλότερη προτεραιότητα σε σχέση με τις βασικότερες, οι οποίες και επιτεύχθηκαν. Παρακάτω παρατίθεται ο πίνακας με τις λειτουργικές και της μη λειτουργικές απαιτήσεις με τον κωδικό, τον τίτλο τους και με μία βαθμίδα που κυμαίνεται ανάμεσα σε τρεις τιμές(Μερικώς, Πλήρως, Καθόλου). Βάση αυτών που υλοποιήθηκαν μερικώς και αυτών που δεν υλοποιήθηκαν καθόλου, θα συνεχίσουμε στην επόμενη επικεφαλίδα, για να διευκρινίσουμε τα επόμενα βήματα της εφαρμογής μας.

### Λειτουργικές

|    |                                 |         |
|----|---------------------------------|---------|
| 1  | Σύνδεση στο διαδίκτυο           | Πλήρως  |
| 2  | Κατοχή Messenger/Facebook       | Πλήρως  |
| 3  | Έγκριση αποστολής τοποθεσίας    | Πλήρως  |
| 4  | Επιλογή Γλώσσας                 | Μερικώς |
| 5  | Εμφάνιση επιλογών               | Μερικώς |
| 6  | Επιλογή λειτουργίας             | Πλήρως  |
| 7  | Αλλαγή λειτουργίας              | Πλήρως  |
| 8  | Reset διαλόγου                  | Μερικώς |
| 9  | Ανακατεύθυνση σε κύριο ιστότοπο | Μερικώς |
| 10 | Διαμόρφωση προφίλ               | Καθόλου |
| 11 | Καλό γραφικό περιβάλλον         | Πλήρως  |
| 12 | Ύπαρξη NLP/NLU                  | Πλήρως  |
| 13 | Σύνδεση σε api                  | Πλήρως  |
| 14 | Ενημέρωση χρήστη                | Καθόλου |
| 15 | Ύπαρξη διαφορετικών api         | Καθόλου |

### Μη λειτουργικές

|    |                                   |         |
|----|-----------------------------------|---------|
| 16 | Βέλτιστη ανταπόκριση              | Πλήρως  |
| 17 | Επιλογή διακομιστή                | Πλήρως  |
| 18 | Επιλογή server-side περιβάλλοντος | Πλήρως  |
| 19 | Ασφάλεια στοιχείων χρήστη         | Καθόλου |
| 20 | Αξιόπιστα δεδομένα                | Πλήρως  |
| 21 | Ενημέρωση για σφάλματα            | Μερικώς |
| 22 | Νόμιμη χρήση api                  | Πλήρως  |
| 23 | Εφάμιλλη λειτουργία               | Μερικώς |
| 24 | Ύπαρξη demo                       | Καθόλου |
| 25 | Απουσία χρονικού περιορισμού      | Μερικώς |

### 8.3 Επόμενα βήματα

Ας ξεκινήσουμε να οραματιζόμαστε τα επόμενα βήματα της εφαρμογής μας παίρνοντας για παράδειγμα τις περισσότερες από τις απαιτήσεις που υλοποιήθηκαν μερικώς είτε καθόλου. Αρχικά, στοχεύουμε μελλοντικά στην εκπλήρωση της εξής λειτουργίας. Όταν ο χρήστης θα ανακατευθύνεται σε έναν ιστότοπο για να πραγματοποιήσει κάποια συναλλαγή, όταν την ολοκληρώνει θα έχει μία σχετική ενημέρωση από την πλευρά του bot για το τι επιτεύχθηκε. Δεύτερον, σε μία μελλοντική έκδοση της εφαρμογής μας οραματιζόμαστε την ύπαρξη προφίλ για τον χρήστη. Ειδικότερα, θα αποθηκεύονται σε μία βάση δεδομένων τα στοιχεία του χρήστη, οι τοποθεσίες που έχει αποστείλει και γενικότερα για ποια μέρη έχει ενδιαφερθεί, έτσι ώστε σε γενικότερο πλαίσιο να γνωρίζει το αυτοματοποιημένο λογισμικό της εφαρμογής μας τι θα προτείνει στον χρήστη στο άμεσο μέλλον. Έτσι η εφαρμογή μας θα χαρακτηρίζεται από μία αμεσότητα και διευκόλυνση προς τον χρήστη. Επιπρόσθετα, θα ήταν ιδανικό να περιλαμβάνουμε στην εφαρμογή μας ως επόμενο βήμα, την ύπαρξη διάφορων API, έτσι ώστε να υπάρχει μία ενδιαφέρουσα ποικιλία δεδομένων για να βοηθούν τον χρήστη να έχει μία πολυεπίπεδη όψη διαφορετικών μαγαζιών, υπηρεσιών. Επίσης ένα άλλο στοιχείο που είναι απαραίτητο να υπάρχει σε μελλοντική έκδοση είναι η ενημέρωση από τον χρήστη( διάδραση) εάν ήταν ευχαριστημένος με τα αποτελέσματα που δέχτηκε και σε περίπτωση που δεν , να του αποσταλούν διαφορετικά. Τέλος όσον αφορά τον πίνακα λειτουργιών που παρατίθεται παραπάνω, χρειάζεται και η βελτιστοποίηση της ενημέρωσης του χρήστη για σφάλματα. Ειδικότερα στον τομέα των πτήσεων , υπάρχει κατανόηση από την πλευρά της εφαρμογής εάν ο χρήστης επιλέξει λάθος αεροδρόμιο ή ημερομηνία, αλλά χρειάζεται ένας πιο έξυπνος αλγόριθμος, για να μην γίνονται περιττά βήματα από πλευράς του χρήστη ( π.χ. εάν στείλει λανθασμένη ημερομηνία, δεν θα ειδοποιηθεί εξαρχής, αλλά στο τέλος της συνομιλίας). Στη συνέχεια, ας προχωρήσουμε στην διατύπωση μερικών επόμενων βημάτων που δεν εξαρτώνται από τους παραπάνω πίνακες των απαιτήσεων. Αρχικά, θα θέλαμε να βελτιώσουμε την ήδη υπάρχουσα βάση δεδομένων που έχουμε για τις πτήσεις έτσι ώστε ο χρήστης να μην περιορίζεται στο να στέλνει μόνο π.χ. Athens, αλλά Αθήνα, Αθήνα κ.λ.π. Τέλος, ένα τελικό αισιόδοξο βήμα θα ήταν η διαμόρφωση μίας πλήρους ιστοσελίδας με μία δικιά μας βάση δεδομένων και πλέον δηλαδή να μην αποτελούμε απλά μια εφαρμογή ‘μεσάζοντα’, αλλά να είναι πλέον κύρια, όπως για παράδειγμα το Foursquare.

## BIBΛΙΟΓΡΑΦΙΑ

- [1] Wikipedia. Internet Bots. Wikimedia Foundation, Inc., 2015
- [2] Russell & Norvig (2003) Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall,
- [3] <https://www.techopedia.com/definition/190/artificial-intelligence-ai>
- [4] <https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/>
- [5] Zeifman, Igal. "Bot Traffic Report 2016". Incapsula. Retrieved 1 February 2017.
- [6] Percy Lipinski. CNN's iReport hit hard by pay-per-view scandal. Allvoices, 2013. , [https://oxford-biochron.com/downloads/OxfordBioChron\\_Quantifying-Online-Advertising-Fraud\\_Report.pdf](https://oxford-biochron.com/downloads/OxfordBioChron_Quantifying-Online-Advertising-Fraud_Report.pdf)
- [7] "How companies are using chatbots for marketing: Use cases and inspiration - MarTech Today". MarTech Today. 2018-01-22. Retrieved 2018-04-10.
- [8] Wikipedia, First-Class function, [https://en.wikipedia.org/wiki/First-class\\_function](https://en.wikipedia.org/wiki/First-class_function)
- [9] Wikipedia, Prototype-based programming, [https://en.wikipedia.org/wiki/Prototype-based\\_programming](https://en.wikipedia.org/wiki/Prototype-based_programming)
- [10] Wikipedia, Imperative programming, [https://en.wikipedia.org/wiki/Imperative\\_programming](https://en.wikipedia.org/wiki/Imperative_programming)
- [11] Wikipedia, Functional programming, [https://en.wikipedia.org/wiki/Functional\\_programming](https://en.wikipedia.org/wiki/Functional_programming)
- [12]MDN web docs, About Javascript, [https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript)
- [13] Jensen, S. H., Møller, A., & Thiemann, P. (2009, August). Type analysis for JavaScript. In International Static Analysis Symposium (pp. 238-255). Springer, Berlin, Heidelberg.
- [14]Wikipedia , History , <https://en.wikipedia.org/wiki/JavaScript#History>
- [15] Wikipedia, Internet Information Services , [https://en.wikipedia.org/wiki/Internet\\_Information\\_Services](https://en.wikipedia.org/wiki/Internet_Information_Services)
- [16] Jenna Sheffield, The Ultimate Travel Bot List, <https://www.30secondstofly.com/ai-software/ultimate-travel-bot-list/>,2016
- [17] Mikkonen, T., & Taivalsaari, A. (2007). Using JavaScript as a real programming language.
- [18]Brad Bourque, 15 Awesome HTML5 and Javascript games we've played on the Web,<https://www.digitaltrends.com/gaming/best-html-5-javascript-games/>, 2014
- [19]Eric Elliot, 10 Must See Web Apps & Games, <https://medium.com/javascript-scene/10-must-see-web-apps-games-36ab9ca60754>, 2016
- [20] Eugeniya Korotya, The Ultimate List of 38 JavaScript Tools, <https://hackernoon.com/the-ultimate-list-of-javascript-tools-e0a5351b98e3> , 2017
- [21] Wikipedia, History, <https://en.wikipedia.org/wiki/Node.js#History>
- [22] Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2017). Node. js in Action. Manning Publications.



- [23] Node.js, About, <https://nodejs.org/en/>
- [24] Developers Google, Chrome V8, <https://developers.google.com/v8/>
- [25] Wikipedia, I/O bound, [https://en.wikipedia.org/wiki/I/O\\_bound](https://en.wikipedia.org/wiki/I/O_bound)
- [26] Byte Archer, I/O-bound vs CPU-bound in Node.js, <https://bytearcher.com/articles/io-vs-cpu-bound/>
- [27] ThinkMobiles, 10 best Node.js app examples for enterprises, with metrics, <https://thinkmobiles.com/blog/node-js-app-examples/>
- [28] Lee Stott, What is Microsoft Bot Framework Overview, [https://blogs.msdn.microsoft.com/uk\\_faculty\\_connection/2016/04/05/what-is-microsoft-bot-framework-overview/](https://blogs.msdn.microsoft.com/uk_faculty_connection/2016/04/05/what-is-microsoft-bot-framework-overview/)
- [29] Robert Standefer, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-3.0#publish>
- [30] Matt Shim, Create bots with Botbuilder Templates, <https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-create-templates?view=azure-bot-service-3.0>
- [31] Duc Cash Vo, Kamran Iqbal, Robert Standefer, Create a bot with Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-quickstart?view=azure-bot-service-3.0>
- [32] Jason Sowers, Vischa Csená, Bot Builder SDK, <https://github.com/Microsoft/BotBuilder>
- [33] Robert Standefer, John Dempsey, Dean Kroker, Create a bot with the Bot Builder SDK for .NET, <https://docs.microsoft.com/en-us/azure/bot-service/dotnet/bot-builder-dotnet-quickstart?view=azure-bot-service-3.0>
- [34] Denise Mac, John Dempsey, Duc Cash Vo, Create a bot with the Bot Builder SDK for Node.js, <https://docs.microsoft.com/en-us/azure/bot-service/nodejs/bot-builder-nodejs-quickstart?view=azure-bot-service-3.0>
- [35] Brian A. Randell, Kamran Iqbal, Bot scenarios , <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-scenario-overview?view=azure-bot-service-3.0>
- [36] Rachel Vanier, The (perfect) startup story of Wit.ai, acquired by Facebook in 21 months, <https://medium.com/unexpected-token/the-perfect-startup-story-of-wit-ai-acquired-by-facebook-in-21-months-1bf35b996808>, 2015
- [37] Josh Constine, Facebook Acquires Wit.ai To Help Its Developers With Speech Recognition And Voice Interfaces, <https://techcrunch.com/2015/01/05/facebook-wit-ai/>, 2015
- [38] Wikipedia, Natural-language processing, [https://en.wikipedia.org/wiki/Natural-language\\_processing](https://en.wikipedia.org/wiki/Natural-language_processing)
- [39] Lola, NLP vs. NLU: What's the Difference? , <https://medium.com/@lolatravel/nlp-vs-nlu-whats-the-difference-d91c06780992>, 2016
- [40] Braun, D., Hernandez-Mendez, A., Matthes, F., & Langen, M. (2017). Evaluating natural language understanding services for conversational question answering systems. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (pp. 174-185).

- [41] Wit.ai, <https://wit.ai/>
- [42] Wit.ai Docs, Quickstart, <https://wit.ai/docs/quickstart>
- [43] Wikipedia, npm(software), [https://en.wikipedia.org/wiki/Npm\\_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))
- [44] Npm Docs, What is npm, <https://docs.npmjs.com/getting-started/what-is-npm>
- [45] Npm Docs, How to Install Local Packages, <https://docs.npmjs.com/getting-started/installing-npm-packages-locally>
- [46] Wikipedia, History, <https://en.wikipedia.org/wiki/Foursquare#History>
- [47] Foursquare, About Foursquare, <https://foursquare.com/about>
- [48] Foursquare Developers, Getting Started, <https://developer.foursquare.com/docs/api/getting-started>
- [49] Google Maps Platform, Geocoding Api, <https://developers.google.com/maps/documentation/geocoding/intro>
- [50] Facebook for Developers, Introduction, <https://developers.facebook.com/docs/messenger-platform/introduction>
- [51] Facebook for Developers, Messaging, <https://developers.facebook.com/docs/messenger-platform/send-messages>
- [52] Facebook for Developers, Payments (beta), <https://developers.facebook.com/docs/messenger-platform/payments>
- [53] Facebook for Developers, Discovery & Re-engagement, <https://developers.facebook.com/docs/messenger-platform/discovery>
- [54] Facebook for Developers, IDs & Profile Information, <https://developers.facebook.com/docs/messenger-platform/identity>
- [55] Facebook for Developers, Built-in NLP, <https://developers.facebook.com/docs/messenger-platform/built-in-nlp>
- [56] Facebook for Developers, Messenger Bot Analytics, <https://developers.facebook.com/docs/messenger-platform/analytics>
- [57] Wikipedia, Webhook, <https://en.wikipedia.org/wiki/Webhook>
- [58] Facebook for Developers, Webhooks, <https://developers.facebook.com/docs/messenger-platform/webhook>

## ΠΑΡΑΡΤΗΜΑ-[ΟΔΗΓΙΕΣ ΑΝΑΠΤΥΞΗΣ]

Παρατίθενται παρακάτω τα βήματα δημιουργίας ενός bot εφάμιλλου με το δικό μας.

1. Δημιουργούμε ένα Facebook Page στο προσωπικό μας λογαριασμό, όπου θα έχει το όνομα καλύτερα του μελλοντικού μας bot. Δημιουργία => Σελίδα=> Επιχείρηση ή Επωνυμία/Κοινότητα. Δίνουμε Page Name και Category, προσθέτουμε εικόνες και είμαστε έτοιμοι.
2. Δημιουργούμε ένα λογαριασμό στο <https://developers.facebook.com/>. Αφού συνδεθούμε, πηγαίνουμε στην καρτέλα MyApps και κάνουμε προσθήκη μίας καινούριας με το 'Add New App', συμπληρώνουμε το όνομα και στο παράθυρο που ανοίγει στο 'Add a Product', επιλέγουμε το 'Messenger' και στη συνέχεια το Webhooks.
3. Δημιουργούμε λογαριασμό στο <https://portal.azure.com>. Αφού συνδεθούμε με τα credentials που δημιουργήσαμε, παρατηρούμε ότι έχουμε μπροστά μας το dashboard μας, τον χώρο δηλαδή εργασίας που περιέχει όλα τα προγράμματα μας- εφαρμογές μας. Ήρθε η ώρα να δημιουργήσουμε ένα bot. Στο πάνω μέρος αριστερά, επιλέγουμε το 'Create a resource', και στην αναζήτηση πληκτρολογούμε Web App Bot. Επιλέγουμε το πρώτο που εμφανίζει και πατάμε 'create'. Συμπληρώνουμε τη λίστα με τα στοιχεία και προσέχουμε τα εξής: α) Στο Bot template επιλέγουμε Node.js και συγκεκριμένα Form, β) στο location επιλέγουμε κάποιον server που να βρίσκεται σχετικά κοντά μας κι όχι π.χ. στην Αμερική. Αφού ολοκληρωθεί το deployment, πηγαίνουμε στο All Resources και βρίσκουμε αυτό που μόλις δημιουργήσαμε κι επιλέγουμε από τα δύο που φαίνονται το Web App Bot κι όχι για την ώρα τουλάχιστον το App Service. Παρατηρείται ότι υπάρχει μία λίστα με επιλογές. Ας ξεκινήσουμε με το Channels, όπου εκεί θα γίνει η διασύνδεση με το messenger( είτε όποιο άλλο κανάλι επιθυμείτε). Επιλέγοντας το Messenger πρέπει να συμπληρώσουμε τα παρακάτω στοιχεία.
  - Facebook Page ID είναι ο αριθμός που αναγράφεται στο url της σελίδας σου μετά το όνομα, π.χ. ( <https://www.facebook.com/BotVoyage-141690516594000/> είναι το 141690516594000)
  - Facebook App ID είναι αυτό που αναγράφεται πάνω πάνω στην εφαρμογή που δημιουργήσαμε στο Facebook for Developers.
  - Facebook App Secret είναι ο κωδικός που βρίσκεται στο Facebook for Developers στο Settings=>Basic=>App Secret
  - Page Access Token βρίσκεται στο Facebook for Developers Products=>Messenger=>Settings=>Token Generation=>Select a Page. Επιλέγουμε τη σελίδα μας που φτιάξαμε στο βήμα 1, και γεννάται ένα token.
4. Πηγαίνουμε στο Facebook for Developers και αντιγράφουμε τα δύο τελευταία στοιχεία που αναγράφονται στο Azure Portal ( Callback Url, Verify Token) και συγκεκριμένα στο path Products=>Webhooks=>Subscribe to this object.
5. Παίρνουμε τα api keys για το Foursquare, Google Geocoding Api, όπως έχουμε εξηγήσει παραπάνω, καθώς είναι απαραίτητα για ταυτοποίηση.
6. Δημιουργούμε λογαριασμό στο <https://wit.ai/> και επιλέγουμε το + για να δημιουργήσουμε μία νέα εφαρμογή. Τοποθετούμε το όνομα και προσοχή να επιλέξουμε στο Language τη γλώσσα που επιθυμούμε( ελληνικά), και ας είναι open στο κοινό.

Οπότε τώρα είμαστε έτοιμοι να πραγματοποιήσουμε οποιαδήποτε λειτουργία όπως έχουμε παρουσιάσει στα σχετικά κεφάλαια.

7. Όσον αφορά τον κώδικα είτε μπορεί να αναπτυχθεί τοπικά, είτε αμέσως σε ζωντανό χρόνο στο Azure Portal. Επιλέγοντας τη δεύτερη επιλογή, πηγαίνουμε στο Web App Bot, διαλέγουμε το Build και επιλέγουμε το 'Open Online Code Editor'. Στο αρχείο app.js θα γίνει η ανάπτυξη του bot μας.
8. Παρακάτω θα εξηγηθούν συγκεκριμένα κομβικά κομμάτια του κώδικα:

- Εκκίνηση server, σε αυτόν στέλνονται τα μηνύματα του χρήστη και επικοινωνεί με το messenger.

```
var server = restify.createServer();  
server.listen(process.env.port || process.env.PORT || 3978, function () {  
  console.log('%s listening to %s', server.name, server.url);  
});  
server.post('/api/messages', connector.listen());
```

- Σύνδεση με το messenger

```
var connector = new builder.ChatConnector({  
  appId: process.env.MicrosoftAppId,  
  appPassword: process.env.MicrosoftAppPassword,  
  openIdMetadata: process.env.BotOpenIdMetadata  
});
```

- Σύνδεση με το Wit.ai

```
bot.recognizer(new WitRecognizer('api_key'));
```

- Σύνδεση με το Foursquare

```
var foursquare = (require('foursquarevenues'))(client_id, 'client_secret');
```

- Λήψη των δεδομένων για τις πτήσεις

```
var aviationJson = require("aviation-json");  
var airports = aviationJson.airports;  
var cityAirports = aviationJson.city_airports;
```

- Λήψη δεδομένων από το Foursquare

```
foursquare.exploreVenues(params, function (error, venues) {
  if (!error) {

    for (var i = 0; i < params.limit; i++) {
      console.log(venues.response);
      if (venues.response.groups[0].items[i] != undefined) {
        places[i] = venues.response.groups[0].items[i].venue;

      }
      else if(venues.response.totalResults==0){
        wrong_location=true;
      }

    }

  }
}
```

- Εμφάνιση δεδομένων σε κάρτα carousel

```
var msg = new builder.Message(session);
msg.attachmentLayout(builder.AttachmentLayout.carousel);
msg.attachments([
  places[0]!=undefined?new builder.HeroCard(session)
    .title(places[0].name)
    .subtitle(places[0].location.address + "-" + places[0].location.city )
    .buttons([builder.CardAction.openUrl(session,
"https://foursquare.com/v/" + places[0].name + "/" + places[0].id,
session.userData.lingua == 'greek' ?"Μια καλύτερη ματιά?":"View more?")])
    .images([builder.CardImage.create(session,
res[0].response.photos.items[0] != undefined ? "" +
res[0].response.photos.items[0].prefix + "300x300" +
res[0].response.photos.items[0].suffix + "" :
"http://www.kashmirnewsobserver.com/newsimages/noimage.jpg")):null
```

Διαμορφώνουμε ένα μήνυμα `msg` και του περνάμε κάποια γνωρίσματα μέσω του `msg.attachments` όπως είναι ο τίτλος, ο υπότιτλος και τα κουμπιά που ανακατευθύνουν στο Foursquare.

- Αποκωδικοποίηση τοποθεσίας

```
function findLocation(session,results){
  if (results) {
    var location = results.response.entity;
    var config = {
      'latitude': location.coordinates.lat,
      'longitude': location.coordinates.long,
      'key': 'AIzaSyAISRr4hhYH1TptRNZckM0bpM1oK-vGjyk'
    };

    geocoding.location(config, function (err, data) {
      if (err) {
        console.log(err);
      } else {
        findPlaces(session,
(data.results[2].address_components[1].long_name));
      }
    });
  }
}
```

Λαμβάνουμε τις συντεταγμένες( `latitude,longitude`) και μέσω του `geocoding` τις αποκωδικοποιούμε και παίρνουμε το `data.results[2].address_components[1].long_name` που ισοδυναμεί με το όνομα του μέρους.

- Επεξήγηση διαλόγου χαιρετισμού(όλοι οι διάλογοι συμπεριφέρονται έτσι)

```
bot.dialog('Hello', [
  function (session, results) {
```

```

//Check if it is greek, then begin Greek_Intro
session.userData.lingua = "greek";
session.beginDialog('Intro');
//Else begin English_Intro
},
//Step 2- End of dialog
function (session, results) {
    session.endDialog();
}

]).triggerAction({
    matches: 'Language.ellhnasego'
});

```

Παρατηρούμε το όνομα του διαλόγου, το οποίο είναι απαραίτητο για να μεταφερθούμε εάν το επιθυμούμε μέσα από έναν άλλο σε αυτόν. Σε όλους τους διαλόγους υπάρχει ένα array διαλόγων ( []). Ο καθένας αναπαριστά κι ένα βήμα του καταρράκτη. Κάθε βήμα περιέχει ένα function με ορίσματα το session και το results. Με το session μπορείς να αποθηκεύσεις δεδομένα, να ξεκινήσεις έναν άλλο διάλογο και τα λοιπά. Με το όρισμα results, λαμβάνεις τα αποτελέσματα ενός προηγούμενου βήματος. Η συνάρτηση triggerAction διαμορφώνει ένα ταίριασμα με τις οντότητες της πλατφόρμας του NLP , Wit.ai.

## **ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ ΣΗΜΕΙΩΜΑ**

**[ΟΝΟΜΑ ΦΟΙΤΗΤΗ/ΤΡΙΑΣ]**

Εδώ προστίθεται ένα σύντομο βιογραφικό σημείωμα του φοιτητή σε διηγηματική μορφή.