

Διπλωματική Εργασία

της φοιτήτριας του Τμήματος Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του
Πανεπιστημίου Πατρών

Βαλαχή Άννας του Δημητρίου

Αριθμός Μητρώου : 227465

Θέμα

«Σχεδίαση και μελέτη εμπειρίας χρήσης υπηρεσίας ERP»

Επιβλέπων
Αβούρης Νικόλαος

Αριθμός Διπλωματικής Εργασίας:

Πάτρα, Μάρτιος 2018

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με θέμα

«Σχεδίαση και μελέτη εμπειρίας χρήσης υπηρεσίας ERP»

Της φοιτήτριας του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

Βαλαχή Άννας του Δημητρίου

Αριθμός Μητρώου : 227465

Παρουσιάστηκε δημόσια και εξετάστηκε στο Τμήμα Ηλεκτρολόγων
Μηχανικών και Τεχνολογίας Υπολογιστών στις
06/03/2018

Ο Επιβλέπων

Ο Διευθυντής του Τομέα

Νικόλαος Αβούρης
Καθηγητής

Ευθύμιος Χούσος
Καθηγητής

Αριθμός Διπλωματικής Εργασίας:

Θέμα: «Σχεδίαση και μελέτη εμπειρίας χρήσης υπηρεσίας ERP»

Φοιτήτρια: Άννα Βαλαχή

Επιβλέπων: Νικόλαος Αβούρης

Περίληψη

Τα τελευταία χρόνια όλο και περισσότερες εταιρείες χρησιμοποιούν «συστήματα επιχειρησιακού σχεδιασμού», δηλαδή ERP συστήματα, ώστε να είναι σε θέση να διαχειριστούν καλύτερα τις λειτουργίες τους. Αυτό έχει ως αποτέλεσμα η εμπειρία χρήσης τέτοιων συστημάτων να είναι όλο και πιο σημαντική. Υπό αυτό το πρίσμα, δημιουργήσαμε έναν ψηφιακό επιχειρησιακό βοηθό σε μορφή bot το οποίο ενσωματώνεται στο σύστημα επιχειρησιακού σχεδιασμού που εμπορεύεται η εταιρεία Softone Technologies, το SoftOne. Με τον συγκεκριμένο επιχειρησιακό βοηθό, θα έχει τη δυνατότητα να επικοινωνεί κάθε χρήστης του SoftOne ERP και να λαμβάνει σημαντικές πληροφορίες που αφορούν τα δεδομένα της επιχείρησης του ή της επιχείρησης στην οποία εργάζεται. Με τον όρο bot ορίζουμε μία αυτοματοποιημένη εφαρμογή (με την μορφή συνομιλητή) που χρησιμοποιείται για την εκτέλεση απλών και επαναλαμβανόμενων εργασιών.

Στη εργασία αυτή, αρχικά αναλύεται τι είναι ένα bot και πως μπορεί να αλλάξει-βελτιώσει τη ζωή των σύγχρονων ανθρώπων. Έπειτα παρουσιάζονται διάφορες τεχνολογίες που μπορούν να προστεθούν σε bots και να αυξήσουν τις δυνατότητες τους, καθώς και παραδείγματα άλλων bots που έχουν δημιουργηθεί για να επιλύσουν διαφορετικά ή και παρόμοια προβλήματα. Έπειτα, ακολουθεί μία αναλυτική παρουσίαση των συστημάτων που χρησιμοποιήθηκαν για την δημιουργία του επιχειρησιακού βοηθού και εξηγείται πως επιτεύχθηκε η επικοινωνία μεταξύ τους. Ακόμη, δείχνονται αναλυτικά οι λειτουργίες και οι δυνατότητες του bot.

Τέλος, γίνεται μία αξιολόγηση της ευχρηστίας του επιχειρησιακού βοηθού με άξονες την αποτελεσματικότητα, την αποδοτικότητα και την ικανοποίηση των χρηστών και προτείνονται μελλοντικές επεκτάσεις για το συγκεκριμένο έργο.

Abstract

In recent years, more and more companies have been using "business planning systems" (ERP) in order to be able to better manage their operations. As a result, the experience of using such systems is increasingly important. In this respect, we have created a digital bot business assistant that integrates into the ERP system traded by the company Softone Technologies (SoftOne). Every user of Softone ERP will have the opportunity to communicate with the business assistant and receive significant information about his business. Bot is defined as an automated application (in the form of a contact) that is used to perform simple and repetitive tasks.

In this thesis, initially we analyze what a bot is and how it can change-improve the lives of modern people. Then, we present various technologies that can be added to bots and increase their capabilities, as well as examples of other bots that have been created to solve different or similar problems. Then, there is a detailed presentation of the systems used to create the business assistant and it is explained how communication has been achieved between them. In addition, the functions and capabilities of the bot are shown in detail.

Finally, an assessment of the usability of the digital assistant is carried out, focusing on the effectiveness, efficiency and satisfaction of users and future extensions for the project are proposed.

Πρόλογος

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε κατά τη διάρκεια της πρακτικής μου άσκησης στην εταιρεία SoftOne Technologies και συνεχίστηκε και στο επόμενο διάστημα. Αντικείμενο της αποτελεί η ενσωμάτωση συστήματος λεκτικής επικοινωνίας (με τη μορφή bot) στο βασικό σύστημα μηχανογράφησης (ERP σύστημα) που εμπορεύεται η συγκεκριμένη εταιρεία. Αποτελεί μία προσπάθεια σχεδίασης και μελέτης ενός διαφορετικού και πρωτοποριακού User Interface και του σε ποιο βαθμό αυτό μπορεί να επηρεάσει την εμπειρία του χρήστη.

Αναλυτικά, στο **κεφάλαιο 1** γίνεται παρουσίαση και εξήγηση του αντικειμένου και του στόχου της εργασίας. Παρουσιάζεται αναλυτικά η δομή όλου του έργου και εξηγείται ποια μέρη αυτού ήταν δική μου αρμοδιότητα και ως εκ τούτου αποτελούν αντικείμενο ανάλυσης στην εργασία αυτή.

Στο **κεφάλαιο 2**, παρουσιάζεται η σχετική εργασία που έχει πραγματοποιηθεί και αφορά το συγκεκριμένο θέμα. Παρατίθενται άλλες περιπτώσεις στις οποίες έχει χρησιμοποιηθεί “συνομιλητικό” User Interface και γίνεται μία περιληπτική παρουσίαση των δυνατοτήτων τους. Ακόμη, γίνεται επεξήγηση των δυνατοτήτων που δίνει στα bots ένα από τα συστήματα που χρησιμοποιήσαμε, το Microsoft Bot Framework.

Στο **κεφάλαιο 3**, γίνεται μία αναλυτική παρουσίαση των συστημάτων που έχουν χρησιμοποιηθεί για την υλοποίηση του συστήματος λεκτικής επικοινωνίας που δημιουργήσαμε και επεξηγείται πως πραγματοποιήθηκε η επικοινωνία μεταξύ αυτών.

Στο **κεφάλαιο 4**, παρουσιάζονται οι δυνατότητες που έχει το bot που σχεδιάστηκε και αναλύεται η δομή του κώδικα.

Στο **κεφάλαιο 5**, γίνεται μία παρουσίαση της αξιολόγησης που έγινε από τους χρήστες για αυτό που δημιουργήσαμε. Αρχικά, αναλύεται το πρωτόκολλο με το οποίο πραγματοποιήθηκε η αξιολόγηση και έπειτα παρατίθενται τα αποτελέσματα αυτής.

Στο **κεφάλαιο 6**, δίνονται τα συμπεράσματα και τα επιτεύγματα της παρούσας εργασίας. Στη συνέχεια προτείνονται θέματα επέκτασης του συγκεκριμένου έργου.

Τέλος, καταγράφεται η βιβλιογραφία που χρησιμοποιήθηκε και στα παραρτήματα ενσωματώνεται ο κώδικας του προγράμματος.

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω την εταιρεία Softone που μου έδωσε τη δυνατότητα να συμμετάσχω σε μία τόσο ενδιαφέρουσα έρευνα, καθώς και τον επιβλέποντα καθηγητή μου, Αβούρη Νικόλαο και τους συνεργάτες του για τη βοήθεια που μου προσέφεραν.

Περιεχόμενα

1	Αντικείμενο και Στόχος	9
1.1	Βασικό Αντικείμενο	9
1.2	Δομή έργου	11
2	Ιστορικό και σχετική εργασία	13
2.1	Εισαγωγή	13
2.2	Τι είναι τα bots και ποια προβλήματα έρχονται να επιλύσουν	13
2.3	APIs με δυνατότητες που μπορούν να προστεθούν στα bots που έχουν δημιουργηθεί με το MBF	14
2.4	Παραδείγματα από Bots που έχουν δημιουργηθεί	16
3	Συστήματα που χρησιμοποιήθηκαν	20
3.1	Εισαγωγή	20
3.2	Microsoft Bot Framework	20
3.2.1	Το User Interface ενός Bot σχεδιασμένου με το MBF	22
3.2.2	Τύποι μηνύματος που υποστηρίζονται	23
3.3	SoftOne	25
3.3.1	Βασικές αρχές λειτουργίας	25
3.3.2	Αρχιτεκτονική SoftOne	27
3.4	Επικοινωνία συστημάτων	32
4	Παρουσίαση και λειτουργικότητα του bot	34
4.1	Εισαγωγή	34
4.2	Δυνατότητες bot	34
4.3	Σημαντικές αποφάσεις για την υλοποίηση του έργου	37
4.3.1	Microservices	37
4.3.2	Ανάλυση λέξεων	42
4.4	Λειτουργικότητα	45
5	Αξιολόγηση	49
5.1	Εισαγωγή	49
5.2	Γενικοί παράγοντες επιτυχίας ενός bot	49
5.3	Πιθανά προβλήματα σε bots	51
5.4	Γενικά χαρακτηριστικά αξιολόγησης ευχρηστίας	54
5.5	Αξιολόγηση ευχρηστίας του bot	60
5.5.1	Διαδικασία που ακολουθήθηκε	60
5.5.2	Σενάριο χρήσης	60
5.5.3	Αποτελέσματα/Ευρήματα Αξιολόγησης	62

6	Συμπεράσματα και Προοπτικές	66
6.1	Εισαγωγή	66
6.2	Συμπεράσματα και επιτεύγματα	66
6.3	Προτεινόμενα θέματα επέκτασης έργου	67
7	Βιβλιογραφία	71
A'	Κώδικας εργασίας	73

Κεφάλαιο 1

Αντικείμενο και Στόχος

1.1 Βασικό Αντικείμενο

Το βασικό αντικείμενο της παρούσας διπλωματικής εργασίας είναι η σχεδίαση και η μελέτη μίας εναλλακτικής εμπειρίας χρήσης ERP συστήματος. Δηλαδή, η διερεύνηση της ενσωμάτωσης συστήματος λεκτικής επικοινωνίας σε ένα ERP σύστημα, στο SoftOne ERP.

Πιο συγκεκριμένα, το τμήμα του παραπάνω έργου το οποίο αναφέρεται στην επιλογή και την ανάπτυξη ομάδας microservices σε γλώσσα Javascript για την εξυπηρέτηση ενός απλού συστήματος λεκτικής επικοινωνίας του SoftOne ERP μέσω Skype, με την χρησιμοποίηση του Microsoft Bot Framework.

Τι είναι ένα ERP σύστημα:

Το ακρωνύμιο ERP προέρχεται από τα αρχικά των λέξεων Enterprise Resource Planning, το οποίο μπορεί να αποδοθεί στα ελληνικά ως «πληροφοριακό σύστημα επιχειρήσεων». Αν προσπαθήσουμε να μεταφράσουμε τον όρο ERP (Enterprise Resource Planning), η μετάφρασή του είναι Προγραμματισμός Επιχειρηματικών Πόρων, η οποία δεν είναι ισοδύναμη όμως με ένα πληροφοριακό σύστημα. Ο όρος ERP πιο πολύ παραπέμπει σε μια διοικητική προσέγγιση – μεθοδολογία παρά σε ένα πληροφοριακό σύστημα. Στην ουσία αυτό είναι σωστό -αφού τα συστήματα ERP δεν είναι μόνο πληροφοριακά συστήματα- αλλά ενσωματώνουν και υλοποιούν διοικητικές πρακτικές καθώς και την οργάνωση της επιχείρησης. Υποστηρίζουν όλες τις λειτουργίες μιας επιχείρησης ή τουλάχιστον το μεγαλύτερο μέρος αυτών. Χαρακτηριστικές περιοχές λειτουργικότητας που καλύπτουν τα συστήματα ERP μεταξύ άλλων είναι: παραγωγή προϊόντων, προμήθειες υλικών, διαχείριση αποθήκης, μεταφορά προϊόντων, οικονομική διαχείριση, πωλήσεις, διαχείριση πελατών κ.λπ.

Μία από τις βασικές έννοιες των συστημάτων ERP είναι η έννοια της ολοκλήρωσης. Σε ένα ολοκληρωμένο σύστημα, οι πληροφορίες της επιχείρησης είναι αποθηκευμένες κεντρικά, σε μια κεντρική βάση δεδομένων. Η προσέγγιση αυτή επιτρέπει τη ροή της πληροφορίας σε όλα τα τμήματα μιας επιχείρησης με προφανή πλεονεκτήματα[1].

Βασικοί στόχοι της εργασίας είναι:

1. Η αύξηση της αποτελεσματικότητας διαχείρισης επιχειρησιακών πόρων μέσα από την αυτοματοποίηση κάποιων διαδικασιών.
2. Η δημιουργία νέων καναλιών επικοινωνίας τα οποία είναι πιο φιλικά προς τους χρήστες.
3. Η εξαγωγή συμπερασμάτων μέσω ανάλυσης της αλληλεπίδρασης ανθρώπου-μηχανής.

Το ζητούμενο ήταν η δημιουργία SoftOne Bot, δηλαδή μίας αυτοματοποιημένης επαφής στο Skype με την οποία θα μπορεί να συνομιλεί ο κάθε χρήστης του SoftOne και να λαμβάνει πληροφορίες που αφορούν την εταιρεία του.

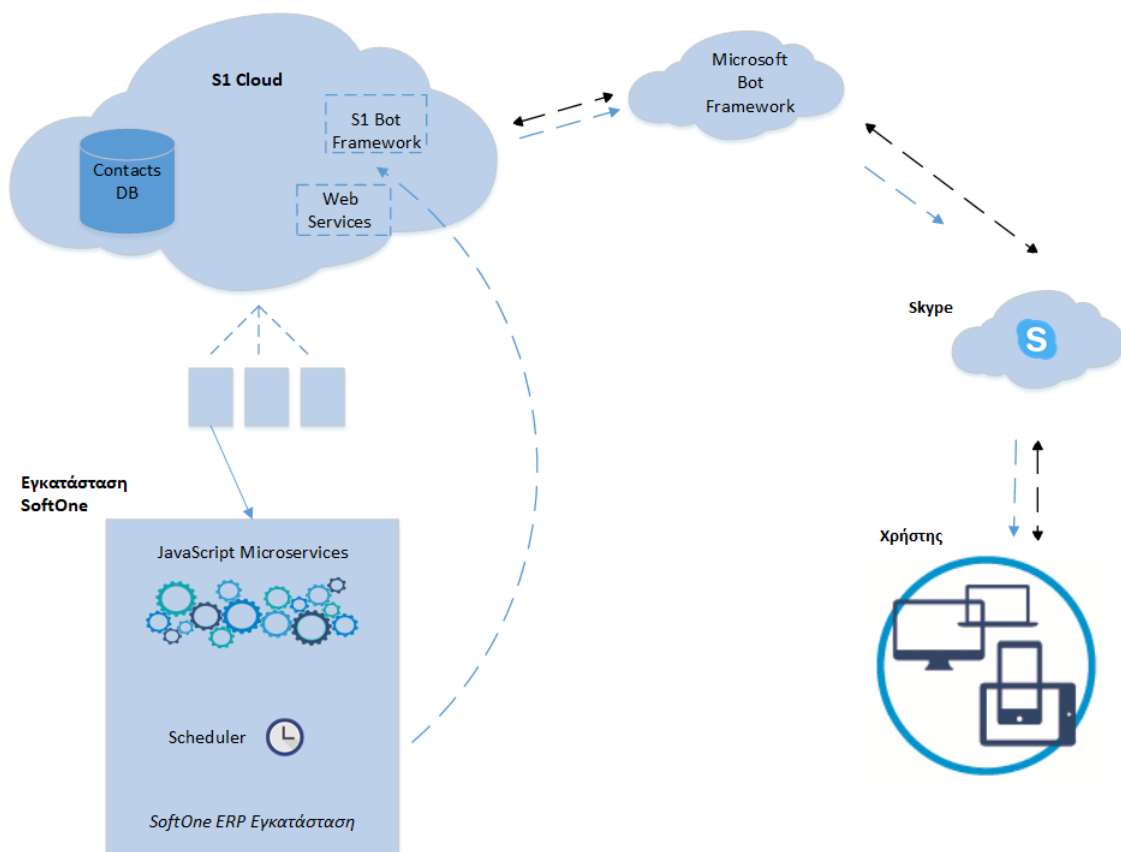
Βασικές προϋποθέσεις για την επίτευξη των παραπάνω ήταν η μελέτη και η κατανόηση του τρόπου με τον οποίο λειτουργεί ένα ERP πρόγραμμα, η μελέτη της βάσης δεδομένων στην οποία αποθηκεύονται όλα τα δεδομένα της επιχείρησης και ο εντοπισμός των απαιτήσεων των πελατών που θα είχε νόημα να μπορούν να καλυφθούν μέσω του bot.

Σε τελικό στάδιο αυτό που έπρεπε να επιτευχθεί ήταν :

- Η ταυτοποίηση κάθε ενδιαφερόμενου χρήστη Skype (μέσω του username του) και η αντιστοίχιση του με συγκεκριμένη εγκατάσταση και συγκεκριμένο χρήστη του SoftOne, έτσι ώστε να έχει πρόσβαση αποκλειστικά στις πληροφορίες που ο ίδιος επιτρέπεται να βλέπει και να διαχειρίζεται. Αυτό είναι απαραίτητο ώστε να μην χρειάζεται η κατασκευή ειδικής επαφής (bot) ανά εγκατάσταση, αλλά με την χρήση μιας μοναδικής επαφής να αποκαθίσταται η αντίστοιχη επικοινωνία. Έτσι ενισχύεται σημαντικά η δυνατότητα τυποποίησης της λύσης (δεν χρειάζεται ειδική εγκατάσταση software).
- Η δυνατότητα του bot να δίνει απάντηση σε διάφορες ερωτήσεις που αφορούν τα δεδομένα του χρήστη.
- Η δυνατότητα κάθε χρήστη να εκτελεί εργασίες μέσω του bot. Εμφανίζοντας τις εργασίες που έχει δικαίωμα να υλοποιήσει ο χρήστης ως υπερσυνδέσμους, ο τελευταίος να μπορεί να μεταβεί στο Web application της SoftOne και αν συνδεθεί με τα στοιχεία του να εκτελέσει κατευθείαν την εργασία που θέλει.
- Τέλος, η αποστολή μηνυμάτων από το bot με μορφή ειδοποιήσεων στον χρήστη, ανάλογα με τις ειδοποιήσεις που ο ίδιος θα θέσει ότι θέλει να λαμβάνει καθώς και στις ώρες και μέρες που αυτός έχει θέσει ότι θέλει να τις λαμβάνει.

1.2 Δομή έργου

Παρακάτω φαίνεται ένα διάγραμμα όλων όσων έπρεπε να επιτευχθούν ώστε να ολοκληρωθεί το συγκεκριμένο έργο. Μόνο τα τμήματα που δημιουργήθηκαν από εμένα είναι αντικείμενο της παρούσας εργασίας, όμως στο διάγραμμα παρουσιάζονται όλα τα τμήματα του συνολικού έργου με σκοπό να βοηθήσουν στην καλύτερη κατανόηση του. Για να γίνουν αντιληπτά όλα τα τμήματα, είναι σημαντικό να περιγραφούν τα στάδια από τα οποία περνάει ένα μήνυμα που αποστέλλει ένας χρήστης έως ότου δρομολογηθεί στην σωστή εγκατάσταση και ανακτήσει την πληροφορία που απαιτείται.



Σχήμα 1.1: Δημιουργήθηκε με το πρόγραμμα Visio

Στο διάγραμμα αυτό φαίνεται η "διαδρομή" που ακολουθεί το κάθε μήνυμα που στέλνει ένας χρήστης στο Skype ώστε να επιτευχθεί η ταυτοποίηση του χρήστη αυτού με την SoftOne εγκατάσταση του και έπειτα να λάβει τις απαντήσεις που επιθυμεί.

Το κάθε μήνυμα που αποστέλλεται από κάποιο χρήστη στο bot περνάει πρώτα από τον server του Skype και έπειτα μεταφέρεται μέσω του Microsoft Bot Framework στο S1 Bot Framework το οποίο στην ουσία είναι το SoftOne Bot. Σε αυτό το σημείο γίνεται η ανάλυση του μηνύματος και η δρομολόγηση του στην κατάλληλη εγκατάσταση SoftOne, που αφορά τον συγκεκριμένο χρήστη.

Όταν το μήνυμα δρομολογηθεί στη σωστή εγκατάσταση, πρέπει να γίνει αναγνώρισή του και

επιλογή του(των) microservice(s)¹ που θα εκτελεσθεί για την εξυπηρέτησή του. Το τμήμα που υλοποιήθηκε από εμένα αφορά την ανάλυση και υλοποίηση του τοπικού λογισμικού και είναι:

- Η διερεύνηση και η συγγραφή αυτών των microservices, δηλαδή η επιλογή, ανάλυση και συγγραφή σε JavaScript, των συναρτήσεων που αφορούν τις λειτουργίες, τις οποίες θα είναι ικανό το bot να εκτελέσει, αφού πρώτα έχει δρομολογηθεί στην σωστή εγκατάσταση.
- Η ανάλυση και η συγγραφή συνάρτησης η οποία επεξεργάζεται το μήνυμα που έστειλε ο χρήστης και επιλέγει την εργασία που ο τελευταίος επιθυμεί να εκτελέσει.

Η τελευταία συνάρτηση καθώς και τα microservices βρίσκονται στην εγκατάσταση του χρήστη, με σκοπό να έχουν και δυνατότητα παραμετροποίησης ανάλογα με τις ανάγκες του εκάστοτε χρήστη.

¹Σύμφωνα με την microservices αρχιτεκτονική, μια εφαρμογή δομείται ως ένα σύνολο υπηρεσιών, οι οποίες συνδέονται με "χαλαρούς" δεσμούς(με τον όρο "χαλαρούς" δεσμούς εννοούμε ότι κάθε υπηρεσία μπορεί να εκτελεστεί αυτόνομα από τις υπόλοιπες). Στην microservices αρχιτεκτονική, ο σκοπός κάθε υπηρεσίας πρέπει να είναι άρτια καθορισμένος(fine-grained) καθώς και τα πρωτόκολλα που την χαρακτηρίζουν να μην είναι αυστηρά(lightweight protocols). Το βασικό πλεονέκτημα αυτής της αρχιτεκτονικής είναι ότι κατά την αποδόμηση της εφαρμογής σε μικρότερες υπηρεσίες(services) βελτιώνεται η αυτονομία και γίνεται πιο εύκολο να κατανοηθεί, να δοκιμαστεί και να αναπτυχθεί[2].

Κεφάλαιο 2

Ιστορικό και σχετική εργασία

2.1 Εισαγωγή

Στόχος του δευτέρου κεφαλαίου είναι να γίνει παρουσίαση και ανάλυση της σχετικής εργασίας που έχει υλοποιηθεί και είναι δυνατό να φανεί χρήσιμη κατά τον σχεδιασμό ενός bot.

Αρχικά, παρουσιάζονται οι εφαρμογές που δίνουν περισσότερες δυνατότητες σε bots που έχουν δημιουργηθεί με χρήση του Microsoft Bot Framework(MBF), όπως και στη παρούσα εργασία. Έπειτα, αναλύονται ενδεικτικά κάποια άλλα bots ή και πιο εξελιγμένα AI(artificial intelligent) συστήματα που έχουν σχεδιασθεί σήμερα.

2.2 Τι είναι τα bots και ποια προβλήματα έρχονται να επιλύσουν

Ένα bot, με τη γενικότερη έννοια, είναι λογισμικό που αυτοματοποιεί την εκτέλεση εργασιών μέσω του Διαδικτύου. Πιο συγκεκριμένα, ένα bot είναι μια αυτοματοποιημένη εφαρμογή (με την μορφή συνομιλητή) που χρησιμοποιείται για την εκτέλεση απλών και επαναλαμβανόμενων εργασιών που θα ήταν χρονοβόρες ή αδύνατες για έναν άνθρωπο να εκτελέσει. Τα bots έχουν το λεγόμενο “Conversational User Interface”, ένα μέσο αλληλεπίδρασης με το λογισμικό μέσω ομιλίας, κειμένου, εικόνων, βίντεο ή άλλων μέσων.

Τα bots μπορούν να χρησιμοποιηθούν για ένα μεγάλο εύρος παραγωγικών εργασιών.

Αν σκεφτεί κάποιος την τελευταία αλληλεπίδραση που είχε με μία εφαρμογή λογισμικού, πιθανώς θα είχε χρησιμοποιήσει ένα ποντίκι, ένα πληκτρολόγιο ή μια οθόνη αφής με σκοπό να πλοηγηθεί σε μια σειρά επιλογών και να ολοκληρώσει μια εργασία. Με αυτόν τον τρόπο ο χρήστης αναγκάζεται να ακολουθήσει τους κανόνες του λογισμικού που είχαν καθοριστεί πριν από δεκαετίες. Παρατηρούμε, λοιπόν, πως το λογισμικό υπαγορεύει τους κανόνες δέσμευσης, όχι ο χρήστης. Τον τελευταίο καιρό, λόγω των bots, οι κανόνες αυτοί αλλάζουν. Τα User Interfaces που θα συναντώνται όλο και περισσότερο τα επόμενα χρόνια δεν θα είναι βασισμένα μόνο σε πολύπλοκα ιεραρχικά μενού, αλλά θα βασίζονται σε απλή συνομιλία [3].

Αναμφισβήτητα, τα chatbots μπορούν να είναι εξαιρετικά χρήσιμα σε περιοχές που οι άνθρωποι είναι καταδικασμένοι να αποτύχουν. Κάποιες από αυτές είναι :

- Μπορούν να περιμένουν επ’ αόριστον τους συνομιλητές τους να ανταποκριθούν.
- Είναι σε θέση να θυμούνται επακριβώς κάθε συζήτηση που είχαν με όλους τους συνομιλητές τους.

- Μπορούν να έχουν πρόσβαση σε ορισμένες πληροφορίες αμέσως.
- Δεν κουράζονται ποτέ.

Αυτά είναι μόνο κάποια από τα ελκυστικά χαρακτηριστικά των σύγχρονων chatbots. Αλλά το δυναμικό των bots είναι αρκετά μεγάλο. Αν οι ερευνητές και οι επιχειρήσεις συνεχίσουν να ενσωματώνουν τέτοιες τεχνολογίες, υπάρχει σημαντική πιθανότητα ότι τα bots θα επηρεάσουν σημαντικά την online ζωή των ανθρώπων και τον τρόπο που εκτελούν τη δουλειά τους, στο άμεσο μέλλον.

Ασφαλώς η χρήση των bots δεν είναι πανάκεια. Πολλές φορές το είδος της πληροφορίας ή ο τρόπος απεικόνισής της εξυπηρετούνται καλύτερα από συμβατικά user interfaces, αλλά σε κάθε περίπτωση ένα σημαντικό μερίδιο στην επικοινωνία ανθρώπου μηχανής στο μέλλον, θα καλύπτεται από τα bots.

2.3 APIs με δυνατότητες που μπορούν να προστεθούν στα bots που έχουν δημιουργηθεί με το MBF

Για την δημιουργία του bot χρησιμοποιήθηκε το Microsoft Bot Framework (MBF), το οποίο θα παρουσιαστεί αναλυτικά σε επόμενο κεφάλαιο. Παρακάτω θα παρουσιαστούν κάποια APIs με διάφορες δυνατότητες, τα οποία μπορούν να προστεθούν στα bots που έχουν δημιουργηθεί με το MBF ώστε να βελτιστοποιήσουν κάποιες συχνές διαδικασίες.

Η Microsoft παρέχει μεγάλη ποικιλία από AI (artificial intelligence) αλγόριθμους, οι οποίοι έχουν σχεδιαστεί από ειδικούς σε θέματα επεξεργασίας φυσικής γλώσσας, εξαγωγής πληροφοριών και αναζήτησης στο διαδίκτυο. Οι αλγόριθμοι αυτοί απλοποιούν μία μεγάλη ποσότητα από συνηθισμένες AI εργασίες και δίνουν την δυνατότητα στον κάθε δημιουργό να προσθέσει “state-of-the-art intelligence” τεχνολογίες στα bots του με λίγες γραμμές κώδικα. Τα APIs αυτά έχουν τη δυνατότητα να ενσωματωθούν στις περισσότερες μοντέρνες γλώσσες προγραμματισμού. Συνεχώς βελτιώνονται, μαθαίνουν και αυξάνουν τις δυνατότητες τους ώστε να ανταποκρίνονται στα νέα τεχνολογικά δεδομένα. Τα APIs αυτά χωρίζονται σε κάποιες κατηγορίες, ανάλογα με τις λύσεις που προσφέρουν.

Κατανόηση γλώσσας :

Η αλληλεπίδραση ανάμεσα στους χρήστες και τα bots έχει στόχο να είναι πιο ελεύθερη. Για τον λόγο αυτό τα bots πρέπει να έχουν την ικανότητα να καταλαβαίνουν φυσική γλώσσα και να καταλήγουν σε κάποιο νόημα από τα συμφραζόμενα. Τα APIs προσφέρουν ισχυρά γλωσσικά μοντέλα που αντιλαμβάνονται τις ανάγκες των χρηστών, αναγνωρίζουν δομή σε προτάσεις που τους δίνονται και επιτρέπουν στα bots να απαντούν κατάλληλα. Οι πέντε εφαρμογές που δίνονται παρακάτω υποστηρίζουν διάφορες δυνατότητες ανάλυσης κειμένου, όπως ορθογραφικό έλεγχο, ανίχνευση συναισθήματος, μοντελοποίηση γλωσσών, και εξαγωγή γνώσεων από ένα κείμενο.[4]

- Language Understanding Intelligent Service (LUIS), που μπορεί να επεξεργάζεται φυσική γλώσσα χρησιμοποιώντας προκατασκευασμένα ή ειδικά εκπαιδευμένα γλωσσικά μοντέλα.
- Text Analytics API που ανιχνεύει συναίσθημα, φράσεις κλειδιά, θεματολογίες και γλώσσα από κείμενα που του δίνονται
- Bing Spell Check API που υποστηρίζει ισχυρές δυνατότητες ορθογραφικού ελέγχου και είναι ικανό να αναγνωρίζει διαφορές μεταξύ ονομάτων και εμπορικών επωνυμιών.

- Linguistic Analysis API που χρησιμοποιεί προχωρημένους αλγορίθμους γλωσσικής ανάλυσης ώστε να επεξεργαστεί το λαμβανόμενο κείμενο και να εκτελεί διαδικασίες όπως ανάλυση της δομής κειμένου.
- Web Language Model API που μπορεί να χρησιμοποιηθεί για να αυτοματοποιήσει μία ποικιλία από εργασίες επεξεργασίας φυσικής γλώσσας, όπως η συχνότητα στην οποία εμφανίζονται οι λέξεις ή η πρόβλεψη λέξεων που έπονται ή προηγούνται άλλων, χρησιμοποιώντας προχωρημένους αλγορίθμους.

Εξαγωγή πληροφοριών :

Παρέχονται πέντε APIs που επιτρέπουν την αναγνώριση ονομαστικών οντοτήτων ή φράσεων σε αδόμητο κείμενο, την πρόσθεση εξατομικευμένων προτάσεων, την παροχή προτάσεων αυτόματης συμπλήρωσης με βάση τη φυσική ερμηνεία των ερωτημάτων των χρηστών και την αναζήτηση ακαδημαϊκών εγγράφων και άλλων ερευνών. Τα APIs αυτά παρουσιάζονται παρακάτω.[4]

- Entity Linking Intelligence Service που επεξηγεί/αναλύει μη δομημένο κείμενο ανάλογα με τις σχετικές οντότητες που αναφέρονται σε αυτό. Ανάλογα με το περιεχόμενο, η ίδια λέξη μπορεί να αναφέρεται σε διαφορετικά πράγματα. Η υπηρεσία αυτή καταλαβαίνει το περιεχόμενο του λαμβανόμενου κειμένου και αναγνωρίζει κάθε ξεχωριστή οντότητα σε αυτό.
- Το Recommendations API το οποίο παρέχει προτάσεις για προϊόντα που συνήθως αγοράζονται μαζί ή και προσωπικές προτάσεις που βασίζονται στο ιστορικό του κάθε χρήστη.
- Το Knowledge Exploration Service το οποίο παρέχει απόδοση των αναζητήσεων των χρηστών σε φυσική γλώσσα και επιστρέφει επεξηγηματικές μεταφράσεις για να επιτρέψει εμπλουτισμένες αναζητήσεις που αντιλαμβάνονται τι πληκτρολογεί ο χρήστης. Οι προτάσεις για άμεση συμπλήρωση και οι βελτιώσεις στη πρόβλεψη των αναζητήσεων βασίζονται στα δεδομένα του συγκεκριμένου χρήστη ώστε να επιτρέψει στον χρήστη να πραγματοποιήσει γρήγορες αναζητήσεις.
- Το API Academic Knowledge το οποίο επιστρέφει ακαδημαϊκά ερευνητικά έγγραφα, συγγραφείς, περιοδικά, συνέδρια και πανεπιστήμια από το Microsoft Academic Graph. Παρέχει μια βάση γνώσεων χρησιμοποιώντας ένα διάλογο με γραφικά με δυνατότητες αναζήτησης σε εκατοντάδες εκατομμύρια ερευνητικές οντότητες.
- Το QnA Maker το οποίο είναι ένα API που εκπαιδεύει AI ώστε να ανταποκρίνεται στις ερωτήσεις των χρηστών με έναν πιο φυσικό τρόπο. Με τελειοποιημένη machine learning λογική μετατρέπει δεδομένα όπως ζεύγη ερωτήσεων-απαντήσεων σε ξεχωριστές απαντήσεις.

Αναγνώριση ομιλίας :

Παρέχονται τρία Speech APIs που επιτρέπουν την παραγωγή ή την σύνθεση ομιλίας.[4]

- Speech API το οποίο παρέχει δυνατότητες μετατροπής λόγου σε κείμενο και κειμένου σε λόγο.
- Custom Recognition Intelligent Service (CRIS) το οποίο επιτρέπει στον χρήστη να δημιουργεί μοντέλα αναγνώρισης ομιλίας ώστε να προσαρμόζει την μετατροπή ομιλίας σε κείμενο στο λεξιλόγιο μίας εφαρμογής ή στον τρόπο με τον οποίο μιλάει ο χρήστης.

- Speaker Recognition API που επιτρέπει την ταυτοποίηση και την εξακρίβωση του χρήστη μέσω της φωνής.

Αναζήτηση στο Διαδίκτυο :

Τα Bing Search APIs επιτρέπουν την προσθήκη δυνατότητας αναζήτησης στο διαδίκτυο στα bots. Με λίγες γραμμές κώδικα είναι εφικτή η πρόσβαση σε εκατομμύρια σελίδες, φωτογραφίες, βίντεο και άλλους τύπους αποτελεσμάτων. Παρακάτω παρουσιάζονται πέντε Bing Search APIs.[4]

- Web Search API.
- Image Search API.
- Video Search API.
- News Search API.
- Autosuggest API, το οποίο προσφέρει προτάσεις για αυτόματη συμπλήρωση των αναζητήσεων του χρήστη ώστε να ολοκληρώνει τις αναζητήσεις του πιο γρήγορα και με λιγότερη πληκτρολόγηση.

Κατανόηση εικόνας και βίντεο :

Τα API Vision φέρνουν προηγμένες δεξιότητες κατανόησης εικόνας και βίντεο στα bots. Οι τελευταίοι αλγόριθμοι επιτρέπουν την επεξεργασία εικόνων ή βίντεο και τη λήψη πληροφοριών που μπορούν να μετατραπούν σε ενέργειες. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για την αναγνώριση αντικειμένων, ανθρώπινων προσώπων, ηλικίας, φύλου ή ακόμα και συναισθήματος. Τα API Vision υποστηρίζουν διάφορες δυνατότητες κατανόησης της εικόνας. Μπορούν να προσδιορίσουν το ώριμο ή σαφές περιεχόμενο, τις εκτιμήσεις και τα χρώματα έμφασης, να ταξινομήσουν το περιεχόμενο των εικόνων, να κάνουν οπτική αναγνώριση χαρακτήρων και να περιγράψουν μια εικόνα με πλήρεις προτάσεις της αγγλικής γλώσσας. Υποστηρίζουν επίσης πολλές δυνατότητες επεξεργασίας εικόνων και βίντεο, όπως έξυπνες δημιουργίες μικρογραφιών εικόνας ή βίντεο ή σταθεροποίηση της εξόδου ενός βίντεο.

Παρέχονται τέσσερα APIs τα οποία επιτρέπουν επεξεργασία εικόνας και βίντεο.[4]

- Το Computer Vision API το οποίο εξάγει πληροφορίες σχετικά με εικόνες (όπως αντικείμενα ή άτομα), καθορίζει εάν η εικόνα περιέχει ώριμο ή ρητό περιεχόμενο και επεξεργάζεται κείμενο στις εικόνες.
- Το Emotion API που αναλύει τα ανθρώπινα πρόσωπα και αναγνωρίζει το συναίσθημά τους ανάμεσα σε οκτώ πιθανές κατηγορίες ανθρώπινων συναισθημάτων.
- Το API Face που ανιχνεύει τα ανθρώπινα πρόσωπα, τα συγκρίνει με παρόμοια πρόσωπα και μπορεί να κατηγοριοποιήσει τους ανθρώπους σε ομάδες ανάλογα με την οπτική ομοιότητα.
- Το Video API το οποίο αναλύει και επεξεργάζεται βίντεο, ανιχνεύει κίνηση και μπορεί να δημιουργήσει μια περίληψη μικρογραφιών κίνησης του βίντεο.

2.4 Παραδείγματα από Bots που έχουν δημιουργηθεί

Τον τελευταίο καιρό τα bots γίνονται όλο και περισσότερο μέρος της καθημερινότητας των ανθρώπων, όπως συνέβαινε παλιότερα με τα applications, και συνεχώς δημιουργούνται νέα που

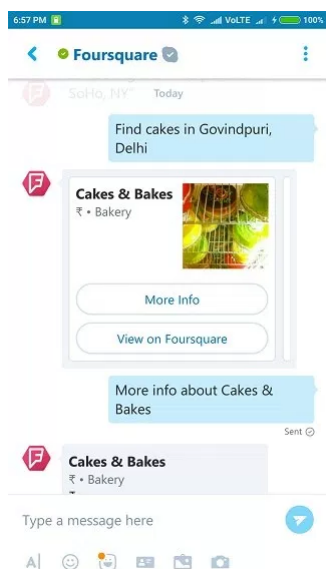
καλύπτουν διαφορετικές ανάγκες. Παρακάτω θα παρουσιαστούν ενδεικτικά κάποια από αυτά, των οποίων το κανάλι επικοινωνίας είναι το Skype.[5]

Bing Music Preview : Είναι ένα bot στο Skype που δίνει τη δυνατότητα στους χρήστες να αναζητήσουν μουσική σε δημοφιλείς ιστοτόπους όπως το YouTube, το Vevo, το LyricFind και πολλά άλλα. Ο χρήστης πρέπει απλά να πληκτρολογήσει το όνομα του τραγουδιού ή του καλλιτέχνη και το bot θα του επιστρέψει τον σύνδεσμο του βίντεο από οποιονδήποτε από τους προαναφερόμενους ιστοτόπους. Είναι δυνατή και η αναζήτηση διαφημίσεων και βίντεο άλλων ειδών εκτός των μουσικών βίντεο.

Foursquare Bot : Το Foursquare Bot το οποίο παρέχει τη δυνατότητα στον χρήστη να αναζητήσει το καλύτερο φαγητό, τη νυχτερινή ζωή και τα αξιοθέατα που βρίσκονται κοντά του ή οπουδήποτε αλλού στον κόσμο. Ζητά από τον χρήστη να εισάγει τις λεπτομέρειες αυτού που θέλει να βρει, ακολουθούμενο από το όνομα της περιοχής στην οποία αναζητεί. Για παράδειγμα, μια αναζήτηση θα μπορούσε να είναι : “Εύρεση καφέ στο Connaught Place, Νέο Δελχί”. Τέλος, ο χρήστης έχει τη δυνατότητα να επιλέξει να δει τις τοποθεσίες σε χάρτη.

Solitaire Bot : Είναι ένα bot το οποίο παρέχει μια εμπειρία αναπαραγωγής του Solitaire (κλασσικής πασιέντζας) σε μορφή συνομιλίας. Το bot λειτουργεί ως υπολογιστής και στη συνέχεια παρέχει στον χρήστη τις πιθανές κινήσεις που μπορεί να επιλέξει. Εμφανίζει επίσης εικόνες της τρέχουσας κατάστασης και προτεινόμενες κινήσεις.

Hello Hipmunk Bot : Είναι ένα bot που προσφέρει ταξιδιωτικές συμβουλές και προτάσεις. Βασικό χαρακτηριστικό του είναι ότι αναζητεί σε εκατοντάδες sites ώστε να εξυπηρετήσει τον χρήστη. Έχει τη δυνατότητα να πραγματοποιήσει κρατήσεις σε ξενοδοχεία και σε πτήσεις και χαρακτηρίζεται από ένα καλό επίπεδο κατανόησης και ομιλίας.



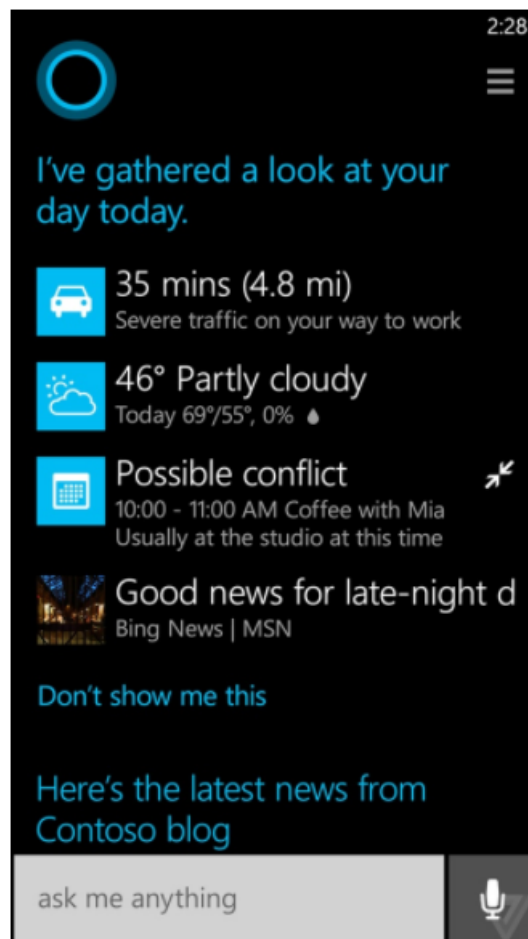
Σχήμα 2.1: Παράδειγμα Foursquare Bot



Σχήμα 2.2: Παράδειγμα Solitaire Bot

Βέβαια, υπάρχουν και πιο γνωστά και αναπτυγμένα AI συστήματα όπως είναι η Siri της Apple και η Cortana της Microsoft. Τα δύο αυτά συστήματα θεωρούνται «προσωπικοί ψηφιακοί βοηθοί», δηλαδή μπορούν να εκτελέσουν τις περισσότερες από τις διαδικασίες που υποστηρίζει η συσκευή ή και να συμβουλευθούν τον χρήστη, με την Siri να βρίσκεται σε συσκευές που

χρησιμοποιούν iOS και την Cortana σε συσκευές Windows. Έχοντας πρόσβαση σε πληροφορίες από άλλες εφαρμογές (όπου αυτή έχει εγκριθεί) και δημιουργώντας ένα εικονικό "Notebook" το οποίο αποθηκεύει προσωπικές πληροφορίες, η Cortana δημιουργεί λίστα με όλα όσα γνωρίζει για τον κάθε χρήστη. Ο χρήστης έχει την δυνατότητα να προσθέσει προσωπικές πληροφορίες και χειροκίνητα. Είναι πραγματικά ένας κόμβος πληροφοριών που μετατρέπεται σε κάρτες και τμήματα του μπορούν να χρησιμοποιηθούν για την παραγωγή ειδοποιήσεων. Εάν, για παράδειγμα, η αγαπημένη ομάδα ποδοσφαίρου του χρήστη σκοράρει, η Cortana μπορεί να τον ειδοποιήσει. Ένα από τα πιο χρήσιμα χαρακτηριστικά της Cortana είναι η ικανότητά της να ενεργοποιεί δράσεις βασισμένες σε γεγονότα. Για παράδειγμα, λέγοντας "Υπενθύμισε μου την επόμενη φορά που θα καλέσω τη σύζυγο μου ότι πρέπει να μιλήσουμε για το παιδί", θα δημιουργήσει μια υπενθύμιση που θα ενεργοποιηθεί όταν ο χρήστης καλέσει την σύζυγο του ή όταν εκείνη τον καλέσει. Η Cortana βασίζεται στις υπηρεσίες υποστήριξης του Bing για την πλειοψηφία των χαρακτηριστικών της και υποστηρίζεται από χιλιάδες διακομιστές (servers)[6].

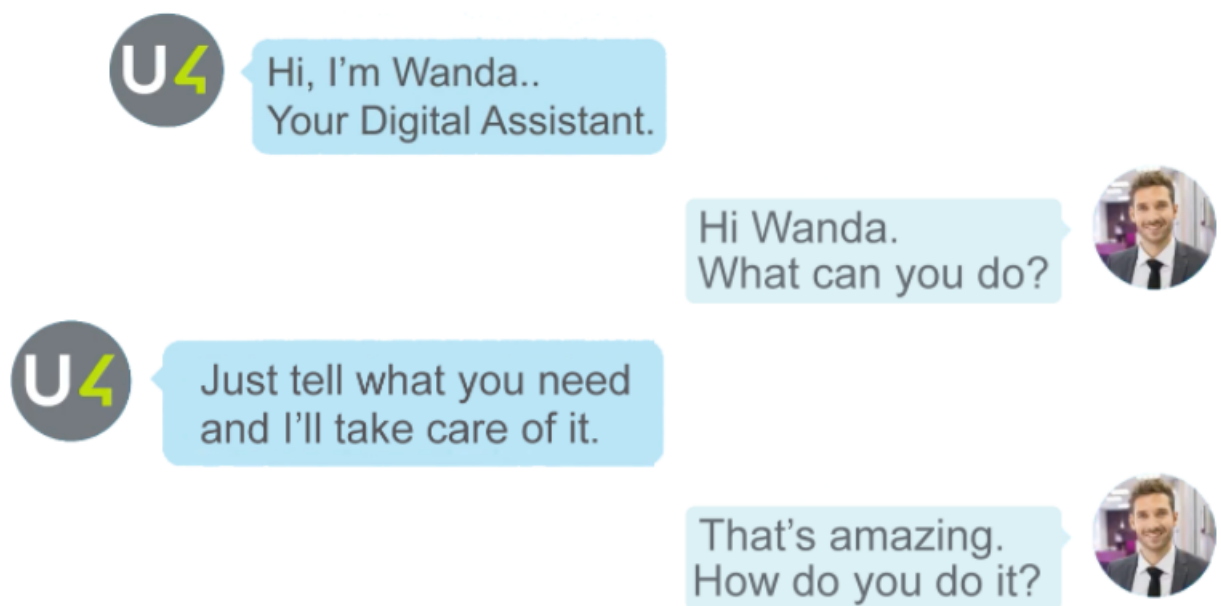


Σχήμα 2.3: Cortana[6]

Τέλος, έχει γίνει ήδη και μία παρόμοια προσπάθεια δημιουργίας «ψηφιακού επιχειρησιακού βοηθού» (bot που επικοινωνεί με ERP σύστημα) από την εταιρεία Unit4. Για να εξασφαλίσει μια φυσική εμπειρία χρήστη, η Unit4 έχει χτίσει την Wanda που ενσωματώνεται στο επιλεγμένο από τον χρήστη περιβάλλον, Skype, Slack και Facebook Messenger. Η Wanda παρέχει μια απλή και φιλική πύλη δεδομένων στα στοιχεία του ERP συστήματος, ώστε να βοηθήσει τους χρήστες να ολοκληρώσουν γρήγορα τα καθήκοντά τους. Οι χρήστες μπορούν να επικοινωνούν και να αλληλεπιδρούν με τα επιχειρησιακά τους συστήματα μέσω της συνομιλίας. Η Unit4 χρησιμοποιεί το API Language Understanding Intelligent Service (LUIS) της Microsoft για να δώσει στην Wanda την ικανότητα να κατανοεί τι θέλει ένας χρήστης με βάση τη γλώσσα που χρησιμοποιεί[7].

Η Wanda προσφέρει πέντε βοηθούς :

- Τον βοηθό προσωπικού ο οποίος υποστηρίζει απλή συμπλήρωση των καθηκόντων αυτοεξυπηρέτησης των εργαζομένων που σχετίζονται με το HR (human resources), όπως η αποστολή αιτημάτων απουσίας και η διερεύνηση των ισορροπιών των διακοπών και των αμοιβών
- Τον βοηθό αγοράς, για την εύρεση προϊόντων και προμηθευτών, τη δημιουργία απαιτήσεων και τη διαχείριση εγκρίσεων.
- Τον βοηθό ώρας ο οποίος δημιουργεί αυτόματα χρονοδιαγράμματα βασισμένα σε πολλαπλές ροές δεδομένων και δεδομένα θέσης GPS και φάρων για την παρακολούθηση του χρόνου.
- Τον Ταξιδιωτικό Βοηθό που δημιουργεί ταξιδιωτικά αιτήματα και διαχειρίζεται εγκρίσεις βάσει προτύπων και προτιμήσεων ταξιδιού.
- Τέλος τον βοηθό έγκρισης ο οποίος ειδοποιεί και υπενθυμίζει στα διευθυντικά στελέχη την έγκριση των καθηκόντων και επισημαίνει σημαντικά καθήκοντα όπου επικρατούν προθεσμίες.



Σχήμα 2.4: Wanda[7]

Κεφάλαιο 3

Συστήματα που χρησιμοποιήθηκαν

3.1 Εισαγωγή

Στο παρόν κεφάλαιο θα παρουσιασθούν και θα αναλυθούν τα βασικά συστήματα που χρησιμοποιήθηκαν για το συγκεκριμένο έργο καθώς και οι μεταβολές που πραγματοποιήθηκαν σε αυτά κατά την υλοποίησή του.

Τα συστήματα που χρησιμοποιήθηκαν είναι το Microsoft Bot Framework και το SoftOne ERP, με βασικό κανάλι επικοινωνίας μεταξύ bot και ERP να είναι το Skype.

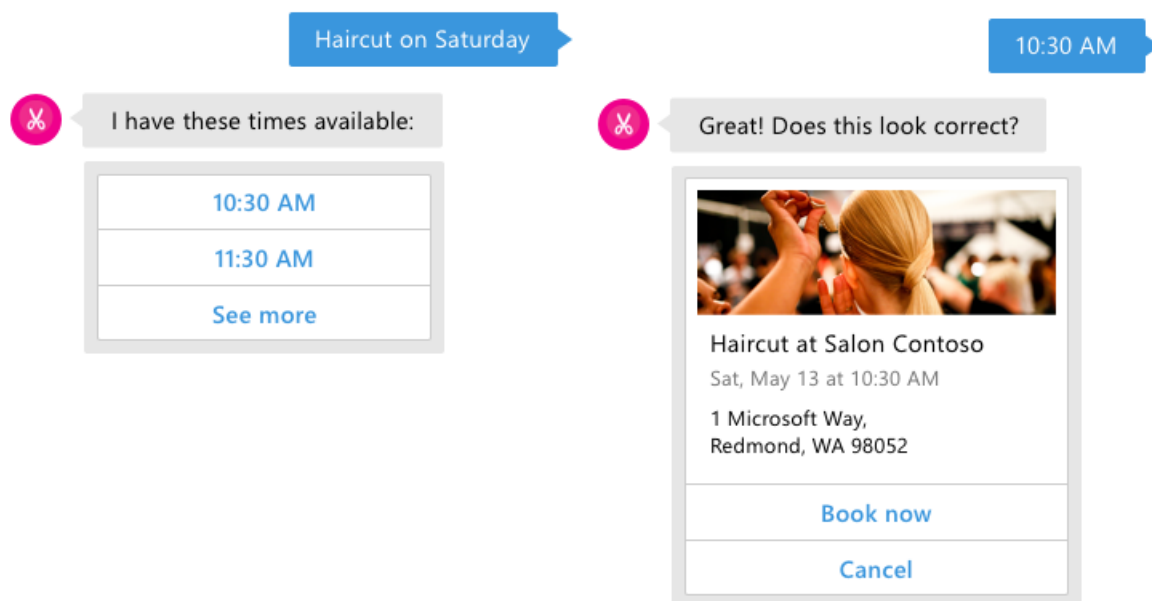
3.2 Microsoft Bot Framework

Το Bot Framework είναι μια πλατφόρμα για την κατασκευή, τη σύνδεση, τη δοκιμή και την ανάπτυξη ισχυρών και ευφυών bots. Με την υποστήριξη των .NET, Node.js και REST, είναι δυνατό να αποκτήσει κάποιος το Bot Builder SDK και να ξεκινήσει την κατασκευή με το Bot Framework.

Τι είναι όμως ένα bot;

Ο καλύτερος τρόπος να σκεφτεί κάποιος ένα bot είναι ως μια εφαρμογή με την οποία οι χρήστες αλληλεπιδρούν με ένα συνομιλητικό τρόπο. Τα bots μπορούν να επικοινωνήσουν με τους χρήστες με κείμενο, κάρτες ή ομιλία. Ένα bot μπορεί να είναι πολύ απλό και να δίνει βασικές απαντήσεις ή μπορεί να είναι μια εξελιγμένη έκφανση των τεχνικών τεχνητής νοημοσύνης.

Ο λόγος που επιλέχθηκε το Bot Framework είναι ότι δίνει τη δυνατότητα στον σχεδιαστή να δημιουργήσει bots που υποστηρίζουν πολλούς διαφορετικούς τύπους αλληλεπιδράσεων με τους χρήστες και δίνει τη δυνατότητα σύνδεσης του bot με πολλά κανάλια επικοινωνίας. Είναι στην ουσία, ένας διαμεσολαβητής μεταξύ του καναλιού και του bot, ο οποίος μετατρέπει τα μηνύματα που στέλνει ο χρήστης σε μία συγκεκριμένη μορφή, ανεξάρτητα από το κανάλι που προέρχονται, ώστε να μπορεί ο σχεδιαστής bot να τα επεξεργαστεί με τον ίδιο τρόπο ασχέτως από την προέλευσή τους. Στην συζήτηση μπορεί να περιλαμβάνονται απλές συμβολοσειρές κειμένου ή πιο πολύπλοκες πλούσιες κάρτες που περιέχουν κείμενο, εικόνες και κουμπιά δράσης. Ακόμη μπορούν να προστεθούν φυσικές αλληλεπιδράσεις γλώσσας, οι οποίες επιτρέπουν στους χρήστες να αλληλεπιδρούν με τα bots με φυσικό και εκφραστικό τρόπο[8].



Σχήμα 3.1: Παράδειγμα χρήσης καρτών για την επικοινωνία με τον χρήστη

Οι σχεδιαστές bots αντιμετωπίζουν τα ίδια προβλήματα: τα bots απαιτούν βασικό I/O, πρέπει να έχουν γλωσσικές και διαλογικές ικανότητες και πρέπει να συνδεθούν με τους χρήστες, κατά προτίμηση σε οποιαδήποτε εμπειρία συνομιλίας και γλώσσας που ο χρήστης επιλέγει. Το Bot Framework παρέχει ισχυρά εργαλεία και χαρακτηριστικά που βοηθούν στην επίλυση αυτών των προβλημάτων.

Για να βοηθήσει τους δημιουργούς, το Bot Framework περιλαμβάνει :

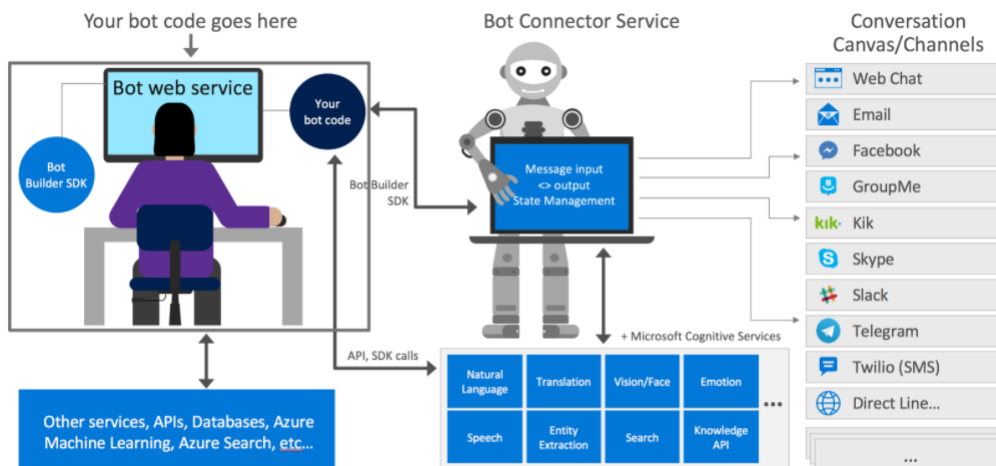
- Το Bot Builder το οποίο παρέχει πλούσιο και πλήρως εξοπλισμένο SDK (Software Development Kit) για τις πλατφόρμες .NET και Node.js. Το SDK παρέχει λειτουργίες που καθιστούν πολύ πιο απλές τις αλληλεπιδράσεις μεταξύ των bots και των χρηστών.
- Το Bot Builder περιλαμβάνει επίσης έναν εξομοιωτή για τον εντοπισμό σφαλμάτων στα bots καθώς και ένα μεγάλο σύνολο ρομποτικών δειγμάτων για να χρησιμοποιηθούν ως δομικά στοιχεία.
- Το Bot Framework portal που καθιστά πιο εύκολη τη διαχείριση ενός bot. Είναι ένα εύχρηστο περιβάλλον για να εγγραφεί, να συνδεθεί και να διαχειρίζεται τα bots του ο εκάστοτε σχεδιαστής.

Το Bot Framework υποστηρίζει αρκετά δημοφιλή κανάλια για τη σύνδεση των bots. Οι χρήστες μπορούν να ξεκινήσουν συνομιλία με κάποιο bot σε οποιοδήποτε κανάλι αυτό έχει ρυθμιστεί να συνεργάζεται, όπως email, Facebook, Skype και SMS.

Παρακάτω παρουσιάζονται οι βασικές έννοιες του Bot Framework[9].

- Κανάλι, το οποίο είναι η σύνδεση μεταξύ του Bot Framework και εφαρμογών επικοινωνίας όπως το Skype, το Slack, το Facebook Messenger, το Office 365 mail και άλλα.
- Bot Connector, που είναι η υπηρεσία η οποία συνδέει ένα bot με ένα ή περισσότερα κανάλια και χειρίζεται την ανταλλαγή μηνυμάτων μεταξύ τους. Αυτή η υπηρεσία σύνδεσης επιτρέπει στο bot να επικοινωνεί μέσω πολλών καναλιών χωρίς να σχεδιάζει χειροκίνητα ένα συγκεκριμένο μήνυμα για το “σχήμα” του καναλιού.

- **Activity/ Δραστηριότητα.** Ο Bot Connector χρησιμοποιεί ένα activity object για την ανταλλαγή πληροφοριών μεταξύ bot και καναλιού. Κάθε επικοινωνία είναι μια δραστηριότητα κάποιου τύπου. Ορισμένες δραστηριότητες είναι αόρατες για το χρήστη, όπως η ειδοποίηση ότι το bot έχει προστεθεί στη λίστα επαφών ενός χρήστη.
- **Μήνυμα,** το οποίο είναι ο πιο συνηθισμένος τύπος δραστηριότητας. Ένα μήνυμα μπορεί να είναι τόσο απλό όσο μια συμβολοσειρά κειμένου ή να περιέχει συνημμένα, διαδραστικά στοιχεία και πλούσιες κάρτες. Για παράδειγμα, η προσθήκη ενός bot στη λίστα επαφών του χρήστη θα μπορούσε να ενεργοποιήσει το bot να απαντήσει με ένα μήνυμα που περιέχει μια συμβολοσειρά λέγοντας "Thank you!" και μια εικόνα ενός ευτυχισμένου προσώπου.
- **Παράθυρα Διαλόγου (Dialog).** Τα παράθυρα διαλόγου βοηθούν στην οργάνωση της λογικής στο bot και στη διαχείριση της ροής συνομιλιών. Τα παράθυρα διαλόγου είναι διατεταγμένα σε μια στοίβα και το επάνω τμήμα διαλόγου στη στοίβα επεξεργάζεται όλα τα εισερχόμενα μηνύματα έως ότου κλείσει ή ανοίξει διαφορετικό παράθυρο διαλόγου.
- **Πλούσιες Κάρτες ή Rich cards.** Μία κάρτα μπορεί να περιλαμβάνει τίτλο, περιγραφή, σύνδεσμο και εικόνες.

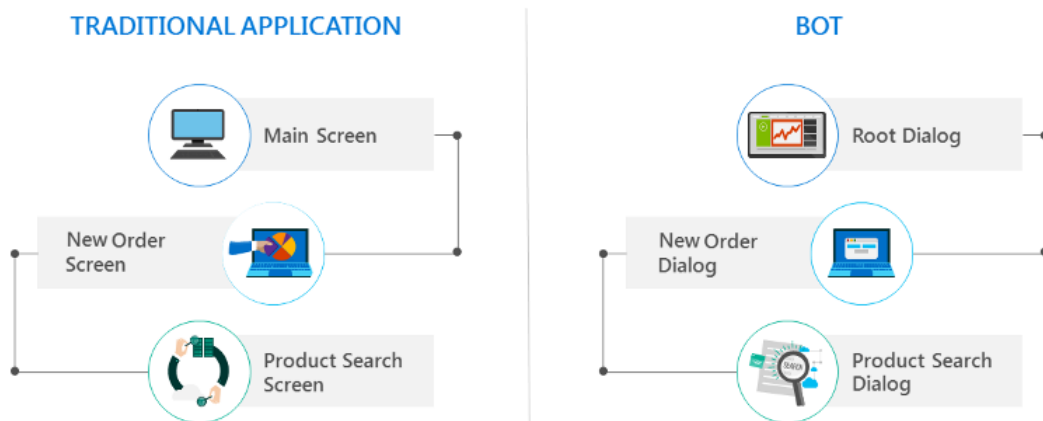


Σχήμα 3.2: Βασικές έννοιες MBF

3.2.1 Το User Interface ενός Bot σχεδιασμένου με το MBF

Σε μια παραδοσιακή εφαρμογή, η διεπαφή χρήστη (User Interface) είναι μια σειρά οθονών. Μια ενιαία εφαρμογή ή ιστότοπος μπορεί να χρησιμοποιήσει μία ή περισσότερες οθόνες, όπως απαιτείται για την ανταλλαγή πληροφοριών με τον χρήστη. Οι περισσότερες εφαρμογές ξεκινούν με μια κύρια οθόνη και μέσω αυτής οι χρήστες μεταφέρονται σε άλλες οθόνες για διάφορες λειτουργίες. Όπως οι εφαρμογές και οι ιστότοποι, τα bots έχουν ένα περιβάλλον χρήστη αλλά αποτελούνται από διαλόγους και όχι από οθόνες.

Τα παράθυρα διαλόγου δίνουν τη δυνατότητα στον προγραμματιστή bot να διαχωρίσει λογικά τις διάφορες λειτουργίες bot και να καθοδηγήσει τη ροή των συνομιλιών. Τα παράθυρα διαλόγου μπορούν να έχουν graphical interfaces. Μπορεί να περιέχουν κουμπιά, κείμενο και άλλα στοιχεία ή να βασίζονται αποκλειστικά σε λόγια. Τα παράθυρα διαλόγου περιέχουν επίσης ενέργειες για την εκτέλεση εργασιών, όπως η κλήση άλλων παραθύρων διαλόγου ή η επεξεργασία των δεδομένων από τους χρήστες.[10]



Σχήμα 3.3: Ροή οθόνης μιας παραδοσιακής εφαρμογής σε σύγκριση με τη ροή διαλόγου ενός bot[10].

Σε μια παραδοσιακή εφαρμογή όλα αρχίζουν με την κύρια οθόνη. Η κύρια οθόνη επικαλείται την επόμενη οθόνη. Η νέα οθόνη παραμένει υπό έλεγχο μέχρι να κλείσει ή να επικαλεστεί άλλες οθόνες. Εάν κλείσει η νέα οθόνη ο χρήστης επιστρέφει στην κύρια οθόνη. Σε ένα bot, όλα ξεκινούν με το βασικό παράθυρο διαλόγου το οποίο επικαλείται τα νέα παράθυρα. Σε αυτό το σημείο, το νέο παράθυρο παίρνει τον έλεγχο της συνομιλίας και παραμένει υπό έλεγχο μέχρι να κλείσει ή να επικαλεστεί άλλους διάλογους. Εάν το νέο παράθυρο διαλόγου τερματιστεί, ο έλεγχος της συνομιλίας επιστρέφει ξανά στο κύριο παράθυρο διαλόγου.

Όταν ένα παράθυρο διαλόγου επικαλείται ένα άλλο, ο Bot Builder προσθέτει το νέο παράθυρο διαλόγου στο επάνω μέρος της στοίβας διαλόγου. Ο διάλογος που βρίσκεται στην κορυφή της στοίβας ελέγχει τη συνομιλία. Κάθε νέο μήνυμα που αποστέλλεται από το χρήστη θα υπόκειται σε επεξεργασία από αυτό το παράθυρο διαλόγου μέχρι να κλείσει ή να επικαλεστεί άλλο παράθυρο διαλόγου. Όταν ένα παράθυρο διαλόγου κλείνει, αφαιρείται από τη στοίβα και ο προηγούμενος διάλογος στη στοίβα παίρνει τον έλεγχο της συνομιλίας[10].

3.2.2 Τύποι μηνύματος που υποστηρίζονται

Το bot στέλνει δραστηριότητες τύπου μηνύματος για να μεταδώσει πληροφορίες στους χρήστες και λαμβάνει επίσης δραστηριότητες μηνυμάτων από τους χρήστες. Ορισμένα μηνύματα που στέλνει ένα bot μπορεί να αποτελούνται από απλό κείμενο, ενώ άλλα μπορεί να περιέχουν πλουσιότερο περιεχόμενο, όπως πλούσιες κάρτες (rich cards) και δεδομένα ειδικά για το κανάλι. Το κείμενο του μηνύματος μπορεί να μορφοποιηθεί χρησιμοποιώντας plain, markdown ή xml. Το επίπεδο υποστήριξης μορφής κειμένου ποικίλλει μεταξύ των καναλιών.

Τα είδη μορφοποίησης κειμένου που υποστηρίζει το Skype, το οποίο είναι το κανάλι στο οποίο συνδέθηκε το SoftOne bot, είναι έντονη γραφή, πλάγια γραφή, υπογραμμισμένο κείμενο, κείμενο σε μορφή υπερσυνδέσμου και φωτογραφίες. Ακόμη υποστηρίζει rich media cards για οπτική αλληλεπίδραση. Οι κάρτες που υποστηρίζονται από το Skype είναι[11]:

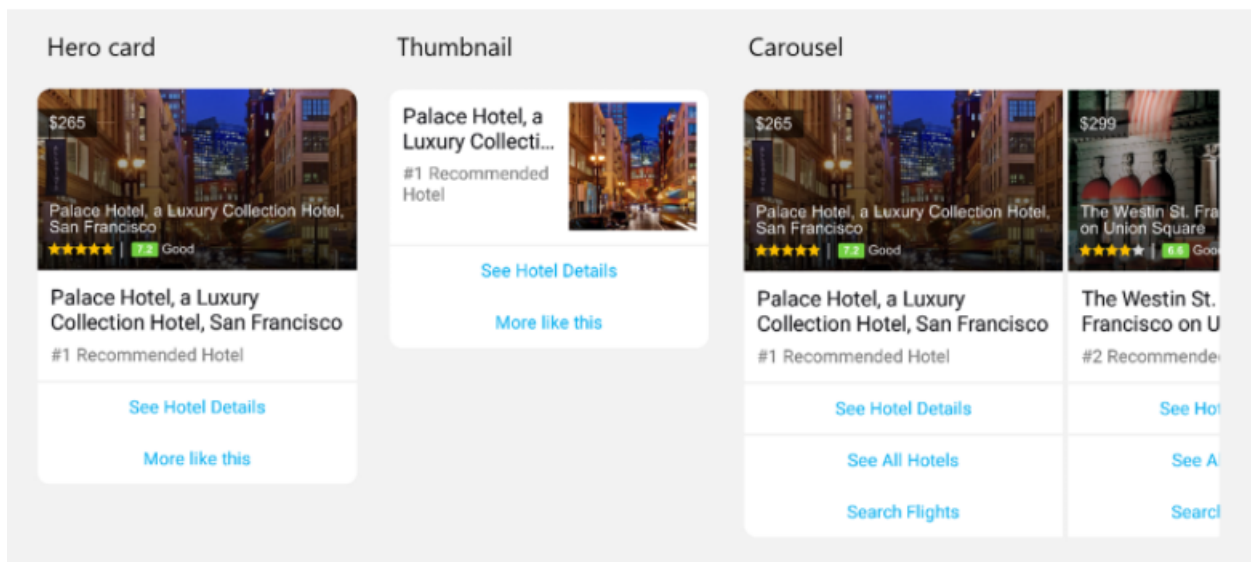
- Η hero card, που περιέχει συνήθως μία μεγάλη εικόνα, ένα ή περισσότερα κουμπιά και κείμενο.
- Η thumbnail, η οποία είναι μια κάρτα που περιέχει συνήθως μια εικονική μικρογραφία, ένα ή περισσότερα κουμπιά και κείμενο.
- Η video card, που μπορεί να αναπαράγει βίντεο.
- Η audio card, η οποία μπορεί να αναπαράγει ένα αρχείο ήχου.

- Η animation card η οποία είναι μια κάρτα που μπορεί να αναπαράγει κινούμενα GIF ή σύντομα βίντεο.
- Η receipt card, η οποία επιτρέπει σε ένα bot να παρέχει μια απόδειξη στον χρήστη. Περιλαμβάνει κατά κανόνα τη λίστα των στοιχείων που πρέπει να συμπεριληφθούν στην απόδειξη παραλαβής, πληροφορίες φόρων και άλλο κείμενο.
- Και η sign in card, η οποία επιτρέπει σε ένα bot να ζητήσει να συνδεθεί ένας χρήστης. Συνήθως περιέχει κείμενο και ένα ή περισσότερα κουμπιά που ο χρήστης μπορεί να κάνει κλικ για να ξεκινήσει τη διαδικασία σύνδεσης.

Επίσης υποστηρίζονται και carousels τα οποία μπορεί να περιέχουν μέχρι έξι κάρτες του ίδιου τύπου.

Οι rich cards στο Skype υποστηρίζουν τις παρακάτω ενέργειες[11] :

1. imBack, κείμενο του μηνύματος που αποστέλλεται στο bot (από τον χρήστη που έκανε κλικ στο κουμπί ή πατήθηκε στην κάρτα). Αυτό το μήνυμα (από χρήστη σε bot) θα είναι ορατό σε όλους τους συμμετέχοντες συνομιλίας μέσω της εφαρμογής που φιλοξενεί τη συνομιλία.
2. postBack, κείμενο του μηνύματος που αποστέλλεται στο bot (από τον χρήστη που έκανε κλικ στο κουμπί ή πατήθηκε στην κάρτα). Ορισμένες αιτήσεις πελατών ενδέχεται να εμφανίζουν αυτό το κείμενο στη ροή μηνυμάτων, όπου θα είναι ορατές σε όλους τους συμμετέχοντες της συνομιλίας.
3. openUrl, όταν γίνει κλικ στο κουμπί ή πατηθεί στην κάρτα ανοίγει ένα URL στο ενσωματωμένο πρόγραμμα περιήγησης.
4. Call, όταν γίνει κλικ στο κουμπί ή πατηθεί στην κάρτα θα πραγματοποιηθεί τηλεφωνική κλήση σε μορφή τηλ : 123123123123.



Σχήμα 3.4: Παραδείγματα καρτών που υποστηρίζονται.

3.3 SoftOne

3.3.1 Βασικές αρχές λειτουργίας

Το SoftOne, όπως αναφέρθηκε και παραπάνω είναι ένα σύστημα επιχειρησιακού σχεδιασμού (ERP). Είναι ένα πολύγλωσσο πακέτο λογισμικού που μπορεί να διαχειρίζεται τις λειτουργίες μίας ή περισσότερων εταιρειών ανά εγκατάσταση, και μπορεί να χρησιμοποιηθεί ταυτόχρονα από πολλούς χρήστες. Η θεματολογία του καλύπτει την διαχείριση της εφοδιαστικής αλυσίδας, των ανθρώπινων πόρων, την οικονομική και λογιστική διαχείριση και τη διαχείριση του marketing και των πωλήσεων, ενώ παράλληλα καλύπτει τις ειδικές λογιστικές και φορολογικές νομοθεσίες αρκετών χωρών και τις γενικά αποδεκτές αρχές της Λογιστικής.

Δομικά αποτελείται από υποενότητες που χρησιμοποιούν κοινή βάση δεδομένων και επικοινωνούν μεταξύ τους ανταλλάσσοντας πληροφορίες σε πραγματικό χρόνο και επιτρέποντας ένα μεγάλο αριθμό αυτόματων ενημερώσεων . Με την κατάλληλη παραμετροποίηση του, είναι δυνατή η προσαρμογή του στις ανάγκες οποιουδήποτε κλάδου της οικονομίας και οποιασδήποτε επιχείρησης.

Παρακάτω παρουσιάζεται ένας πίνακας, ο οποίος συνοψίζει τις βασικές ενότητες του συγκεκριμένου λογισμικού[1].

Διαχείριση Εφοδιαστικής Αλυσίδας	Οικονομική και Λογιστική Διαχείριση	Ανθρώπινοι Πόροι	Μάρκετινγκ και Πωλήσεις
Προγραμματισμός και πρόγραμμα παραγωγής (Production Planning and Scheduling)	Γενική Λογιστική (General ledger)	Διαχείριση Προσωπικού (Personnel planning)	Διαχείριση Πελατών (Customer Management)
Διαχείριση Υλικών και Διαχείριση Αποθήκης (Inventory management and Warehouse management)	Διαχείριση Παγίων (Assets accounting)	Μισθοδοσία (Payroll)	Διαχείριση Παραγγελιών Order management
MRP (Material Requirements Planning)	Κέντρα κόστους/κερδών (Cost center accounting/ Profit center accounting)	Διαχείριση Οργανωτικής Δομής (Organisational structure)	Πρόβλεψη Πωλήσεων (Forecasting)
	Λογαριασμοί Εισπρακτέοι/ Πληρωτέοι (Accounts receivable and payable)	Διαχείριση Χρόνου Εργαζομένων (Time management)	Προγραμματισμός Πωλήσεων (Sales planning)
	Προϋπολογισμός (Budgeting)	Πρόσληψη προσωπικού	
	Κοστολόγηση προϊόντων/ υπηρεσιών (Costing)		

3.3.2 Αρχιτεκτονική SoftOne

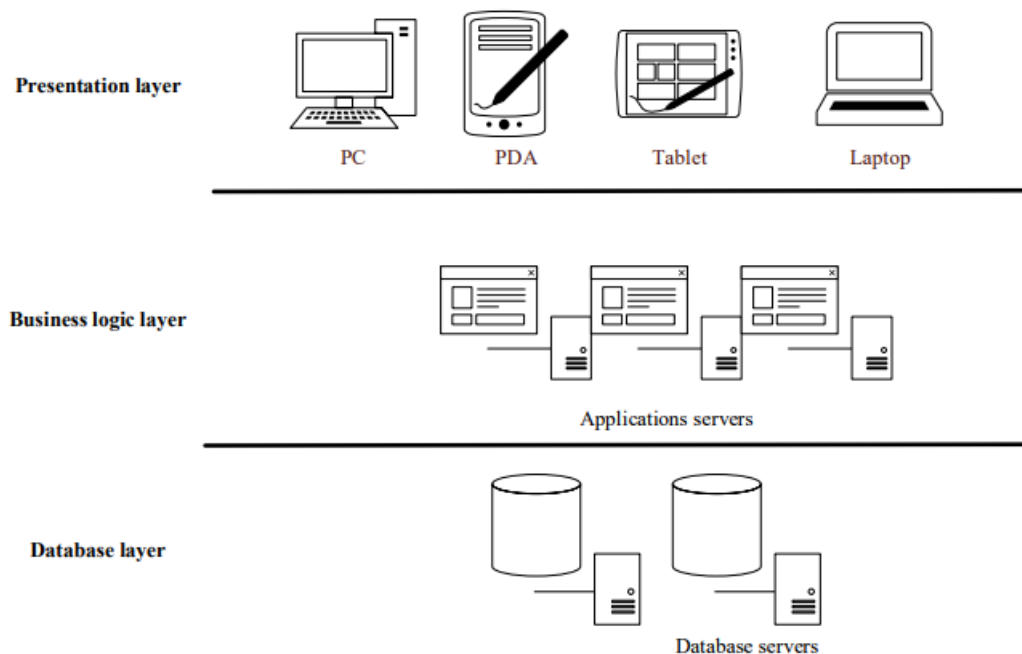
Σχεδιαστικά, έχει γίνει προσπάθεια ώστε το σύστημα να έχει υψηλό βαθμό τυποποίησης και παραμετροποίησης, να μπορεί να λειτουργεί στις ελάχιστες αναγκαίες υποδομές (πχ. Laptop), με μικρό ίχνος (small footprint), ενώ παράλληλα να μπορεί να λειτουργεί σε απαιτητικά περιβάλλοντα client-server εγκαταστάσεων με πολλούς χρήστες, καθώς και σε περιβάλλον Cloud.

Παρακάτω παρουσιάζονται τα βασικά χαρακτηριστικά της αρχιτεκτονικής του SoftOne ERP[1]:

- Δομή που βασίζεται σε πολλές ευδιάκριτες επιχειρησιακές ενότητες (modular design).
- Χρήση κοινής σχεσιακή βάση δεδομένων (Relational Database Management System - RDBMS) για την αποθήκευση όλων των επιχειρησιακών δεδομένων.
- Χρήση τυποποιημένων διεπαφών (interface) και γραφικών διεπαφών χρήστη.
- Δυνατότητα παραμετροποίησης.
- Δυνατότητα ορισμού επιχειρηματικών διαδικασιών (business processes) και ορισμού ροής εργασιών (workflows).
- Δυνατότητα προσθήκης επιπλέον κώδικα για επέκταση της λειτουργικότητας των ενότητων που υπάρχουν (σε JavaScript, C#).

Το SoftOne, όπως και τα περισσότερα ERP συστήματα έχει αρχιτεκτονική τριών επιπέδων.

1. Το επίπεδο βάσης δεδομένων (Database layer).
2. Το επίπεδο στο οποίο αναπτύσσεται η επιχειρησιακή λογική του προγράμματος (Business logic layer).
3. Το επίπεδο παρουσίασης (Presentation layer), που είναι η εμφάνιση στον τελικό χρήστη. Η παρουσίαση μπορεί να γίνει σε υπολογιστή ή laptop με desktop ή web interface και σε κινητό ή tablet με mobile interface.



Σχήμα 3.5: Αρχιτεκτονική τριών επιπέδων[1]

Το επίπεδο της βάσης δεδομένων και το επίπεδο λογικής είναι δυνατό να βρίσκονται στο ίδιο μηχάνημα, σε διαφορετικό μηχάνημα ή ακόμα και στο Cloud, με σκοπό την διευκόλυνση του πελάτη. Οι βάσεις που υποστηρίζονται είναι ο SQL Server, η Oracle και η MySQL (σε non production environments).

Το επίπεδο λογικής (application server) λειτουργεί σε περιβάλλον Windows. Αποτελείται από ένα αριθμό executable modules (περίπου 60 Dlls), τα οποία φορτώνονται ανάλογα με τη λειτουργικότητα και τον τρόπο κατανομής της εγκατάστασης σε ένα ή περισσότερα μηχανήματα. Οι επικοινωνίες μεταξύ των μηχανημάτων γίνονται με TCP πρωτόκολλο ¹, εκτός από την επικοινωνία με Web browsers, η οποία γίνεται σε HTTP² με την χρήση του IIS Server.

Το επίπεδο παρουσίασης βρίσκεται πάντα στο μηχάνημα του χρήστη. Αυτό μπορεί να είναι είτε ο ειδικός Client της εταιρείας που τρέχει σε περιβάλλοντα Windows, είτε Web Browsers σε διάφορα περιβάλλοντα, είτε ειδικές εφαρμογές σε mobile devices (Tablets, PDAs κλπ). Οι Web και mobile λύσεις υλοποιούνται με την χρήση Web Services σε Json format.

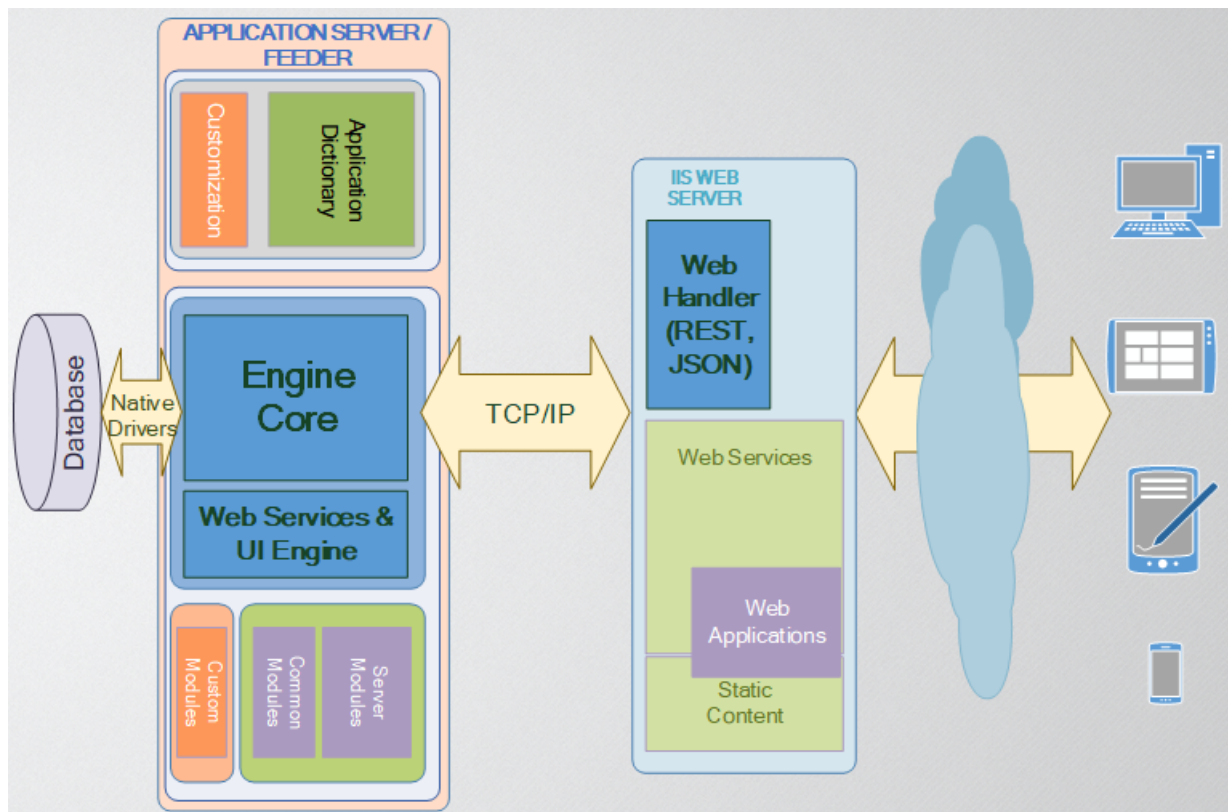
Όπως αναφέρθηκε, στις Web και mobile λύσεις μεσολαβεί και ένα ενδιάμεσο επίπεδο, στο Cloud, στο οποίο βρίσκεται ο Web Server. Η πραγματική υλοποίηση των Web Services γίνεται στον Application Server, και ο Web Server χρησιμοποιείται για την μεταφορά των απαντήσεων, καθώς και για τη φιλοξενία στατικού περιεχομένου των εφαρμογών. Η παραπάνω υλοποίηση προσφέρει υψηλή τυποποίηση των Web λύσεων, καθώς μπορεί να εξυπηρετήσει πελάτες οι οποίοι δεν έχουν δικές τους υποδομές στο Cloud, και απλά μπορούν να εξασφαλίσουν την παρουσία τους στο διαδίκτυο με την επικοινωνία της τοπικής τους εγκατάστασης με τους Web Servers της εταιρείας μέσω απλών TCP συνδέσεων (client initiated, ώστε να μπορούν να περάσουν τα

¹Το πρωτόκολλο ελέγχου μετάδοσης (TCP) είναι ένα από τα κύρια πρωτόκολλα του Internet. Σημαντικές εφαρμογές Internet, όπως το World Wide Web, το ηλεκτρονικό ταχυδρομείο, η απομακρυσμένη διαχείριση και η μεταφορά αρχείων, βασίζονται στο TCP[12].

²Το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) είναι ένα πρωτόκολλο εφαρμογής για κατανεμημένα, συνεργατικά και hypermedia πληροφοριακά συστήματα. Το HTTP είναι το θεμέλιο της επικοινωνίας δεδομένων για τον Παγκόσμιο Ιστό[13].

τοπικά firewalls).

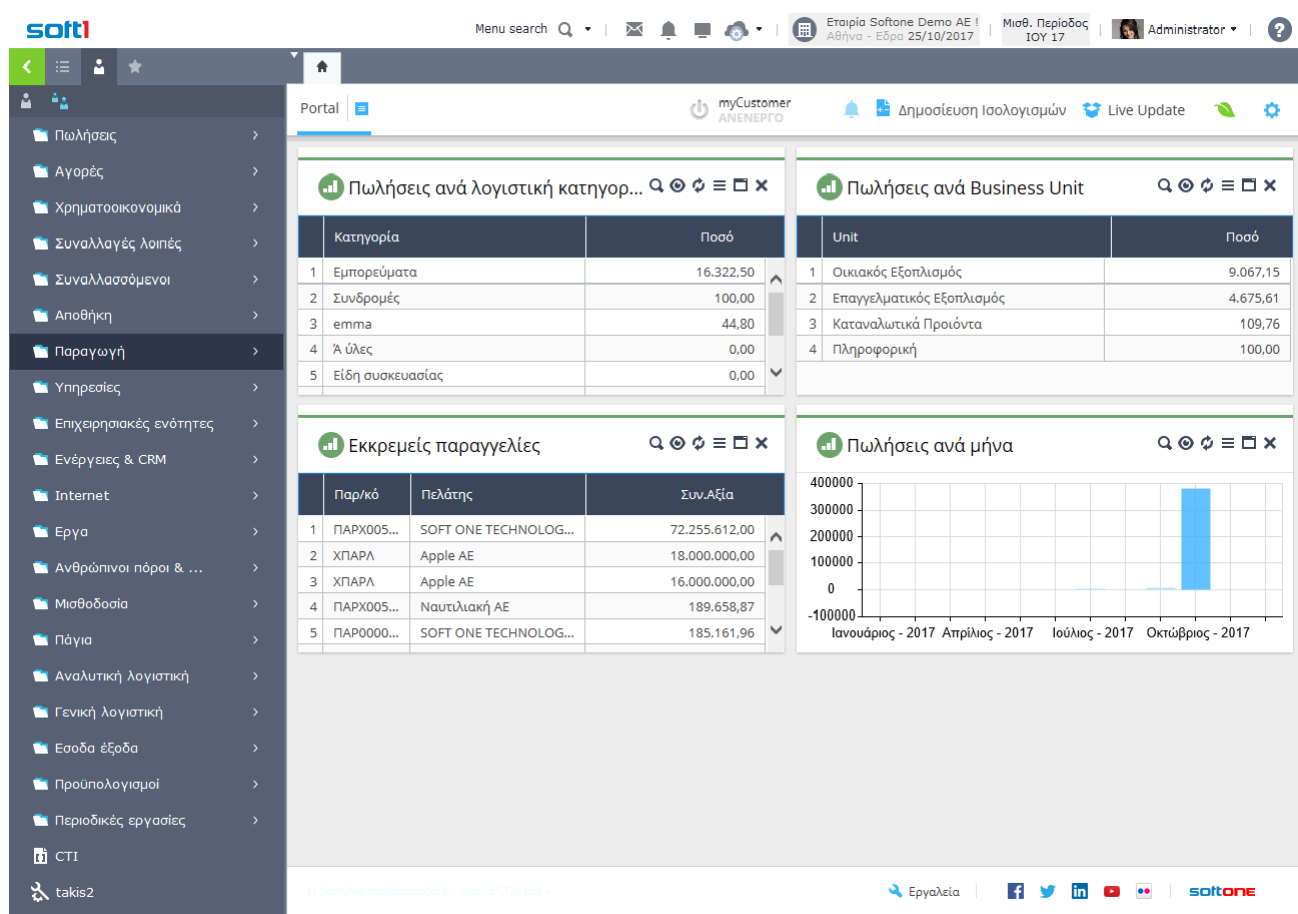
Στο παρακάτω σχήμα φαίνεται μια απεικόνιση της επικοινωνίας, όπως αναλύθηκε.



Σχήμα 3.6: Αρχιτεκτονική Web και mobile εφαρμογών

Στο σύστημα μπορεί να προστεθεί λειτουργικότητα και επιπλέον κώδικας που τρέχει στον application server (custom code). Ο κώδικας μπορεί να είναι είτε σε περιβάλλον .NET μέσω της χρήσης ενός ειδικού API της εταιρείας, είτε σε περιβάλλον Javascript. Στο σύστημα είναι ενσωματωμένη μία μηχανή Javascript (Chakra της Microsoft – η μηχανή του πλέον σύγχρονου Web Browser της εταιρείας - Edge), που επιτρέπει την ταχεία εκτέλεση κώδικα. Ο κώδικας αυτός είτε λειτουργεί παράλληλα με τον υπάρχοντα κώδικα των διαφόρων modules (πχ. Τιμολόγηση, διαχείριση Πελατών κλπ), είτε μπορεί να είναι τελείως αυτόνομος (Custom modules from scratch, νέα web services κλπ). Ο συγκεκριμένος τρόπος χρησιμοποιήθηκε για να επεκταθεί η λειτουργικότητα στην παρούσα εργασία.

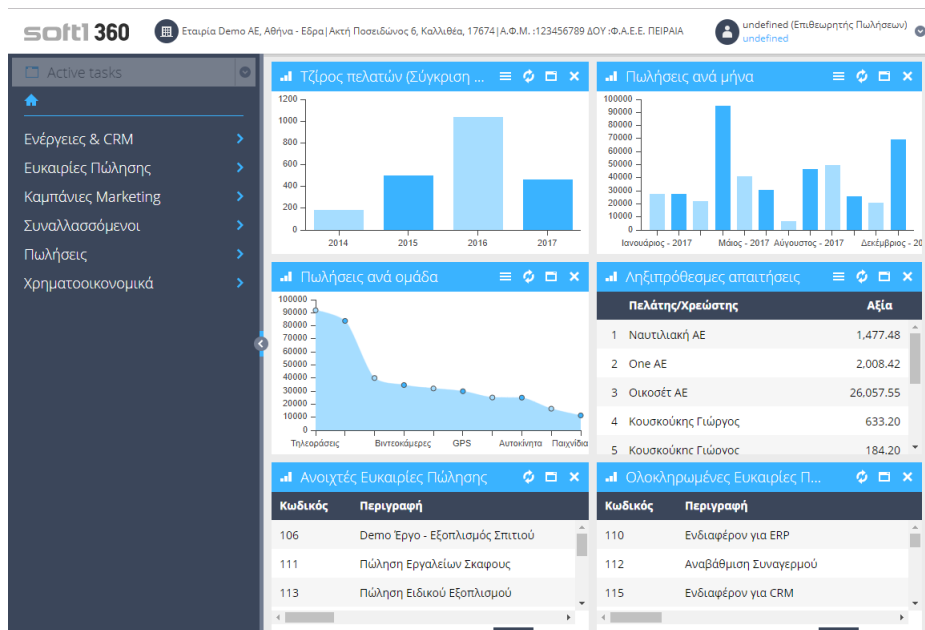
Παρακάτω φαίνονται κάποιες εικόνες από τα διάφορα user interfaces του SoftOne.



Σχήμα 3.7: Desktop User Interface

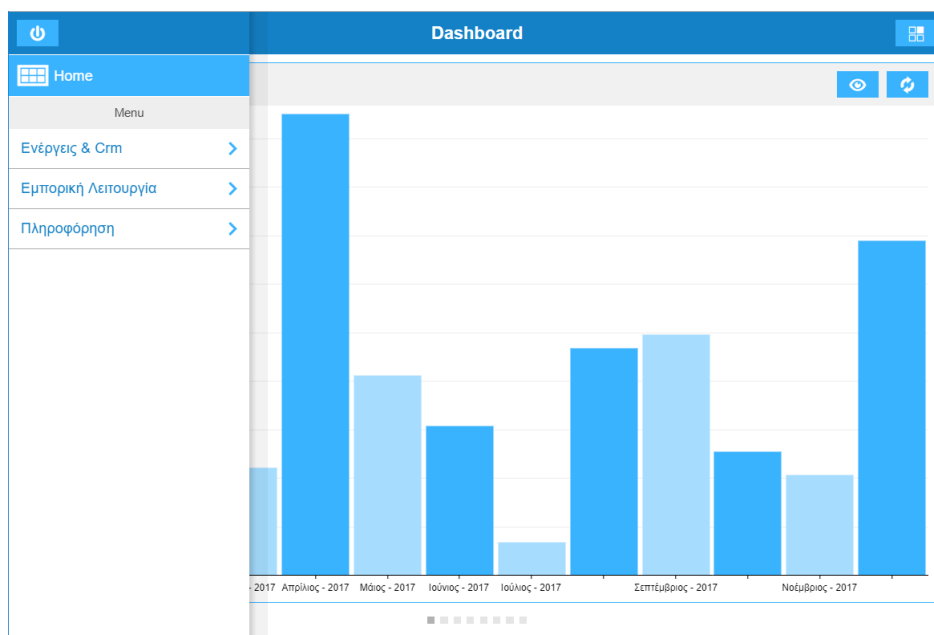
Όπως φαίνεται, αριστερά βρίσκεται το μενού της εφαρμογής, το οποίο μπορεί κάθε χρήστης να επεξεργαστεί ανάλογα με τις ανάγκες του και τα δικαιώματα που έχει στην προβολή δεδομένων.

Στο δεξί μέρος, βλέπουμε το portal του χρήστη που περιέχει gadgets. Τα gadgets είναι μία μορφή προβολής χρήσιμων πληροφοριών ή πρόσθετων λειτουργιών, όπως για παράδειγμα οικονομικά δεδομένα σε μορφή γραφημάτων. Ο κάθε χρήστης μπορεί να επιλέξει ή ακόμα και να δημιουργήσει (αν γνωρίζει) τα gadgets που αυτός επιθυμεί να εμφανίζονται στην βασική του σελίδα. Η αυτοτέλεια των gadgets και το γεγονός ότι περιέχουν συγκεντρωμένη πληροφορία, μας φάνηκε ιδιαίτερα χρήσιμη για το συγκεκριμένο έργο, καθώς τα προσαρμόσαμε έτσι ώστε να τα χρησιμοποιήσουμε και να προσθέσουμε επιπλέον λειτουργικότητα στο bot.



Σχήμα 3.8: Web User Interface

Το mobile interface, είναι πιο απλοϊκό, όπως είναι λογικό λόγω της χωρητικότητας της οθόνης και του περιορισμού των διαθέσιμων λειτουργιών μέσω κινητού.



Σχήμα 3.9: Mobile User Interface

3.4 Επικοινωνία συστημάτων

Έχουμε δημιουργήσει το “S1 bot Framework” όπου είναι το πρόγραμμα το οποίο επιτρέπει την δρομολόγηση του χρήστη Skype, στη σωστή εγκατάσταση. Όπως έχει αναφερθεί, το κάθε μήνυμα που αποστέλλεται στο bot περνάει πρώτα από το MBF και έπειτα μεταφέρεται στο S1 bot Framework. Το πρόγραμμα αυτό είναι υπεύθυνο :

1. Για την παραλαβή του μηνύματος που έστειλε ο χρήστης, από το MBF. Όπως αναφέρθηκε και παραπάνω, το Microsoft Bot Framework μετατρέπει τα μηνύματα που στέλνει ο χρήστης σε μία συγκεκριμένη μορφή, ανεξάρτητα από το κανάλι από το οποίο προέρχονται. Η μορφή αυτή φαίνεται παρακάτω.

Μορφή μηνύματος που λαμβάνουμε από MBF

```
Message:{ "text":"Γεια",
  "type":"message",
  "timestamp":"2017-11-20T14:36:25.533Z",
  "localTimestamp":"2017-11-20T16:36:25.533+02:00",
  "id":"1511188585540",
  "channelId":"skype",
  "serviceUrl":"https://smba.trafficmanager.net/apis/",
  "from":{
    "id":"29:1nn-IXY2R-NuE7KWqv4DxywDldw_tPL7GFAoEH65j2silLBe0guts6ZZaHVFzLV",
    "name":"Anna Valachi"
  },
  "conversation":{
    "id":"29:1nn-IXY2R-NuE7KWqv4DxywDldw_tPL7GFAoEH65j2silLBe0guts6ZZaHVFzLV"
  },
  "recipient":{
    "id":"28:37f3371d-5804-46cf-91d6-c5aa84a39378",
    "name":"S1 Eva"
  },
  "entities":[
    {
      "locale":"el-GR",
      "country":"GR",
      "platform":"Windows",
      "type":"clientInfo"
    }
  ],
  "channelData":{
    "text":"Γεια"
  }
}
```

Σχήμα 3.10: Message object

Το παραπάνω μήνυμα, μας αποστέλλεται από το MBF όταν ένας χρήστης στείλει το μήνυμα “Γεια” στο bot.

Από την παραπάνω εικόνα γίνονται αντιληπτά :

- Το όνομα και το id του χρήστη (αποστολέα), τα οποία περιέχονται στην παράμετρο from και στην παρούσα περίπτωση είναι Anna Valachi και 29:1nn-IXY2R-NuE7KWqv4DxywDldw_tPL7GFAoEH65j2silLBe0guts6... αντίστοιχα. Το id είναι ένας μοναδικός κωδικός που λαμβάνει ο κάθε χρήστης από το Skype. Αυτός ο κωδικός αντιστοιχίζεται από εμάς με έναν χρήστη SoftOne, ώστε κάθε φορά που αυτός ο χρήστης Skype αποστέλλει ένα μήνυμα, να αναγνωρίζεται ο αντίστοιχος χρήστης SoftOne και να γίνεται δρομολόγηση στη δική του εγκατάσταση.

- Το id και το όνομα του bot που περιέχονται στην παράμετρο recipient τα οποία είναι 28:37f3371d-5804-46cf-91d6-c5aa84a39378 και "S1 Eva" αντίστοιχα.
 - Το id της συνομιλίας, στην παράμετρο conversation το οποίο όπως βλέπουμε ταυτίζεται με εκείνο του χρήστη.
 - Το id του καναλιού από το οποίο έχει αποσταλεί το μήνυμα, όπου είναι το Skype στην συγκεκριμένη περίπτωση.
 - Το ακριβές μήνυμα που έστειλε ο χρήστης το οποίο βρίσκεται στην παράμετρο text.
 - Η χώρα καθώς και το είδος του λειτουργικού συστήματος από το οποίο στάλθηκε το μήνυμα.
2. Για την αντιστοίχιση του χρήστη Skype (αποστολέα) με τον σωστό χρήστη και την σωστή εγκατάσταση SoftOne στην περίπτωση που στέλνει μήνυμα πρώτη φορά ή για την δρομολόγηση στην σωστή εγκατάσταση στην περίπτωση που αυτός ο χρήστης έχει κάνει ήδη εγγραφή μια φορά. Τα στοιχεία που απαιτείται να δώσει ο χρήστης κατά την πρώτη του εγγραφή είναι το serial number της εγκατάστασης, το όνομα και ο κωδικός του χρήστη SoftOne με τον οποίο θέλει να συνδεθεί. Το συγκεκριμένο πρόγραμμα, αποθηκεύει σε μία βάση δεδομένων όλους τους χρήστες που έχουν εγγραφεί και αντιστοιχιστεί ώστε όταν ένας χρήστης στείλει μήνυμα, να τον αναζητήσει σε αυτή τη βάση.
 3. Τέλος, για την αποστολή της απάντησης στο Microsoft Bot Framework στην κατάλληλη μορφή.

Μορφή μηνύματος που στέλνουμε στο MBF

```
{
  "type": "message",
  "from": {
    "id": "28:37f3371d-5804-46cf-91d6-c5aa84a39378",
    "name": "S1 Eva"
  },
  "conversation": {
    "id": "29:1nn-IXY2R-NuE7KWqv4DxywDldw_tPL7GFAoEH65j2silLBe0guts6ZZaHVFzLV"
  },
  "recipient": {
    "id": "29:1nn-IXY2R-NuE7KWqv4DxywDldw_tPL7GFAoEH65j2silLBe0guts6ZZaHVFzLV",
    "name": "Anna Valachi"
  },
  "text": "Γεια σου"
}
```

Σχήμα 3.11: Message object

Κεφάλαιο 4

Παρουσίαση και λειτουργικότητα του bot

4.1 Εισαγωγή

Στο τέταρτο κεφάλαιο γίνεται επεξήγηση της λειτουργικότητας που έχει το SoftOne bot. Παρουσιάζονται οι αποφάσεις που έχουν ληφθεί και αφορούν τη λειτουργικότητα, η δομή του κώδικα και αναλύεται ο τρόπος με τον οποίο γίνεται η “κατανόηση” της επιθυμητής από τον χρήστη εργασίας από το bot.

4.2 Δυνατότητες bot

Το SoftOne, όπως και τα περισσότερα επιχειρησιακά λογισμικά, έχει πολλές λειτουργίες. Το βασικό ερώτημα για αυτή την μελέτη ήταν για ποιες από αυτές θα υπήρχε ενδιαφέρον να εξυπηρετηθούν μέσω ενός επιχειρησιακού βοηθού.

Όπως έχει ήδη αναφερθεί, ο προσανατολισμός ήταν το bot να έχει τις τρεις παρακάτω λειτουργίες.

Ερωτήσεις

Να έχει τη δυνατότητα να απαντάει σε συγκεκριμένες ερωτήσεις που εισάγει ο χειριστής.

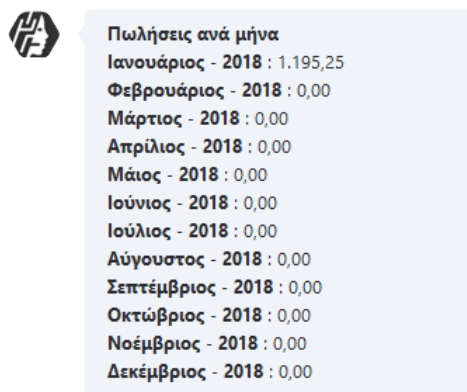
Είναι σημαντικό να αναφερθεί ότι ανάλογα με την θέση ή την ειδικότητα του , ο χρήστης έχει συγκεκριμένα δεδομένα στα οποία μπορεί να έχει πρόσβαση. Για παράδειγμα ένας υπάλληλος θα έχει πρόσβαση σε πολύ λιγότερα δεδομένα από ότι ο διευθυντής της εταιρείας και αυτό έπρεπε να ληφθεί σοβαρά υπόψη κατά τον σχεδιασμό του bot. Ο σκοπός, λοιπόν, ήταν να μεταφερθεί η ήδη υπάρχουσα στο Softone δυνατότητα πρόσβασης σε δεδομένα και λειτουργίες ανά χρήστη στο bot, ώστε η πρόσβαση να είναι ελεγχόμενη.

Σε ότι αφορά τις ερωτήσεις, έχει επιλεγεί ένα σύνολο ερωτήσεων που αρχικά θα υποστηρίζονται. Παραδείγματα αυτών αποτελούν οι εισπράξεις από πελάτες και χρεώστες και οι πληρωμές από προμηθευτές και πιστωτές που έχουν πραγματοποιηθεί σε ένα χρονικό διάστημα το οποίο επιλέγεται από τον χρήστη. Ακόμη, τα υπόλοιπα που έχει η εταιρεία από πελάτες, είτε από όλους σαν σύνολο είτε από κάποιον μεμονωμένο που θα επιλέξει ο χρήστης καθώς και τα υπόλοιπα που έχει η εταιρεία σε προμηθευτές, είτε σε όλους σαν σύνολο είτε σε κάποιον μεμονωμένο που επίσης θα επιλεγεί από τον χρήστη. Ο τελευταίος έχει την δυνατότητα να ρωτήσει και για το υπόλοιπο του είδους, που ο ίδιος θα επιλέξει, στην αποθήκη της εταιρείας. Επιπλέον μία από τις επιλεγμένες ερωτήσεις αποτελεί και η εικόνα της εταιρείας, στην οποία περιλαμβάνονται οι παραγγελίες της εταιρείας, οι εισπράξεις, οι πωλήσεις και οι βραχυπρόθεσμες απαιτήσεις και υποχρεώσεις της. Κάθε ερώτηση αντιστοιχεί σε ένα microservice, στο οποίο καθορίζεται όλη

η αντιμετώπιση του bot στη συγκεκριμένη ερώτηση.

Στις ερωτήσεις, έχει προστεθεί και κάποια επιπλέον λειτουργία που αφορά στην εκτέλεση κάποιων απλών καταχωρίσεων στο SoftOne από τον χρήστη μέσω του bot. Για παράδειγμα, ο τελευταίος έχει τη δυνατότητα να καταχωρίσει μία άδεια ή μία παραγγελία μέσω του bot. Η εκτέλεση τέτοιου τύπου εργασιών έχει προστεθεί στην κατάσταση των ερωτήσεων καθώς έχει ακριβώς ίδια λειτουργικότητα με τις ερωτήσεις, δηλαδή κάθε μία από αυτές τις εργασίες αντιστοιχεί σε ένα microservice, στο οποίο καθορίζεται όλη η αντιμετώπιση του bot σε περίπτωση που ζητηθεί η τελευταία.

Ακόμη, κάθε φορά που συνδέεται ένας χρήστης φορτώνονται όλα τα gadgets που εκείνος μπορεί να βλέπει, με χρήση COM(Component Object Model) τεχνολογίας. Ο ορισμός των gadgets δόθηκε σε προηγούμενο κεφάλαιο. Το γεγονός ότι τα gadgets είναι αυτόνομα, δηλαδή εύκολα διαχειρίσιμα και περιέχουν χρήσιμες πληροφορίες μας οδήγησε να τα χρησιμοποιήσουμε ώστε να προσθέσουμε επιπλέον λειτουργικότητα στο bot. Για κάθε ένα από αυτά, δημιουργείται δυναμικά ένα microservice που περιέχει όλη τη λειτουργικότητα του. Ο χειριστής έχει τη δυνατότητα να ρωτήσει το bot τις πληροφορίες που περιέχουν κάποια από αυτά.



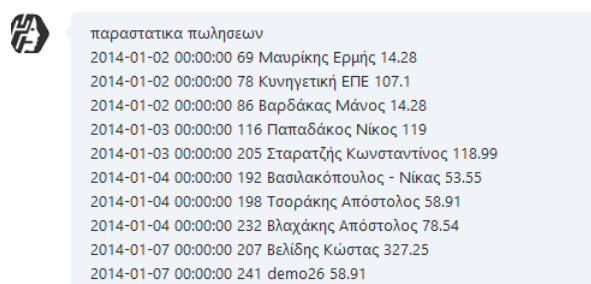
Σχήμα 4.1: Παράδειγμα εμφάνισης gadget στο bot

Πωλήσεις ανά μήνα

Μήνας	Ποσό
1 Ιανουάριος - 2018	1.195,25
2 Φεβρουάριος - 2018	0,00
3 Μάρτιος - 2018	0,00
4 Απρίλιος - 2018	0,00
5 Μάιος - 2018	0,00
6 Ιούνιος - 2018	0,00
7 Ιούλιος - 2018	0,00
8 Αύγουστος - 2018	0,00
9 Σεπτέμβριος - 2018	0,00
10 Οκτώβριος - 2018	0,00
11 Νοέμβριος - 2018	0,00
12 Δεκέμβριος - 2018	0,00

Σχήμα 4.2: Παράδειγμα εμφάνισης gadget στο SoftOne

Ένα πρόβλημα που αντιμετωπίσαμε και αφορά την απεικόνιση των gadgets στο bot είναι οι περιορισμοί του user interface του Skype. Δηλαδή, η αδυναμία μας να μορφοποιήσουμε σωστά τα δεδομένα που λαμβάνουμε από τα gadgets που περιέχουν πάνω από δύο στήλες δεδομένων. Λόγω του περιορισμού αυτού, επιλέξαμε να προσθέσουμε στην λειτουργικότητα του bot μόνο τα gadgets που περιέχουν δύο στήλες δεδομένων όπως αυτό που φαίνεται παραπάνω.



Σχήμα 4.3: Παράδειγμα εμφάνισης gadget με παραπάνω από δύο στήλες

Εργασίες

Να έχει τη δυνατότητα ο χρήστης να εκτελέσει όποια εργασία από το μενού του επιθυμεί.

Ο κάθε χρήστης, έχει ένα δικό του προσωπικό μενού στο περιβάλλον του SoftOne, το οποίο έχει δημιουργήσει ο ίδιος ή η επιχείρηση στην οποία εργάζεται. Αυτό συμβαίνει, ώστε να ελέγχεται η πρόσβαση αλλά και να μπορεί ο κάθε χρήστης να τοποθετήσει στο προσωπικό του μενού τις ενότητες τις οποίες επισκέπτεται πιο συχνά.

Κάθε φορά που ένας χρήστης συνδέεται, φορτώνεται το μενού του και του εμφανίζονται οι επιλογές του ως υπερσύνδεσμοι. Με αυτό τον τρόπο του δίνεται η δυνατότητα να ζητήσει μία εργασία από το bot και το τελευταίο να τον μεταφέρει, με ένα κλικ, απευθείας στο παράθυρο εκτέλεσης της εργασίας στο Web ή το mobile (ανάλογα με το περιβάλλον στο οποίο βρίσκεται) application του SoftOne.

Ειδοποιήσεις

Να στέλνει ειδοποιήσεις στον χρήστη. Οι ειδοποιήσεις αυτές χωρίζονται σε τρεις κατηγορίες:

Τύπου 1 : Τακτικές, για γενική ενημέρωση (πχ. Πωλήσεις, Εισπράξεις, Πληρωμές ημέρας, λιανικές πωλήσεις ανά κατάστημα, παρακολούθηση δεικτών ανά ώρα, κλπ). Οι δυνατότητες που έχει ο χρήστης για αυτού του τύπου τις ειδοποιήσεις είναι ακριβώς οι ίδιες με τις δυνατότητες που έχει στις ερωτήσεις.

Τύπου 2 : Έκτακτες, με την εμφάνιση συγκεκριμένων γεγονότων(πχ.έλλειψη είδους, χαμηλό ταμείο, υπέρβαση συγκεκριμένων ορίων κλπ).

Τύπου 3 : Ειδοποιήσεις που απαιτούν αντίδραση από τον χρήστη. Στην κατηγορία αυτή υπάγονται διαδικασίες εγκρίσεων που μπορούν να βοηθηθούν με τη χρήση του bot. Για παράδειγμα η έγκριση μιας πληρωμής ή μιας αγοράς πάνω από ένα ορισμένο όριο, η αποδοχή μιας μεγάλης επιταγής, η έγκριση μιας σημαντικής για την επιχείρηση ενέργειας κλπ.

Οι Τύπου 2 και 3 εξυπηρετούνται ιδιαίτερα από την δυνατότητα των των bots να λειτουργούν σε κινητές συσκευές και προσφέρουν άμεση ενημέρωση και δυνατότητα αντίδρασης στον χρήστη όπου και να βρίσκεται.

4.3 Σημαντικές αποφάσεις για την υλοποίηση του έργου

Τα βασικά ζητούμενα για την υλοποίηση του έργου ήταν :

1. Η αναγνώριση της απαίτησης του χρήστη, δηλαδή αποκωδικοποίηση της φράσης που ίδιος στέλνει στο bot.
2. Η απόφαση του τι πρέπει να εκτελεσθεί για να ανακτήσουμε την απάντηση στην ερώτηση του χρήστη και πως θα την παρουσιάσουμε σε αυτόν.

Για να καλυφθούν τα παραπάνω ήταν αναγκαίο να ληφθούν κάποιες σημαντικές αποφάσεις κατά την υλοποίηση του έργου. Αυτές αφορούσαν τη δομή των microservices και τον τρόπο που θα πραγματοποιείται η ανάλυση των λέξεων ώστε να καθορίσουμε την κατανόηση που θα έχει το bot.

4.3.1 Microservices

Η περιγραφή όλων των microservices είναι αντίστοιχη και γίνεται σε Json format, ενώ ο κώδικας υλοποίησης αναμειγνύεται με την περιγραφή με την χρήση της συμβατής γλώσσας Javascript. Ο συνδυασμός Javascript και Json, δίνει την δυνατότητα συνύπαρξης στον ίδιο πηγαίο κώδικα των ιδιοτήτων (properties) του microservice (μέσω Json) και του αντίστοιχου προγράμματος που χρειάζεται να εκτελεσθεί (μέσω Javascript). Έτσι δίνεται αυτονομία στην οργάνωση του κώδικα, δυνατότητα επέκτασης του αριθμού και των δυνατοτήτων των microservices, και πιθανόν δυνατότητα customization (επέκταση των δυνατοτήτων του συστήματος από χρήστες με γνώσεις προγραμματισμού). Κάθε ένα από αυτά έχει τις παρακάτω ιδιότητες:

- Τίτλο.
- Id.
- Access
- Command.

Τίτλος :

Ο τίτλος είναι οι λέξεις κλειδιά, τις οποίες αν πληκτρολογήσει ο χρήστης θα εκτελεστεί το συγκεκριμένο microservice. Ο τρόπος με τον οποίο γίνεται η σύγκριση αυτών που πληκτρολογεί ο χρήστης με τους τίτλους θα παρουσιαστεί αναλυτικά παρακάτω.

Id :

Το id είναι ένα μοναδικό κλειδί που έχει το κάθε microservice, ώστε να υπάρχει η δυνατότητα αναφοράς σε αυτό.

Access :

Αυτή η ιδιότητα περιέχει το όνομα του αντικείμενου όπως είναι αποθηκευμένο στη βάση δεδομένων του SoftOne. Ο σκοπός αυτής της ιδιότητας είναι να δώσει τη δυνατότητα για τον έλεγχο του δικαιώματος του χρήστη ως προς το συγκεκριμένο αντικείμενο.

Η ιδιότητα access είναι προαιρετική. Υπάρχει αποκλειστικά στα microservices στα οποία υπάρχει ανάγκη να ελεγχθεί το δικαίωμα του χρήστη για πρόσβαση στις πληροφορίες. Για παράδειγμα δεν υφίσταται η ιδιότητα access στο microservice που ορίζει το βασικό μενού του bot.

Command :

Το command είναι το σημείο όπου καθορίζεται όλη η λειτουργικότητα του microservice. Είναι η συνάρτηση που καθορίζει πως θα βρούμε τις πληροφορίες που έχουν ζητηθεί από τον χρήστη. Με άλλα λόγια σε αυτό το σημείο βρίσκεται ο κώδικας (σε JavaScript) που θα εκτελεσθεί ώστε να εξυπηρετηθεί η απαίτηση του χρήστη.

Παρακάτω φαίνεται ένα πολύ απλό παράδειγμα microservice, το οποίο ενεργοποιείται όταν ο χρήστης γράφει «Γεια» και εμφανίζει την απάντηση «Γεια σου» και ένα πιο σύνθετο παράδειγμα στο οποίο όταν ο χρήστης γράφει «Εισπράξεις» του εμφανίζονται οι ημερήσιες εισπράξεις της εταιρείας (μόνο στην περίπτωση που έχει δικαίωμα για αυτή την πληροφορία). Ακόμη φαίνεται και ένα παράδειγμα ενός gadget microservice.

Παράδειγμα απλού microservice

```
var simpleExample = {  
  title : ["Γεια"],  
  id : "welcome",  
  command : function(input){  
    var result = {};  
    result.text = "Γεια σου";  
    return result;  
  } }  
}
```

Παράδειγμα πιο σύνθετου microservice

```
var collections = {  
  title : ["Εισπράξεις"],  
  id : "receipt",  
  access : ["CFNCUSDOC"],  
  command : function(input){  
    var result = {};  
    var d = new Date(),  
    day = d.getDate();  
    if (day < 10){  
      day = "0" + day;  
    }  
    var month = d.getMonth();  
    month += 1;  
    if (month < 10){  
      month = "0" + month;  
    }  
    var year = d.getFullYear().toString(),  
    dateStr = year + month + day ,  
    queryCash = sql query to find cash collections,  
    queryCard = sql query to find card collections,  
    queryEp = sql query to find collections from checks,  
    queryEmv = sql query to find collections from remittances;  
    var dataCash = sql2json(queryCash),  
    dataCard = sql2json(queryCard),  
    dataEp = sql2json(queryEp),  
    dataEmv = sql2json(queryEmv);  
    result.text = "Μετρητά : " + X.FORMATFLOAT(dataCash[0].XAMOUNT,'V') +  
      "\nΚάρτες : " + X.FORMATFLOAT(dataCard[0].XAMOUNT,'V'+  
      "\nΕπιταγές : " + X.FORMATFLOAT(dataEp[0].XAMOUNT,'V'+  
      "\nΕμβάσματα : " + X.FORMATFLOAT(dataEmv[0].XAMOUNT,'V');  
    return checkResult(result, input.id, input); } }  
}
```

Παράδειγμα gadget microservice

```
var newCommand = {
  title : [gadget.name],
  id : "gdt"+i,
  axes : gadget.data.axes,
  sql : gadget.data.sqlQuery,
  command : function(input){
    var result = {};
    result.text = this.title[0] + "\n";
    var data = sql2json(this.sql);
    if (data==null){
      result.text = "Δεν υπάρχουν δεδομένα";    }
    else{
      var count = data.length;
      for (var i=0; i<count; i++){
        var temp = data[i][this.axes.X];
        if (temp.length>23){
          temp =temp.slice(0,20)+" ...";
        }
        result.text += "***"+temp + " **: " +
          parseFloat(Math.round(data[i][this.axes.Y] * 100) / 100).toFixed(2) + '\n';
      }
    }
    return checkResult(result, input.id, input); } };
```

Όπως παρατηρούμε, όλα τα παραπάνω microservices έχουν τουλάχιστον ένα τίτλο. Η παράμετρος title περιλαμβάνει ένα πίνακα (array), ο οποίος περιέχει έναν ή περισσότερους τίτλους, δηλαδή ένα ή περισσότερα λεκτικά με τα οποία έχουμε τη δυνατότητα να αναφερθούμε σε αυτό το microservice. Ακόμη, έχουν ένα id το οποίο είναι το μοναδικό κλειδί τους και ένα πίνακα, που περιέχεται στην παράμετρο access, ο οποίος περιέχει το ένα ή τα περισσότερα αντικείμενα της βάσης που ο χρήστης πρέπει να έχει το δικαίωμα για πρόσβαση ώστε να λάβει τις συγκεκριμένες πληροφορίες. Τέλος, όλα έχουν ένα command στο οποίο όπως βλέπουμε πραγματοποιείται ολόκληρη η επικοινωνία με τη βάση.

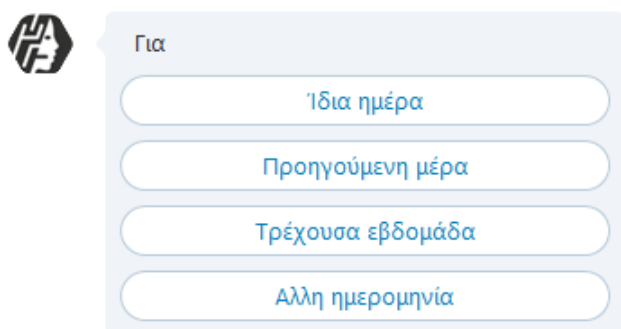
Παρότι τα συγκεκριμένα microservices λειτουργούν ικανοποιητικά, δημιουργήθηκε η ανάγκη για λήψη πληροφοριών από τον χρήστη ώστε να εκτελεστεί μία εργασία. Η ανάγκη, αυτή που υπήρξε για την εισαγωγή ενός ή περισσότερων στοιχείων (παραμέτρων) για την εκτέλεση μιας εργασίας απο το bot, μας οδήγησε στην κατασκευή μιας βιβλιοθήκης για την εξυπηρέτηση των πολλαπλών ερωτήσεων με σειριακή μορφή (είτε με απλή διακλάδωση). Μία επιπλέον ιδιότητα που προσθέσαμε είναι αυτή της φόρμας. Για παράδειγμα στο microservice των εισπράξεων θεωρήσαμε απαραίτητο ο χρήστης να εισάγει την ημερομηνία για την οποία θέλει να ενημερωθεί. Η φόρμα είναι μια γλώσσα περιγραφής σειράς ερωτήσεων, δυνατοτήτων επιλογών και διακλαδώσεων σε μορφή Json, που προβάλλονται σαν ομάδα στον χρήστη και οι απαντήσεις επιστρέφουν υπό μορφή πεδίων σε ένα Javascript object. Είναι δυνατό να περιέχει ένα ή περισσότερα πεδία (fields) ανάλογα με τον αριθμό των πληροφοριών που επιθυμούμε να λάβουμε από τον χρήστη. Το κάθε πεδίο περιλαμβάνει το όνομα του πεδίου, με το οποίο έχει κάποιος τη δυνατότητα να αναφερθεί σε αυτό, το μήνυμα το οποίο θα εμφανιστεί στον χρήστη όταν ο έλεγχος δοθεί στο συγκεκριμένο πεδίο και τον τύπο του μηνύματος του πεδίου.

Οι τύποι μηνυμάτων που έχουμε επιλέξει και δημιουργήσει είναι οι παρακάτω :

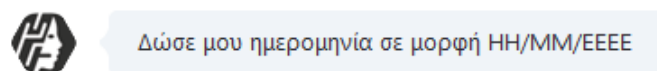
1. Απλό μήνυμα κειμένου (type = text), το οποίο έχει λεκτικό περιεχόμενο και πιο

συγκεκριμένα μία συνήθως διευκρινιστική ερώτηση που απαιτείται ώστε να αναγνωριστεί τι αναζητά ο χρήστης.

2. Μήνυμα τύπου option (type = option). Το μήνυμα αυτού του τύπου εμφανίζει κάποιες επιλογές στον χρήστη και αυτός καλείται να επιλέξει κάποια από αυτές ώστε να συνεχίσει την εργασία του.
3. Μήνυμα τύπου optiongoto (type = optiongoto). Ο συγκεκριμένος τύπος δημιουργήθηκε για να καλύψει την ανάγκη που είχαμε να εμφανίζουμε επιλογές στον χρήστη όπου η κάθε μία θα οδηγεί τον χρήστη σε κάτι διαφορετικό. Για παράδειγμα όταν ο χρήστης θέλει να μάθει τις εισπράξεις του έχει τις επιλογές : Για ίδια ημέρα, προηγούμενη ημέρα, τρέχουσα εβδομάδα, άλλη ημερομηνία. Όταν επιλέξει ίδια ημέρα θα του εμφανιστούν κατευθείαν οι εισπράξεις της τρέχουσας ημέρας ενώ όταν επιλέξει άλλη ημερομηνία θα πρέπει να μεταβεί σε ένα πεδίο (text) που θα του ζητείται η ημερομηνία η οποία τον ενδιαφέρει.



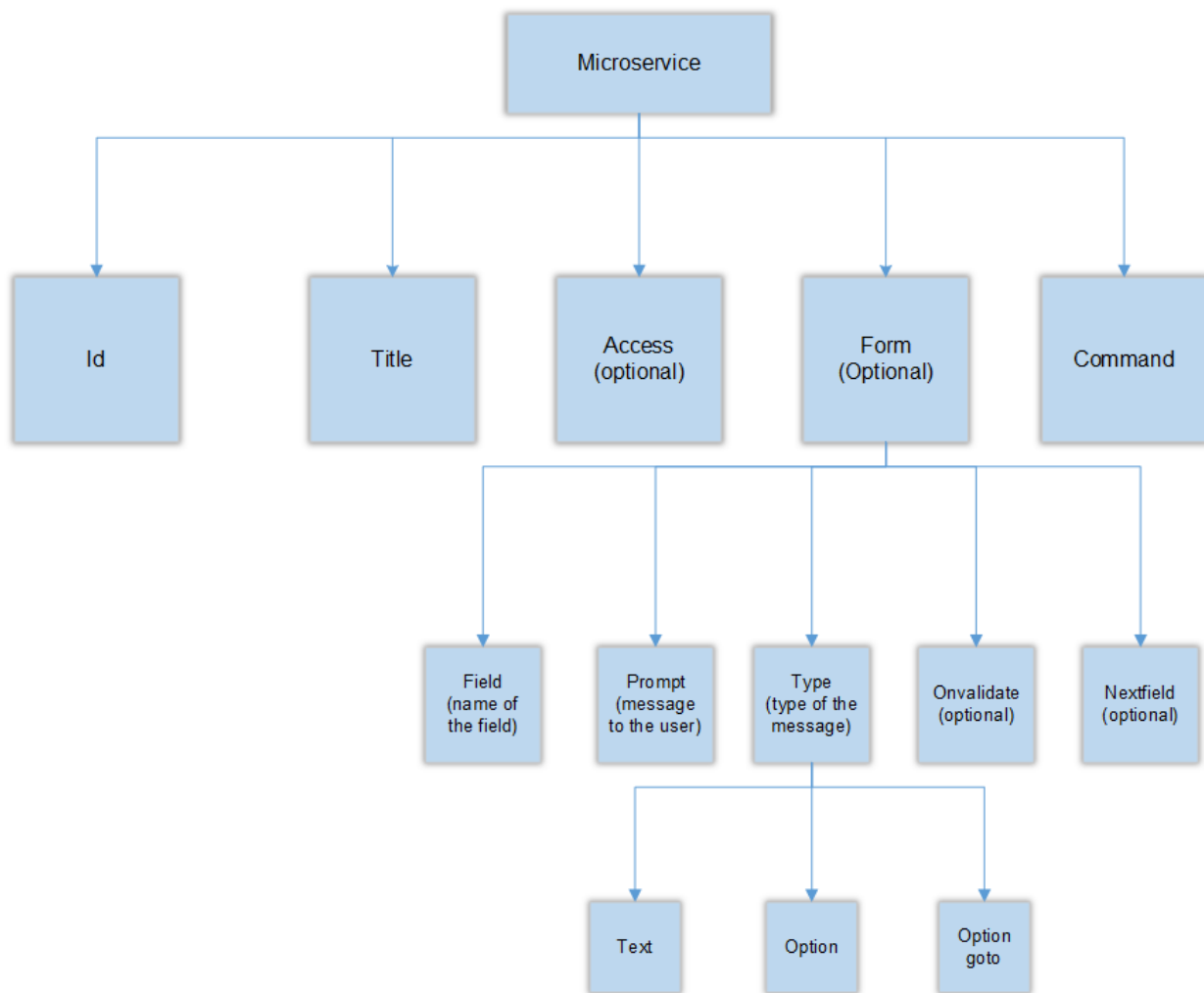
Σχήμα 4.4: Μήνυμα τύπου optiongoto



Σχήμα 4.5: Απλό μήνυμα κειμένου

Ακόμη, τα πεδία μπορεί να έχουν και κάποια προαιρετικά στοιχεία που είναι το onvalidate, το οποίο αν υπάρχει περιέχει μία συνάρτηση η οποία ελέγχει την εγκυρότητα του στοιχείου που πληκτρολόγησε ο χρήστης και το nextfield που αν υπάρχει περιέχει το επόμενο πεδίο της φόρμας στο οποίο θα δοθεί ο έλεγχος αυτόματα μετά από αυτό το πεδίο. Σε περίπτωση που το nextfield δεν υπάρχει στο πεδίο, ο έλεγχος δίνεται στο επόμενο σε σειρά πεδίο, αν υπάρχει, αλλιώς πραγματοποιείται έξοδος από την φόρμα.

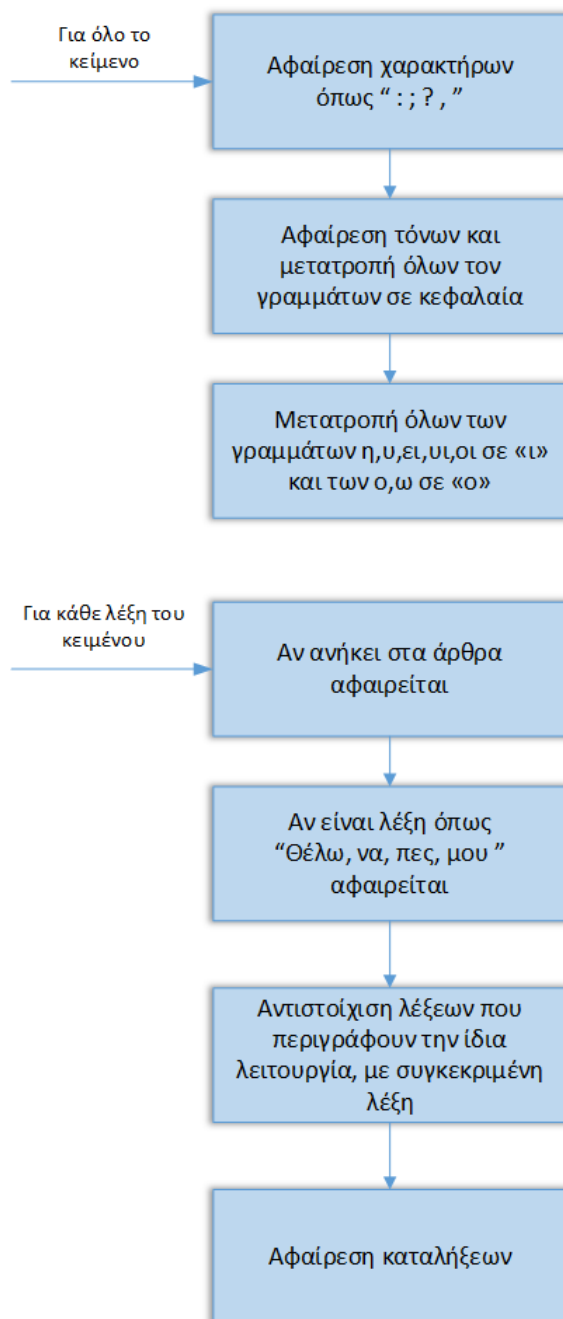
Πρέπει να σημειωθεί ότι η φόρμα είναι δυνατό και να μην υπάρχει, στην περίπτωση που δεν υπάρχει κανένα στοιχείο επιπλέον που να χρειάζεται από τον χρήστη ώστε να εκτελεστεί η εργασία που ζητήθηκε.



Σχήμα 4.6: Δομή microservice

4.3.2 Ανάλυση λέξεων

Ένα από τα σημαντικότερα προβλήματα που πρέπει να λύσει ένα bot σύστημα, είναι να αποκωδικοποιήσει την απαίτηση του χρήστη , μέσα από λεκτική ανάλυση και να την αντιστοιχίσει με κάποια διαθέσιμη εργασία που μπορεί να εκτελέσει, είτε με κάποια συγκεκριμένη, είτε, - αν δεν μπορέσει να την ταυτοποιήσει, -να προτείνει παρόμοιες εργασίες. Για αυτό το σκοπό δημιουργήθηκε μία συνάρτηση η οποία πραγματοποιεί την ανάλυση (ή parsing όπως ονομάζεται) του κειμένου που αποστέλλεται στο bot. Η συνάρτηση αυτή παίρνει σαν όρισμα το κείμενο που γράφτηκε από τον χρήστη.



Σχήμα 4.7: Διάγραμμα αλγορίθμου ανάλυσης κειμένου

Αρχικά, όπως φαίνεται και στο διάγραμμα ελέγχεται αν στο κείμενο περιέχονται οι χαρακτήρες « ' : ; ? , » και στην περίπτωση που υπάρχουν αφαιρούνται από το αυτό. Αφαιρούνται όλοι οι τόνοι (αν υπάρχουν) και μετατρέπονται όλοι οι χαρακτήρες στους αντίστοιχους κεφαλαίους. Στη συνέχεια, μετατρέπει τους χαρακτήρες [η, υ, ει, υι, οι] σε “ι” και τους [ο,ω] σε “ο” ώστε να παρακαμφθούν τυχόν ορθογραφικά λάθη.

Έπειτα, για κάθε ξεχωριστή λέξη που περιλαμβάνει το κείμενο :

- Ελέγχεται αν ανήκει σε μία λίστα, που έχει δημιουργηθεί, από λέξεις που είναι πιθανό να πληκτρολογήσει ο χρήστης, οι οποίες δεν έχουν σχέση με την εργασία που θέλει να εκτελέσει ο τελευταίος. Στην περίπτωση που ανήκει σε αυτή τη λίστα, αφαιρείται από το κείμενο. Η συγκεκριμένη λίστα περιλαμβάνει τις λέξεις : [Ο, Η, ΤΟ, ΤΑ, ΤΩΝ, ΤΗΣ, ΤΙΣ, ΤΟΝ, ΤΗΝ, ΤΗ, ΤΟΥΣ, ΤΟΥ, ΟΙ, ΣΤΑ, ΣΤΟΥΣ, ΣΤΟ, ΣΤΟΝ, ΣΤΗ, ΣΤΗΝ, ΣΤΙΣ, ΣΤΩΝ, ΣΟΥ, ΜΟΥ, ΝΑ, ΘΑ, ΣΕ, ΠΑΡΑΚΑΛΩ, ΘΕΛΩ, ΔΩΣΕ, ΓΙΑ, ΠΕΣ, ΠΕΙΣ, ΤΙ, ΠΩΣ, ΜΑΘΩ, ΜΕ].
- Ελέγχεται αν ανήκει σε μία λίστα που περιλαμβάνει διαφορετικές λέξεις που έχουν το ίδιο νόημα και τις αντιστοιχίζει με μία μεμονωμένη. Στην περίπτωση που ανήκει αντικαθίσταται με εκείνη με την οποία έχει αντιστοιχηθεί. Ένα παράδειγμα τέτοιων λέξεων είναι “ΠΡΟΪΟΝ”, “ΕΜΠΟΡΕΥΜΑ”, οι οποίες αντικαθίστανται με τη λέξη “ΕΙΔΟΣ”, με τον όρο, δηλαδή, με τον οποίο περιέχεται στο SoftOne.
- Αφαιρείται η κατάληξη της, με σκοπό να μην έχει σημασία ο χρόνος ή το πρόσωπο που έχει χρησιμοποιήσει ο χρήστης στην αναγνώριση της λέξης.

Για παράδειγμα, η φράση “Θέλω να μου πεις τις εισπράξεις μου”, έπειτα από την ανάλυση της με την παραπάνω συνάρτηση μετατρέπεται στην φράση “ΙΣΠΡΑΞ”, με αποτέλεσμα να έχουμε ανακτήσει τη βασική εργασία που επιθυμεί ο χρήστης.

Από την ίδια συνάρτηση ανάλυσης λέξεων περνούν και όλοι οι τίτλοι των microservices κατά την εκκίνηση, ώστε να βρίσκονται στην ίδια ακριβώς μορφή με τα μηνύματα του χρήστη. Για παράδειγμα, ο τίτλος “Εισπράξεις” θα μετατραπεί στον τίτλο “ΙΣΠΡΑΞ” κατά την εκκίνηση.

Έχει δημιουργηθεί και μία ακόμα συνάρτηση η οποία ελέγχει σε ποιο βαθμό η φράση που πληκτρολόγησε ο χρήστης (αφού πρώτα αναλυθεί με την μέθοδο που περιγράψαμε) μοιάζει με κάθε έναν από τους τίτλους των microservices, τα οποία περιέχονται σε ένα πίνακα (array) ώστε να είναι εύκολα προσπελάσιμα. Η συνάρτηση αυτή δέχεται ως ορίσματα την επεξεργασμένη φράση του χρήστη και τον τίτλο του microservice που ελέγχεται εκείνη τη στιγμή. Η βασική λειτουργία της συγκεκριμένης συνάρτησης είναι ο υπολογισμός του βαθμού ομοιότητας των δύο ορισμάτων (τον οποίο αποκαλούμε σκορ). Όσο μικρότερο είναι το σκορ, τόσο μεγαλύτερη είναι η ομοιότητα μεταξύ των δύο φράσεων. Όταν στο σκορ είναι 0 υποδεικνύει απόλυτη ταύτιση. Όταν το σκορ είναι 10 υποδεικνύει ότι δεν υπάρχει καμία συσχέτιση μεταξύ των δύο φράσεων. Τα υπόλοιπα σκορ υπολογίζονται με τον εξής τρόπο :

1. Εντοπίζεται ο αριθμός των λέξεων που υπάρχουν στη φράση του χρήστη (count).
2. Εντοπίζεται ο αριθμός αυτών που υπάρχουν και στον τίτλο (found).
3. Το σκορ είναι ίσο με $1 + \text{count} - \text{found}$.

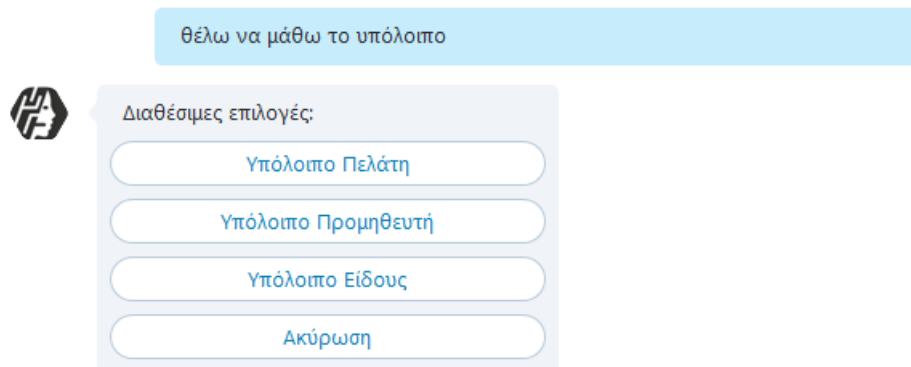
Δηλαδή όταν η φράση του χρήστη αποτελείται από δύο λέξεις και μόνο η μία εξ αυτών περιέχεται στον τίτλο το σκορ είναι 2.

Με αυτό τον τρόπο ορίστηκε ένα μέτρο ομοιότητας μεταξύ των δύο φράσεων.

Όταν το σκορ μίας πρότασης του χρήστη με ένα microservice είναι 0, τότε εκτελείται αυτόματα το τελευταίο.

Όταν το σκορ μίας πρότασης του χρήστη με όλα τα microservices είναι 10 τότε το bot στέλνει μήνυμα στον χρήστη ότι δεν κατανόησε αυτό που του ζητήθηκε.

Σε οποιαδήποτε άλλη περίπτωση, εμφανίζονται τα microservices ως επιλογές στον χρήστη, με σειρά από το μικρότερο στο μεγαλύτερο σκορ, ώστε αυτός να επιλέξει τι επιθυμεί να εκτελέσει.



Σχήμα 4.8: Εμφάνιση επιλογών με σκορ μικρότερο από 10

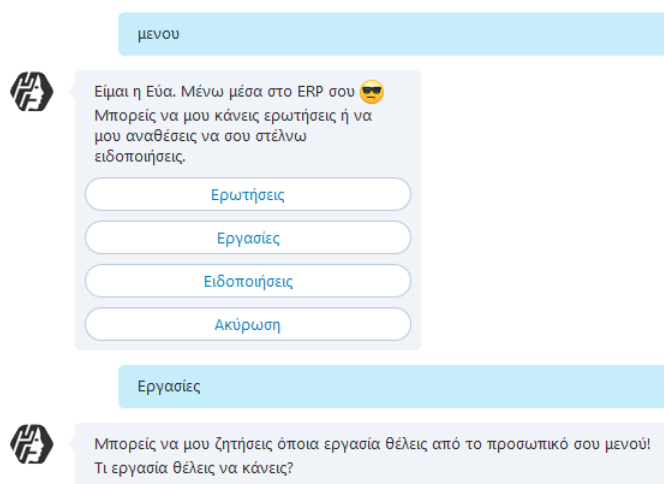
Στο παραπάνω παράδειγμα, ο χρήστης έχει πληκτρολογήσει τη λέξη “υπόλοιπο”. Η συγκεκριμένη λέξη έχει σκορ 10 με τους τίτλους όλων των microservices εκτός από αυτούς που φαίνονται στην εικόνα, οι οποίοι είναι “Υπόλοιπο Πελάτη”, “Υπόλοιπο Προμηθευτή” και “Υπόλοιπο Είδους”, με τους οποίους έχει σκορ 1. Σε αυτή την περίπτωση, στην οποία οι επιλογές έχουν όλες τον ίδιο βαθμό ομοιότητας με την φράση του χρήστη, αυτές εμφανίζονται με τη σειρά που είναι τοποθετημένες στον κώδικα.

4.4 Λειτουργικότητα

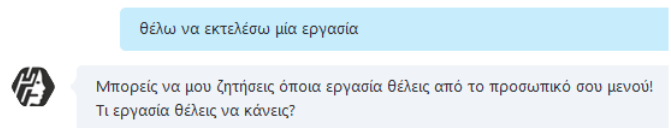
Σε αυτή την ενότητα θα αναλυθεί η συμπεριφορά του bot.

Όταν το SoftOne bot προστεθεί στη λίστα επαφών ενός χρήστη, ο τελευταίος έχει τη δυνατότητα να κάνει εγγραφή(register) σε αυτό με τα στοιχεία που χρησιμοποιεί στο Softone και έπειτα να του στείλει ένα μήνυμα.

Όπως έχουμε αναφέρει το bot έχει τρεις καταστάσεις στις οποίες είναι δυνατό να βρίσκεται, δηλαδή μπορεί να ανταποκριθεί σε τρία είδη λειτουργιών, τις ερωτήσεις, τις εργασίες και τις ειδοποιήσεις. Η προεπιλεγμένη κατάσταση στην οποία βρίσκεται είναι αυτή των ερωτήσεων. Για να βρεθεί σε οποιαδήποτε άλλη κατάσταση θα πρέπει αυτό να επιλεγεί από το μενού (το οποίο εμφανίζεται όταν πληκτρολογηθεί “μενού”) ή ο χρήστης να στείλει ένα μήνυμα στο οποίο να περιέχονται οι λέξεις “εργασία” ή “ειδοποίηση” σε οποιαδήποτε πτώση τους. Παρακάτω παρουσιάζεται ένα παράδειγμα για κάθε μία από τις δύο περιπτώσεις.



Σχήμα 4.9: Επιλογή απο το μενού



Σχήμα 4.10: Αποστολή μηνύματος

Όταν το bot βρίσκεται σε κατάσταση ερωτήσεων, αναζητούμε το βαθμό ομοιότητας της φράσης του χρήστη με τον τίτλο των υπάρχοντων microservices, με τον τρόπο που περιγράφηκε παραπάνω και απαντά αναλόγως.

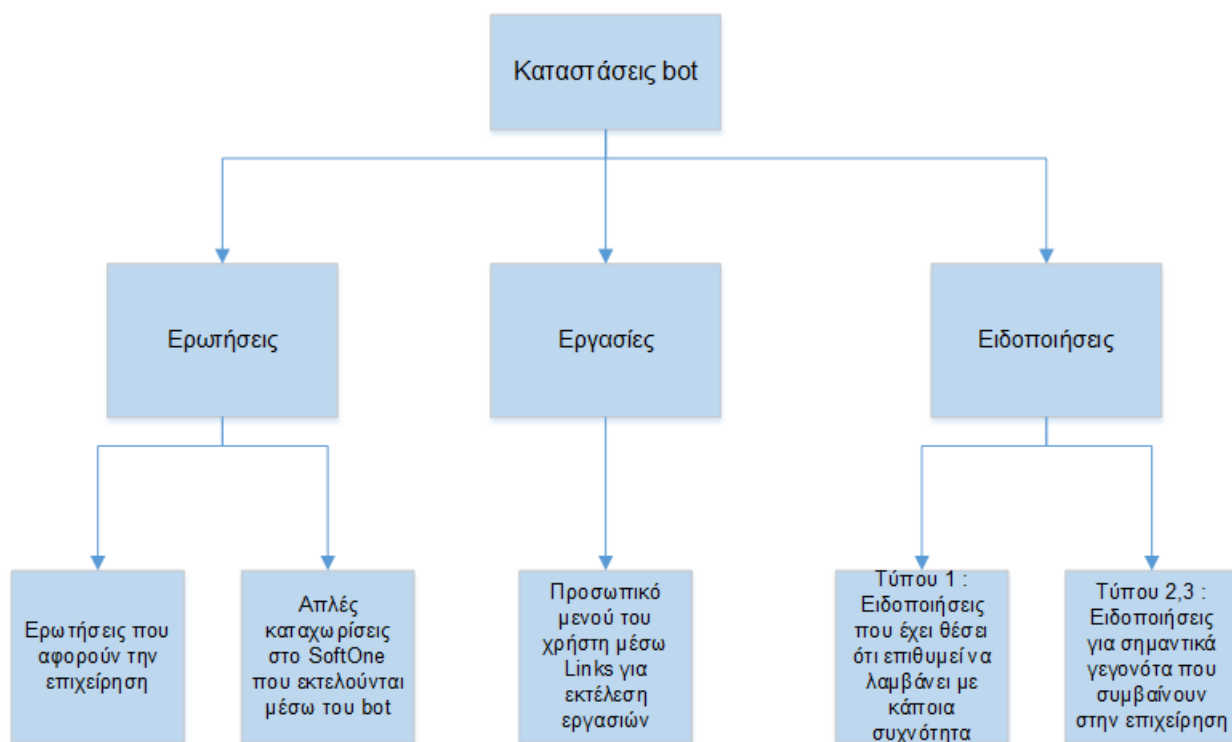
Είναι σημαντικό να αναφερθεί ότι τα microservices των ερωτήσεων είτε περιέχουν SQL queries τα οποία πρέπει να αποσταλούν στο SoftOne και να λάβουν μία απάντηση είτε, στην περίπτωση των καταχωρίσεων, ενημερώνουν τα πεδία που απαιτείται στο SoftOne μέσω Web Service¹. Για την αποστολή των queries στο SoftOne έχουμε δημιουργήσει τις συναρτήσεις sql2json και sql2array. Οι συναρτήσεις αυτές, επιτρέπουν την επικοινωνία με το SoftOne με COM (Component Object Model)² τεχνολογία και μετατρέπουν την απάντηση που λαμβάνεται σε Json Object και σε πίνακα αντίστοιχα ώστε να είναι πιο εύκολη η διαχείριση της.

¹Web Services είναι η τεχνολογία που επιτρέπει την διασύνδεση μεταξύ εφαρμογών, ανεξάρτητα από την πλατφόρμα και τις γλώσσες προγραμματισμού. Πρόκειται για μια διασύνδεση λογισμικού που ορίζει μια συλλογή λειτουργιών που μπορούν να αποκτήσουν πρόσβαση από τον ιστό μέσω τυποποιημένων μηνυμάτων[14].

²COM (Component Object Model) είναι ένα ανεξάρτητο από την πλατφόρμα, αντικειμενοστρεφές σύστημα για τη δημιουργία δυαδικών στοιχείων λογισμικού που μπορούν να αλληλεπιδρούν. Τα COM objects μπορούν να δημιουργηθούν με μια ποικιλία γλωσσών προγραμματισμού[15].

Όταν βρεθεί σε κατάσταση ειδοποιήσεων (που αφορά τις τύπου 1 ειδοποιήσεις), ακολουθείται ακριβώς η ίδια πορεία με εκείνη που ακολουθείται στην κατάσταση των ερωτήσεων με τη διαφορά ότι αντί να εκτελεστεί το microservice το οποίο έχει επιλεγεί, αποθηκεύεται το μοναδικό του κλειδί (id) καθώς και οι πληροφορίες (αν υπάρχουν) που έχουν δοθεί από τον χρήστη με σκοπό να καλεστεί και να εκτελεσθεί τη χρονική στιγμή που έχει οριστεί. Στις ειδοποιήσεις τύπου 2 και 3 ο χρήστης απλά λαμβάνει ένα μήνυμα από το SoftOne μέσω Skype. Αν στο συγκεκριμένο μήνυμα απαιτείται η έγκριση του χρήστη για να προχωρήσει κάποια διαδικασία, ενημερώνονται τα πεδία του SoftOne που απαιτούνται (μέσω Web Service) ανάλογα με την έγκριση ή μη από τον χρήστη.

Όταν το bot βρεθεί σε κατάσταση εργασίας, ο έλεγχος μεταφέρεται σε μία άλλη συνάρτηση. Αυτή οποία ψάχνει το βαθμό ομοιότητας της φράσης του χρήστη με τις επιλογές από το προσωπικό του μενού (οι οποίες έχουν ανακτηθεί σε ένα πίνακα), ώστε να του εμφανίσει, επίσης με τον τρόπο που περιγράφηκε, εκείνες με τις οποίες έχει ικανοποιητικό βαθμό ομοιότητας, ως υπερσυνδέσμους. Όταν εμφανιστούν οι επιλογές στο χρήστη και εκείνος εκτελέσει αυτή που επιθυμεί, το bot επιστρέφει ξανά σε κατάσταση ερωτήσεων. Το προσωπικό μενού του χρήστη αναχτάται μέσω Web Service.



Σχήμα 4.11: Δυνατές καταστάσεις bot

Θελω να μάθω τις εισπράξεις μου

Για

Ίδια ημέρα

Προηγούμενη μέρα

Τρέχουσα εβδομάδα

Τρέχοντα μήνα

Άλλη ημερομηνία

Τρέχοντα μήνα

Μετρητά : 0,00
Κάρτες : 224,00
Αξιόγραφα : 0,00
Εμβάσματα : 0,00

Σχήμα 4.12: Παράδειγμα λειτουργικότητας ερωτήσεων

ΠΕΛΑΤΕΣ

Κάνε κλικ για μετάβαση:

Παράπονα Πελατών

Υποψήφιοι Πελάτες

Πελάτες

Εισπράξεις πελατών

Ενηλ. Υπόλοιπα Πελατών

Σχήμα 4.13: Παράδειγμα λειτουργικότητας εργασιών

θέλω να λάβω μία ειδοποίηση

Κάθε πότε θέλεις ειδοποίηση?

Πρωί

Μεσημέρι

Απόγευμα

Βράδυ

Μεσημέρι

Πες μου τι ειδοποίηση θέλεις να λαμβάνεις

για τις εισπράξεις μου

Για

Ίδια ημέρα

Προηγούμενη μέρα

Τρέχουσα εβδομάδα

Τρέχοντα μήνα

Άλλη ημερομηνία

Τρέχοντα μήνα

ok 😊

Σχήμα 4.14: Παράδειγμα λειτουργικότητας ειδοποιήσεων

Κεφάλαιο 5

Αξιολόγηση

5.1 Εισαγωγή

Κατά τη διαδικασία της δημιουργίας ενός bot, ο δημιουργός επιθυμεί να προτιμάται η εμπειρία bot για την επίλυση του συγκεκριμένου προβλήματος από τους χρήστες έναντι εναλλακτικών εμπειριών όπως οι εφαρμογές και οι ιστότοποι. Με άλλα λόγια το bot ανταγωνίζεται τις εφαρμογές ή τους ιστοτόπους σε θέματα που αφορούν την πιο γρήγορη επίλυση του προβλήματος των χρηστών. Για να είναι το bot ανταγωνιστικό πρέπει να έχουν ληφθεί υπόψη οι σωστοί παράγοντες κατά τον σχεδιασμό του.

Το βασικό αντικείμενο του παρόντος κεφαλαίου είναι η παρουσίαση και η ανάλυση αυτών των παραγόντων καθώς και η αξιολόγηση της ευχρηστίας του bot από χρήστες, η οποία είναι βασισμένη σε συγκεκριμένο πρωτόκολλο το οποίο θα αναλυθεί.

5.2 Γενικοί παράγοντες επιτυχίας ενός bot

Παρακάτω παρουσιάζονται κάποιοι παράγοντες που καθιστούν το bot πιο εντυπωσιακό αλλά ταυτόχρονα δεν μπορούν να εγγυηθούν απαραίτητα την επιτυχία του[16].

- Το πόσο έξυπνο είναι. Στις περισσότερες περιπτώσεις είναι απίθανο ότι ένα πιο έξυπνο bot θα εγγυηθεί ευτυχισμένους χρήστες και υιοθέτηση της πλατφόρμας από αυτούς. Στην πραγματικότητα, πολλά bots έχουν μικρή προηγμένη μηχανική μάθηση ή δυνατότητες φυσικής γλώσσας. Φυσικά, ένα bot μπορεί να περιλαμβάνει αυτές τις δυνατότητες εάν είναι απαραίτητες για την επίλυση των προβλημάτων που έχει σχεδιαστεί για να αντιμετωπίσει. Ωστόσο, δεν πρέπει ο σχεδιαστής να θεωρεί ότι υπάρχει συσχέτιση μεταξύ της νοημοσύνης ενός bot και της υιοθέτησης του από τους χρήστες.
- Το πόση φυσική γλώσσα υποστηρίζει. Το bot μπορεί να είναι εξαιρετικό σε συνομιλίες. Μπορεί να έχει τεράστιο λεξιλόγιο και να κάνει εύστοχα αστεία. Αλλά αν δεν έχει την ικανότητα να αντιμετωπίσει τα προβλήματα των χρηστών, αυτές οι δυνατότητες μπορεί να συμβάλλουν πολύ λίγο στην επιτυχία του. Στην πραγματικότητα, μερικά bots δεν έχουν καθόλου δυνατότητες συνομιλίας όμως επιλύουν άριστα το πρόβλημα για το οποίο έχουν σχεδιαστεί.
- Η φωνή. Δεν είναι απαραίτητο ότι η ενεργοποίηση της ομιλίας σε ένα bot θα οδηγήσει σε πολύ καλές εμπειρίες χρήστη. Συχνά, αναγκάζοντας τους χρήστες να χρησιμοποιούν φωνή είναι πιθανή μια απογοητευτική εμπειρία χρήστη. Κατά τον σχεδιασμό του bot είναι απαραίτητο να εξετασθεί αν η φωνή είναι το κατάλληλο μέσο για να επιλυθεί το συγκεκριμένο πρόβλημα καθώς και να προσμετρηθεί ότι μπορεί να υπάρξει αλλοίωση του ήχου ή της πληροφορίας κατά την μετάδοση.

Υπάρχουν, όμως και κάποιοι παράγοντες που αν δεν ληφθούν υπόψη από τον σχεδιαστή, ο χρήστης δεν θα έχει καλή εμπειρία χρησιμοποιώντας το bot.

1. Το αν έχει τη δυνατότητα να επιλύσει τα προβλήματα των χρηστών με τον ελάχιστο αριθμό βημάτων.
2. Το αν μπορεί να λύσει το πρόβλημα του χρήστη καλύτερα / ευκολότερα / ταχύτερα από οποιαδήποτε άλλη εναλλακτική εμπειρία.
3. Το αν λειτουργεί στις συσκευές και τις πλατφόρμες που ενδιαφέρουν τον χρήστη, αν είναι εύκολα ανιχνεύσιμο και αν είναι εύκολο για τον ενδιαφερόμενο να το χρησιμοποιήσει.

Πρέπει να σημειωθεί ακόμη ότι μία καλή εμπειρία χρήστη bot δεν απαιτεί από τους χρήστες να πληκτρολογούν πάρα πολύ, να μιλάνε πάρα πολύ, να επαναλαμβάνουν αρκετές φορές ή να εξηγούν πράγματα που θα έπρεπε να γνωρίζει αυτόματα.

Η πρώτη αλληλεπίδραση μεταξύ του χρήστη και του bot είναι κρίσιμη για την εμπειρία του χρήστη. Κατά το σχεδιασμό του πρέπει ο σχεδιαστής να γνωρίζει ότι πρέπει να “καλωσορίσει” τον χρήστη με πιο έξυπνο τρόπο από ένα απλό “γεια”. Όπως όταν δημιουργείται μία εφαρμογή όπου η πρώτη οθόνη σχεδιάζεται για να δώσει σημαντικά σημάδια πλοήγησης, έτσι και στο bot η πρώτη επαφή θα πρέπει να βοηθά το χρήστη να κατανοήσει κάποια βασικά πράγματα, όπως το που βρίσκεται το μενού και πως λειτουργεί, τι πρέπει να πληκτρολογήσει για βοήθεια κ.α.

Η σημαντικότητα του μενού :

Δεν συνιστάται να ξεκινάει η εμπειρία με το bot με μία ανοιχτή ερώτηση όπως “Πως μπορώ να σας βοηθήσω;”. Εάν το bot έχει πολλές διαφορετικές λειτουργίες, το πιθανότερο είναι ότι οι χρήστες δεν θα είναι σε θέση να γνωρίζουν τις περισσότερες από αυτές χωρίς κάποια ενημέρωση. Τα μενού παρέχουν μια απλή λύση σε αυτό το πρόβλημα. Πρώτον, καταγράφοντας τις διαθέσιμες επιλογές, το bot μεταφέρει τις δυνατότητές του στους χρήστες. Ακόμη, τα μενού απαλλάσσουν τους τελευταίους από το να πληκτρολογήσουν πάρα πολύ. Μπορούν απλά να κάνουν κλικ. Τέλος, η χρήση μενού μπορεί να απλοποιήσει σημαντικά τα μοντέλα φυσικής γλώσσας, μειώνοντας το εύρος των εισροών που θα μπορούσε να λάβει το bot από τους χρήστες[17].

Design 1



Hello user, how can I help you?

Design 2



Hello! How can I help you?

Orders

Products

Help

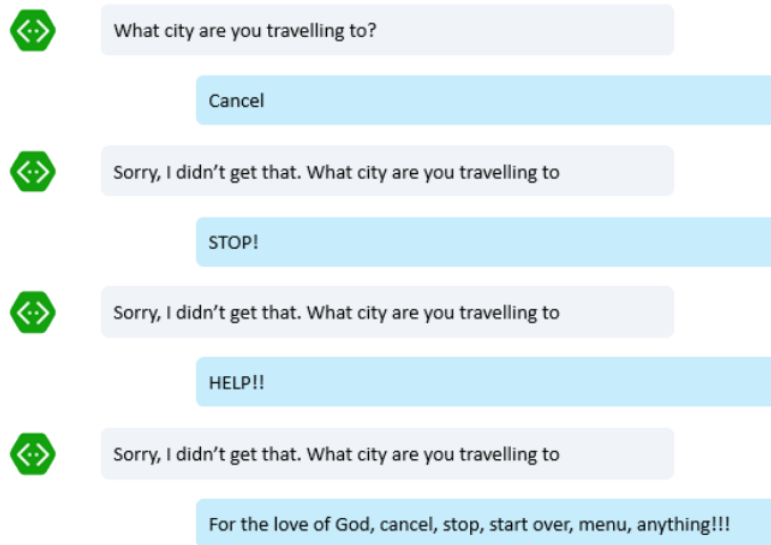
Σχήμα 5.1: Παράδειγμα χωρίς μενού και με μενού[17]

5.3 Πιθανά προβλήματα σε bots

Οι ιδιαιτερότητες του σχεδίου πλοήγησης του κάθε bot εξαρτώνται σε μεγάλο βαθμό από τα χαρακτηριστικά και τη λειτουργικότητα που υποστηρίζει. Ωστόσο, ανεξάρτητα από το είδος του bot είναι σημαντικό να αποφεύγονται οι κοινές παγίδες των κακώς σχεδιασμένων συνομιλητικών διεπαφών. Παρακάτω, θα περιγράψουμε αυτές τις παγίδες σε πέντε προσωπικότητες: το “επίμονο bot”, το “clueless bot”, το “μυστηριώδες bot”, το “προφανές bot” και το “bot που δεν μπορεί να ξεχάσει” [18].

Το επίμονο bot

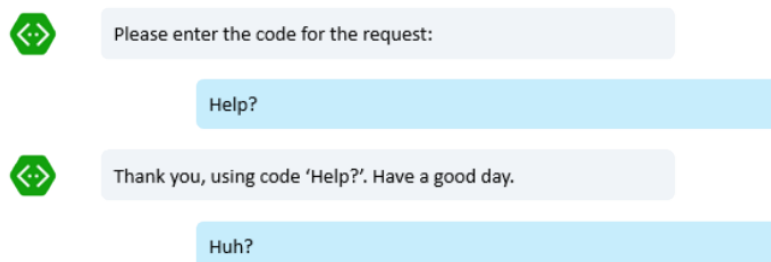
Το επίμονο bot επιμένει στην διατήρηση της τρέχουσας πορείας της συζήτησης, ακόμα και όταν ο χρήστης επιχειρεί να κατευθύνει τα πράγματα σε διαφορετική κατεύθυνση.



Σχήμα 5.2: Παράδειγμα επίμονου bot[19]

Το clueless bot

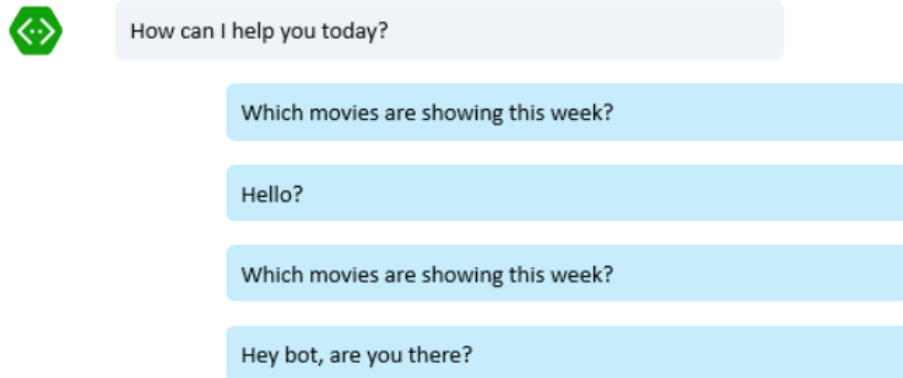
Το clueless bot ανταποκρίνεται με ανόητο τρόπο όταν δεν κατανοεί την προσπάθεια ενός χρήστη να αποκτήσει πρόσβαση σε ορισμένες λειτουργίες. Ένας χρήστης μπορεί να δοκιμάσει κοινές εντολές λέξεων-κλειδιών όπως “βοήθεια” ή “ακύρωση” με εύλογες προσδοκίες ότι το bot θα ανταποκριθεί κατάλληλα.



Σχήμα 5.3: Παράδειγμα clueless bot[18]

Το μυστηριώδες bot

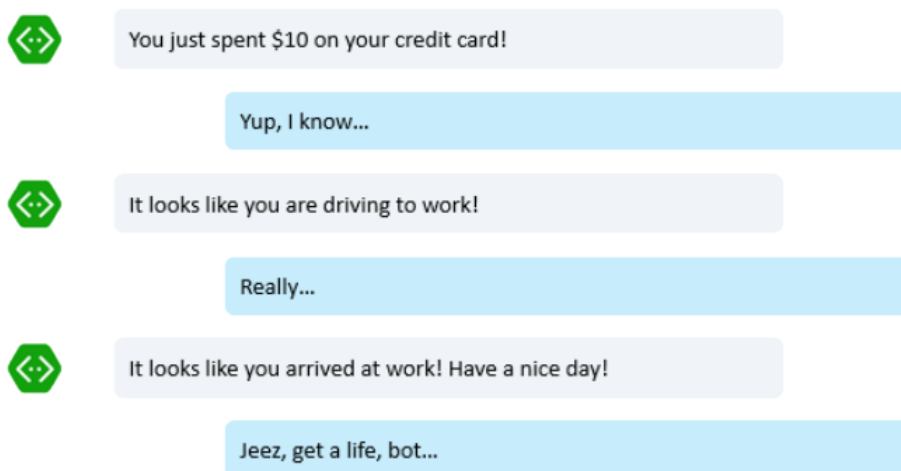
Το μυστηριώδες bot αποτυγχάνει να αναγνωρίσει αμέσως την είσοδο του χρήστη με οποιονδήποτε τρόπο.



Σχήμα 5.4: Παράδειγμα μυστηριώδους bot[18]

Το προφανές bot

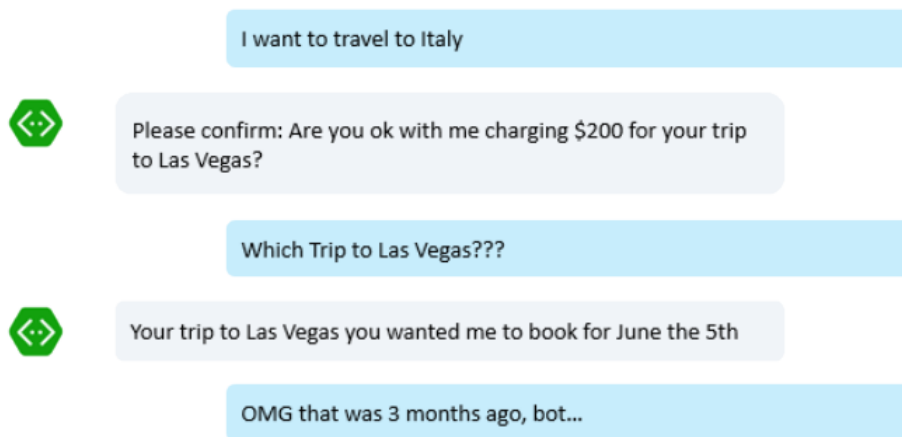
Το προφανές bot παρέχει ανεπιθύμητες πληροφορίες που είναι απολύτως προφανείς και ως εκ τούτου άχρηστες για τον χρήστη.



Σχήμα 5.5: Παράδειγμα προφανούς bot[18]

Το bot που δεν μπορεί να ξεχάσει

Το bot που δεν μπορεί να ξεχάσει ενσωματώνει ακατάλληλα πληροφορίες από προηγούμενες συνομιλίες στην τρέχουσα συζήτηση.



Σχήμα 5.6: Παράδειγμα bot που δεν μπορεί να ξεχάσει[18]

Με σκοπό να αποφύγουμε κάποια από τα παραπάνω προβλήματα έχουμε ορίσει τις εντολές stop και back. Όταν ένας χρήστης βρίσκεται μέσα σε μία συζήτηση με το bot και επιθυμεί να μεταβεί στην προηγούμενη ερώτηση για οποιοδήποτε λόγο, μπορεί να πληκτρολογήσει -back και να μεταβεί σε αυτή. Ακόμη, όταν ο χρήστης θέλει να διακόψει στη μέση τη συζήτηση για να ρωτήσει κάτι καινούριο έχει τη δυνατότητα να πληκτρολογήσει -stop, η συζήτηση διακόπτεται και ο τελευταίος μπορεί να ζητήσει κάτι νέο από το bot.

5.4 Γενικά χαρακτηριστικά αξιολόγησης ευχρηστίας

Μια μέτρηση είναι ένα “σύστημα ή πρότυπο μέτρησης” που αντιπροσωπεύεται σε μονάδες που μπορούν να χρησιμοποιηθούν για να περιγράψουν περισσότερα από ένα χαρακτηριστικά. Οι μετρήσεις είναι πολύ χρήσιμες όταν πρόκειται να ποσοτικοποιήσουν την ευχρηστία κατά τη διάρκεια της αξιολόγησης της ευχρηστίας του λογισμικού, των ιστοτόπων και των εφαρμογών.

Υπάρχουν πολλοί λόγοι για τους οποίους κάποιος θα χρειαζόταν να μετρήσει την ευχρηστία. Ο συνηθέστερος λόγος είναι η ανάγκη αποτελεσματικής επικοινωνίας με τους ενδιαφερόμενους του συστήματος που αξιολογείται. Άλλες χρήσεις μπορούν να είναι η ικανοποίηση της ανάγκης σύγκρισης της ευχρηστίας δύο ή περισσότερων προϊόντων και η ποσοτικοποίηση της σοβαρότητας ενός προβλήματος ευχρηστίας.

Το πρότυπο ISO 9241-11 (Διεθνής Οργανισμός Τυποποίησης) ορίζει τη δυνατότητα χρήσης ως “το βαθμό στον οποίο ένα προϊόν μπορεί να χρησιμοποιηθεί από συγκεκριμένους χρήστες για την επίτευξη συγκεκριμένων στόχων με αποτελεσματικότητα, αποδοτικότητα και ικανοποίηση σε συγκεκριμένο πλαίσιο χρήσης”.

Ο ορισμός αυτός δηλώνει σαφώς ότι η ευχρηστία δεν είναι μια ενιαία, μονοδιάστατη ιδιότητα, αλλά ένας συνδυασμός παραγόντων[19].

Οι μετρήσεις ISO / IEC 9126-4 συνιστούν ότι οι μετρήσεις ευχρηστίας πρέπει να περιλαμβάνουν[19]:

- Αποτελεσματικότητα: Η ακρίβεια και η πληρότητα με την οποία οι χρήστες επιτυγχάνουν καθορισμένους στόχους.
- Αποδοτικότητα: Οι πόροι που δαπανώνται σε σχέση με την ακρίβεια και την πληρότητα με την οποία οι χρήστες επιτυγχάνουν στόχους.
- Ικανοποίηση: Η άνεση και η αποδοχή της χρήσης.

Ωστόσο, οι πραγματικοί τρόποι με τους οποίους θα πρέπει να μετρηθούν αυτά, παραμένουν πολύ συχνά στη διακριτική ευχέρεια του αξιολογητή.

Μέτρηση ευχρηστίας που αφορά την αποτελεσματικότητα[19]:

1. Ποσοστό ολοκλήρωσης

Η αποτελεσματικότητα μπορεί να υπολογιστεί με μέτρηση του ποσοστού ολοκλήρωσης. Αναφερόμενη ως η θεμελιώδης μέτρηση ευχρηστίας, το ποσοστό ολοκλήρωσης υπολογίζεται με την ανάθεση δυαδικής τιμής «1» αν ο συμμετέχων στη δοκιμή καταφέρει να ολοκληρώσει μια εργασία και το «0» αν δεν το κάνει.

Λόγω της απλότητας του, το ποσοστό ολοκλήρωσης είναι μια μέτρηση που είναι πολύ εύκολο να κατανοηθεί, εξ ου και ο λόγος για τον οποίο είναι πολύ δημοφιλής. Επιπλέον, μπορεί να συλλεχθεί σε οποιοδήποτε στάδιο ανάπτυξης. Επομένως, η αποτελεσματικότητα μπορεί να αναπαρασταθεί ως ποσοστό χρησιμοποιώντας αυτή την απλή εξίσωση:

$$Effectiveness = \frac{\text{Number of tasks completed successfully}}{\text{Total number of tasks undertaken}} \times 100\%$$

Παρόλο που κάποιος πρέπει πάντα να επιδιώκει ποσοστό ολοκλήρωσης 100%, σύμφωνα

με μια μελέτη του Jeff Sauro¹, ο μέσος ρυθμός ολοκλήρωσης εργασιών είναι 78% (βάσει ανάλυσης 1.100 εργασιών). Στην ίδια μελέτη, παρατηρήθηκε επίσης ότι το ποσοστό ολοκλήρωσης εξαρτάται σε μεγάλο βαθμό από το πλαίσιο της αξιολογούμενης εργασίας.

2. Αριθμός λαθών

Μια άλλη μέτρηση περιλαμβάνει την καταμέτρηση του αριθμού των σφαλμάτων που κάνει ο συμμετέχων όταν προσπαθεί να ολοκληρώσει μια εργασία. Τα σφάλματα μπορεί να είναι ακούσιες ενέργειες, λάθη ή παραλείψεις που κάνει κάποιος χρήστης όταν επιχειρεί μια εργασία. Θα πρέπει να ορισθεί μια σύντομη περιγραφή, μια βαθμολογία σοβαρότητας και να ταξινομηθεί κάθε λάθος στην αντίστοιχη κατηγορία. Αν και μπορεί να είναι χρονοβόρα, η μέτρηση του αριθμού σφαλμάτων παρέχει εξαιρετικές διαγνωστικές πληροφορίες.

Βάσει μιας ανάλυσης 719 εργασιών που εκτελούνται με χρήση λογισμικού καταναλωτών και επιχειρήσεων, ο Jeff Sauro κατέληξε στο συμπέρασμα ότι ο μέσος αριθμός σφαλμάτων ανά εργασία είναι 0,7, με 2 στους 3 χρήστες να κάνουν λάθος. Μόνο το 10% των παρατηρημένων εργασιών εκτελέστηκε χωρίς λάθη, οδηγώντας έτσι στο συμπέρασμα ότι είναι απολύτως φυσιολογικό για τους χρήστες να κάνουν λάθη κατά την εκτέλεση εργασιών.

Μέτρηση ευχρηστίας που αφορά την αποδοτικότητα:

Μια καθιερωμένη μέθοδος για τη δοκιμή ενός συστήματος με τη συμμετοχή χρηστών είναι η μέτρηση της απόδοσης, όπου σκοπός είναι η πιστοποίηση για το αν ένας στόχος ευχρηστίας επιτυγχάνεται ή όχι. Η απόδοση των χρηστών μετράται συνήθως υποβάλλοντας μια ομάδα χρηστών σε εκτέλεση ενός προκαθορισμένου συνόλου στοιχειωδών εργασιών, συλλέγοντας στοιχεία που αφορούν σε σφάλματα που διαπιστώνονται, στους χρόνους εκτέλεσης των εργασιών, στον αριθμό ενεργειών κ.ά. Σε αυτή τη μέθοδο, παρουσιάζονται δυσκολίες στη δειγματοληψία και επιλογή των “αντιπροσωπευτικών” χρηστών – ώστε συνολικά να παρουσιάζουν παρόμοια χαρακτηριστικά με τους αναμενόμενους τελικούς χρήστες- αλλά και μεθοδολογικές αδυναμίες στον προγραμματισμό, την εγκυρότητα και την αξιοπιστία των αποκτηθέντων μετρήσεων. Τέλος, η αξιολόγηση της αποδοτικότητας σε καθημερινή χρήση απαιτεί την ανεύρεση χρηστών, που να ανταποκρίνονται πλήρως στα πραγματικά χαρακτηριστικά των ανθρώπων που χρησιμοποιούν το σύστημα.[20]

Η απόδοση μετριέται σε όρους χρόνου εργασίας. Δηλαδή, ο χρόνος (σε δευτερόλεπτα και / ή λεπτά) που χρειάζεται ο συμμετέχων για να ολοκληρώσει επιτυχώς μια εργασία. Ο χρόνος που απαιτείται για την ολοκλήρωση μιας εργασίας μπορεί στη συνέχεια να υπολογιστεί με απλή αφαίρεση του χρόνου έναρξης από την ώρα λήξης όπως φαίνεται στην παρακάτω εξίσωση:[19]

$$\text{Task Time} = \text{End Time} - \text{Start Time}$$

¹Ο Jeff Sauro είναι ο ιδρυτής της εταιρείας MeasuringU, η οποία παρέχει στατιστικές και συμβουλευτικές υπηρεσίες χρηστικότητας. Είναι πρωτοπόρος στην ποσοτικοποίηση της εμπειρίας των χρηστών και έχει συγγράψει πάνω από 15 επιστημονικά άρθρα και 5 βιβλία σχετικά με τις στατιστικές και την εμπειρία του χρήστη.

Η αποδοτικότητα μπορεί να μετρηθεί με δύο τρόπους:

1. Χρονική απόδοση

$$\text{Χρονική Απόδοση} = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR}$$

Όπου:

N = Ο συνολικός αριθμός εργασιών (στόχοι)

R = Ο αριθμός των χρηστών

n_{ij} = Το αποτέλεσμα της εργασίας i από το χρήστη j, εάν ο χρήστης ολοκληρώσει με επιτυχία την εργασία, τότε $N_{ij} = 1$, αν όχι, τότε $N_{ij} = 0$

t_{ij} = Ο χρόνος που δαπανά ο χρήστης j για να ολοκληρώσει την εργασία i. Εάν η εργασία δεν ολοκληρωθεί με επιτυχία, τότε μετράται η ώρα μέχρι τη στιγμή που ο χρήστης τερματίσει την εργασία.

2. Συνολική σχετική απόδοση

Η συνολική σχετική απόδοση χρησιμοποιεί τον λόγο του χρόνου που χρειάζονται οι χρήστες που ολοκλήρωσαν με επιτυχία την εργασία σε σχέση με το συνολικό χρόνο που έχει ληφθεί από όλους τους χρήστες. Η εξίσωση μπορεί έτσι να αναπαρασταθεί ως εξής:

$$\text{Συνολική Σχετική Απόδοση} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100\%$$

Μέτρηση ευχρηστίας που αφορά την ικανοποίηση:

Η ικανοποίηση των χρηστών μετριέται μέσω τυποποιημένων ερωτηματολογίων ικανοποίησης. Τα ερωτηματολόγια αποτελούν μια χρήσιμη προσέγγιση, όταν ο στόχος είναι να διερευνηθούν ζητήματα σχετικά με πιθανές ανησυχίες-παρανοήσεις των χρηστών και την υποκειμενική τους ικανοποίηση. Τα ερωτηματολόγια μπορούν να διανεμηθούν εύκολα σε σημαντικό αριθμό χρηστών. Επιπλέον, αποτελούν μια σχετικά ανέξοδη μέθοδο ερευνών. Εντούτοις, είναι συχνά δύσκολο να συγκεντρωθούν αντικειμενικά αποτελέσματα κατά την αξιοποίηση των ερωτηματολογίων, δεδομένου ότι οι απαντήσεις των χρηστών είναι βασισμένες σε αυτό που έχουν την αίσθηση ότι κάνουν και όχι σε αυτό που κάνουν πραγματικά[20].

1. Ικανοποίηση επιπέδου εργασίας[19]

Αφού οι χρήστες επιχειρήσουν μια εργασία (ανεξάρτητα από το αν καταφέρνουν να επιτύχουν το στόχο τους ή όχι), θα πρέπει να τους δοθεί άμεσα ένα ερωτηματολόγιο ώστε να μετρηθεί η δυσκολία που έχει το καθήκον αυτό. Συνήθως αποτελούμενα από 1 έως και 5 ερωτήσεις, τα ερωτηματολόγια αυτά έχουν συχνά τη μορφή βαθμολόγησης

Likert² και ο στόχος τους είναι να παρέχουν μια διορατικότητα στη δυσκολία της εργασίας όπως τη βλέπουν οι συμμετέχοντες.

2. Ικανοποίηση επιπέδου δοκιμής[19]

Η ικανοποιητική ικανοποίηση του επιπέδου δοκιμής μετράται με την παροχή ενός τυποποιημένου ερωτηματολογίου δοκιμής σε κάθε συμμετέχοντα στο τέλος της περιόδου δοκιμής. Αυτό χρησιμεύει στη μέτρηση της εντυπώσεως τους σχετικά με τη συνολική ευκολία χρήσης του συστήματος που δοκιμάζεται.

Το ερωτηματολόγιο που χρησιμοποιήθηκε για την αξιολόγηση στην παρούσα εργασία είναι το USE. USE (Usefulness, Satisfaction, Ease of use) σημαίνει χρησιμότητα, ικανοποίηση και ευκολία χρήσης. Αυτές είναι οι τρεις διαστάσεις που εμφανίζονται πιο έντονα στην πρόωμη ανάπτυξη του ερωτηματολογίου αυτού.

Το USE είναι ένα σύντομο ερωτηματολόγιο που θα μπορούσε να χρησιμοποιηθεί για τη μέτρηση των σημαντικότερων διαστάσεων ευχρηστίας για τους χρήστες και για τη μέτρηση αυτών των διαστάσεων σε όλους τους τομείς. Επιτρέπει ουσιαστικές συγκρίσεις προϊόντων σε διάφορους τομείς, παρόλο που οι δοκιμές των προϊόντων μπορεί να έχουν συμβεί σε διαφορετικές χρονικές στιγμές και ίσως υπό διαφορετικές συνθήκες.

Το ερωτηματολόγιο κατασκευάστηκε ως κλίμακα βαθμολόγησης Likert επτά σημείων. Ζητήθηκε από τους χρήστες να αξιολογήσουν τη συμφωνία με τις δηλώσεις, που κυμαίνεται από την απόλυτη διαφωνία έως την απόλυτη συμφωνία. Ο αριθμός 1 αντιστοιχεί σε «διαφωνώ απόλυτα», 2=«διαφωνώ», 3=«διαφωνώ λίγο», 4=«ουδέτερος/η», 5=«συμφωνώ λίγο», 6=«συμφωνώ», 7=«συμφωνώ απόλυτα»[19].

Οι ερωτήσεις που περιλαμβάνονται στο συγκεκριμένο ερωτηματολόγιο παρουσιάζονται παρακάτω[22].

Χρησιμότητα

- Με βοηθά να γίνω πιο αποτελεσματικός.
- Με βοηθά να γίνω πιο παραγωγικός.
- Είναι χρήσιμο.
- Μου δίνει μεγαλύτερο έλεγχο στις καθημερινές μου εργασίες.
- Κάνει πιο εύκολες πολλές πράξεις που θέλω να διεκπεραιώσω.
- Μου εξοικονομεί χρόνο όταν το χρησιμοποιώ.
- Καλύπτει τις ανάγκες μου.
- Λειτουργεί όπως θα περίμενα να λειτουργεί.

²Η κλίμακα Likert είναι μια κλίμακα ψυχομετρίας που συνήθως εμπλέκεται στην έρευνα που απασχολεί ερωτηματολόγια. Είναι η πιο ευρέως χρησιμοποιούμενη προσέγγιση για την κλιμάκωση των απαντήσεων στα ερωτηματολόγια έρευνας και έτσι ο όρος (ή ακριβέστερα η κλίμακα τύπου Likert) χρησιμοποιείται συχνά εναλλακτικά με την κλίμακα αξιολόγησης, παρόλο που υπάρχουν και άλλοι τύποι κλιμάκων αξιολόγησης[21].

Ευκολία στη χρήση

- Είναι εύκολο στη χρήση του.
- Είναι απλό στη χρήση του.
- Είναι φιλικό προς το χρήστη.
- Απαιτεί τα λιγότερα βήματα για την επίτευξη των εργασιών που θέλω να κάνω με αυτό.
- Είναι ευέλικτο.
- Η χρήση του δεν απαιτεί ιδιαίτερη προσπάθεια.
- Μπορώ να το χρησιμοποιήσω χωρίς γραπτές οδηγίες.
- Δεν παρατηρώ ασυνέπειες καθώς το χρησιμοποιώ.
- Θεωρώ ότι θα αρέσει τόσο σε περιστασιακούς όσο και σε τακτικούς χρήστες.
- Μπορώ να επανορθώσω λάθος χειρισμούς εύκολα και γρήγορα.
- Μπορώ να το χρησιμοποιήσω κάθε στιγμή.

Ευκολία στη μάθηση

- Έμαθα να το χρησιμοποιώ γρήγορα.
- Μπορώ να θυμηθώ εύκολα την λειτουργία του.
- Είναι εύκολο να μάθει κάποιος να το χρησιμοποιεί.
- Έγινα γρήγορα επιδέξιος χρήστης του.

Ικανοποίηση

- Είμαι ικανοποιημένος από την χρήση του.
- Θα το συνιστούσα σε έναν φίλο.
- Η χρήση του είναι διασκεδαστική.
- Λειτουργεί όπως ακριβώς θέλω να λειτουργεί.
- Είναι υπέροχο.
- Αισθάνομαι ότι το χρειάζομαι.
- Η χρήση του είναι ευχάριστη.

Για κάθε μία από τις διαστάσεις του ερωτηματολογίου χρησιμοποιούμε τον παρακάτω τύπο ώστε να υπολογίσουμε την επί τοις εκατό ικανοποίηση των χρηστών για αυτή τη διάσταση.

$$\text{Ικανοποίηση} = \sum_{j=1}^R \sum_{i=1}^Q \frac{p_{ij}}{Q * R} * 100\%$$

Όπου :

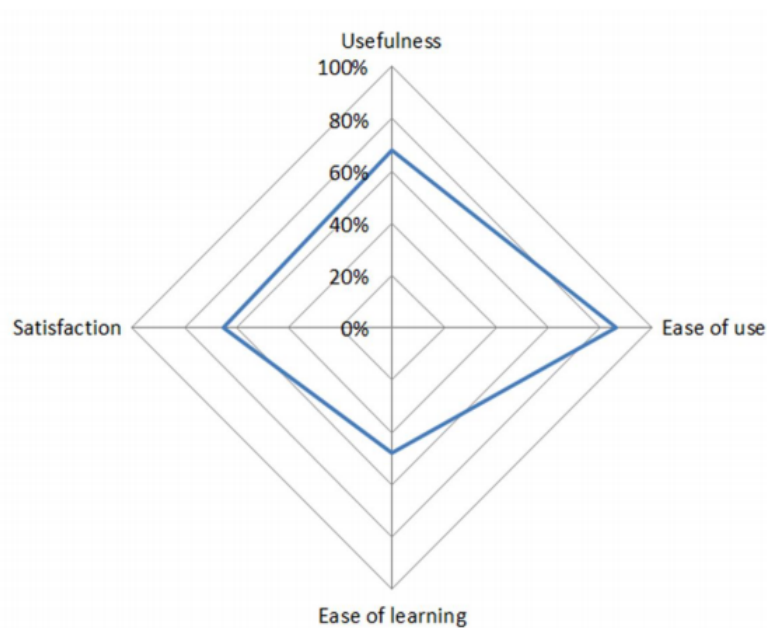
R = Ο αριθμός των χρηστών

Q = Ο αριθμός των ερωτήσεων

p_{ij} = Το βάρος της απάντησης του χρήστη j για την πρόταση i

P = Το βάρος της πρότασης

Η απεικόνιση των αποτελεσμάτων του ερωτηματολογίου USE μπορεί να γίνει περιγραφικά με διαγράμματα ραντάρ. Θα πρέπει προηγουμένως να έχει υπολογισθεί το μέσο αποτέλεσμα για κάθε ερώτηση, καθώς και το μέσο αποτέλεσμα για κάθε διάσταση. Με τα διαγράμματα ραντάρ δίνεται μια αφ' υψηλού εικόνα των διαστάσεων που καθορίζουν την ευχρηστία, καθώς και ενδείξεις επί των κατευθύνσεων που πρέπει να ακολουθηθούν για βελτιώσεις και διορθώσεις[34].



Σχήμα 5.7: Ενδεικτικό διάγραμμα ραντάρ για το ερωτηματολόγιο USE[23].

5.5 Αξιολόγηση ευχρηστίας του bot

5.5.1 Διαδικασία που ακολουθήθηκε

Με βάση όσα αναλύθηκαν στην προηγούμενη ενότητα οι άξονες της αξιολόγησης για το bot ήταν η αποτελεσματικότητα, η αποδοτικότητα και η ικανοποίηση των χρηστών σε ότι αφορά αυτό.

Το bot δόθηκε για δοκιμή και αξιολόγηση σε 10 χρήστες. Οι χρήστες αποτελούσαν εργαζόμενους στην εταιρεία SoftOne, γεγονός που διασφαλίζει ότι γνώριζαν ήδη πως να χρησιμοποιήσουν το SoftOne.

Αρχικά, τους δόθηκαν κάποιες οδηγίες που αφορούν τη χρήση του.

Έπειτα, επιλέχθηκαν κάποια σενάρια χρήσης τα οποία θα έπρεπε οι ίδιοι να ολοκληρώσουν με δύο τρόπους, με χρήση του bot και με χρήση του SoftOne, και να καταγράψουν τον χρόνο εκτέλεσής τους και στις δύο αυτές περιπτώσεις. Με αυτό τον τρόπο επιδιώκουμε να μετρήσουμε την αποτελεσματικότητα, δηλαδή αν θα καταφέρει ένα σημαντικό ποσοστό των χρηστών να ολοκληρώσει την εργασία, και την αποδοτικότητα, δηλαδή πόσος χρόνος χρειάστηκε για την εκτέλεση της εργασίας μέσω του bot συγκριτικά με το χρόνο που χρειάστηκε μέσω του Softone.

Στη συνέχεια, έκαναν χρήση του bot για άλλες δύο ημέρες χρησιμοποιώντας δικά τους σενάρια.

Τέλος, τους δόθηκε να συμπληρώσουν το ερωτηματολόγιο USE, το οποίο αναλύθηκε παραπάνω. Η συμπλήρωση του ερωτηματολογίου ζητήθηκε να πραγματοποιηθεί μία φορά για τη χρήση του SoftOne χωρίς χρήση του ψηφιακού βοηθού (bot) και μία για τη χρήση του συστήματος με χρήση του bot. Αυτό έγινε με σκοπό να καταγράψουμε την ικανοποίηση των χρηστών όταν χρησιμοποιούν το σύστημα χωρίς το bot σε σύγκριση με όταν το χρησιμοποιούν με την συμβολή του επιχειρησιακού βοηθού σε μορφή bot. Με το συγκεκριμένο ερωτηματολόγιο επιδιώκουμε να μετρήσουμε την ικανοποίηση των χρηστών σε ότι αφορά το bot.

5.5.2 Σενάριο χρήσης

Όπως αναφέρθηκε παραπάνω δόθηκε στους χρήστες ένα σενάριο χρήσης το οποίο έπρεπε να υλοποιήσουν με χρήση του bot και με χρήση του SoftOne και να καταγράψουν αν κατάφεραν να το εκτελέσουν σε κάθε μία από τις περιπτώσεις και τον χρόνο εκτέλεσής τους. Πρέπει να σημειωθεί ότι όλοι οι χρήστες ήταν συνδεδεμένοι στην ίδια βάση δεδομένων, δηλαδή είχαν τη δυνατότητα να δουν τα ίδια στοιχεία. Οι χρήστες ήταν εργαζόμενοι στην εταιρεία Softone, κάτι που εξασφαλίζει βασική γνώση χρήσης του συγκεκριμένου λογισμικού, όμως δεν εξασφαλίζει απαραίτητα ότι θα γνωρίζουν κάτι παραπάνω από τις πολύ βασικές διαδικασίες. Ακόμη, στους χρήστες δεν δόθηκαν οδηγίες για τον τρόπο που πρέπει να κάνουν τις ερωτήσεις ώστε να ελεγχθεί αν το bot ανταποκρίνεται στον τρόπο που εκφράζεται ο τυπικός χρήστης. Τους επισημάνθηκε μόνο ότι έχουν τη δυνατότητα να μεταβούν σε προηγούμενη ερώτηση γράφοντας -back καθώς και να διακόψουν τη ροή της συζήτησης και να ρωτήσουν κάτι νέο γράφοντας -stop. Παρακάτω φαίνεται το σενάριο χρήσης.

Σενάριο για bot(EVA)

Παρακαλώ να εκτελέσετε το σενάριο ξεχωριστά στην Eva και στο Softone και να καταγράψετε αν καταφέρατε να εκτελέσετε την εργασία και σε πόσο χρόνο.

Όνομα :

Ερώτηση 1

Βρείτε τους πελάτες με τον μεγαλύτερο τζίρο(τους καλύτερους πελάτες της εταιρείας).

Επιτυχία εκτέλεσης με bot :

Χρόνος εκτέλεσης με bot :

Επιτυχία εκτέλεσης με Softone :

Χρόνος εκτέλεσης με Softone :

Ερώτηση 2

Βρείτε τις εισπράξεις που είχε η εταιρεία στις 6 Φεβρουαρίου 2018.

Επιτυχία εκτέλεσης με bot :

Χρόνος εκτέλεσης με bot :

Επιτυχία εκτέλεσης με Softone :

Χρόνος εκτέλεσης με Softone :

Ερώτηση 3

Βρείτε το υπόλοιπο που έχει η εταιρεία στο είδος «φακοί χρωματιστοί»

Επιτυχία εκτέλεσης με bot :

Χρόνος εκτέλεσης με bot :

Επιτυχία εκτέλεσης με Softone :

Χρόνος εκτέλεσης με Softone :

Ερώτηση 4

Καταχωρήστε μία παραγγελία για το είδος **φακοί χρωματιστοί**, για τον πελάτη **Apple AE** και στην ποσότητα που επιθυμείτε.

Επιτυχία εκτέλεσης με bot :

Χρόνος εκτέλεσης με bot :

Επιτυχία εκτέλεσης με Softone :

Χρόνος εκτέλεσης με Softone :

Ερώτηση 5

Καταχώρηση ειδοποίησης για ενημέρωση στις 6 το απόγευμα για την εικόνα της εταιρείας για την ίδια ημέρα.

Επιτυχία εκτέλεσης με bot :

Χρόνος εκτέλεσης με bot :

Επιτυχία εκτέλεσης με Softone :

Χρόνος εκτέλεσης με Softone :

Σχήμα 5.8: Σενάριο χρήσης bot.

5.5.3 Αποτελέσματα/Ευρήματα Αξιολόγησης

Αφού συμπληρώθηκαν το σενάριο χρήσης και το ερωτηματολόγιο από τους χρήστες, ήμασταν σε θέση να εξάγουμε κάποια συμπεράσματα σε ότι αφορά την αποτελεσματικότητα, την αποδοτικότητα και την ικανοποίηση των χρηστών ως προς το συγκεκριμένο προϊόν. Τα αποτελέσματα από το σενάριο χρήσης είναι τα εξής :

Ερώτηση 1

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του bot : 10

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του Softone : 10

Ερώτηση 2

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του bot : 10

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του Softone : 10

Ερώτηση 3

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του bot : 10

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του Softone : 10

Ερώτηση 4

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του bot : 2

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του Softone : 10

Ερώτηση 5

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του bot : 10

Αριθμός χρηστών που κατάφεραν να ολοκληρώσουν την εργασία με χρήση του Softone : 0

Οι χρόνοι εκτέλεσης των εργασιών δεν καταγράφονται αναλυτικά λόγω του όγκου των χρηστών όμως έχουν ληφθεί υπόψη για τους υπολογισμούς όλων των απαιτούμενων τιμών.

Αποτελεσματικότητα

Με βάση τα παραπάνω και τον τύπο που παραθέσαμε στην προηγούμενη ενότητα υπολογίσθηκε ότι η αποτελεσματικότητα του επιχειρησιακού βοηθού (bot) είναι 84%, ενώ η αποτελεσματικότητα του SoftOne σε ότι αφορά το συγκεκριμένο σενάριο είναι 80%. Δηλαδή οι χρήστες κατάφεραν να ολοκληρώσουν 4% επιπλέον εργασίες μέσω του επιχειρησιακού βοηθού από τις εργασίες που περιείχε το σενάριο χρήσης.

Αποδοτικότητα

Με βάση τα αποτελέσματα από το σενάριο χρήσης ως προς την ολοκλήρωση των εργασιών αλλά και του χρόνου ανταπόκρισης των χρηστών υπολογίσαμε την χρονική αποδοτικότητα (time Based Efficiency) και την συνολική αποδοτικότητα (overall relative efficiency) για την εκτέλεση στο bot και την εκτέλεση στο Softone.

Εκτέλεση στο Softone	Εκτέλεση στο bot
Χρονική αποδοτικότητα : 0.01686 εργασίες/δευτερόλεπτο	Χρονική αποδοτικότητα : 0.0278 εργασίες/δευτερόλεπτο
Συνολική σχετική απόδοση : 53%	Συνολική σχετική απόδοση : 74.5%

Τα παραπάνω νούμερα δείχνουν ότι οι χρήστες εκτέλεσαν αρκετά πιο γρήγορα τις εργασίες μέσω του bot σε σύγκριση με το Softone. Σε όλες τις ερωτήσεις ο μέσος χρόνος εκτέλεσης στο bot ήταν μικρότερος από εκείνον στο Softone. Μία, όμως, ερώτηση καθόρισε σε μεγαλύτερο βαθμό το αποτέλεσμα, η ερώτηση για καταχώρηση ειδοποίησης. Η ερώτηση αυτή δεν μπορεί να επιτευχθεί μόνο με χρήση του Softone και οι χρήστες σπατάλησαν αρκετό χρόνο αναζητώντας αυτή την εργασία μέσα στο πρόγραμμα.

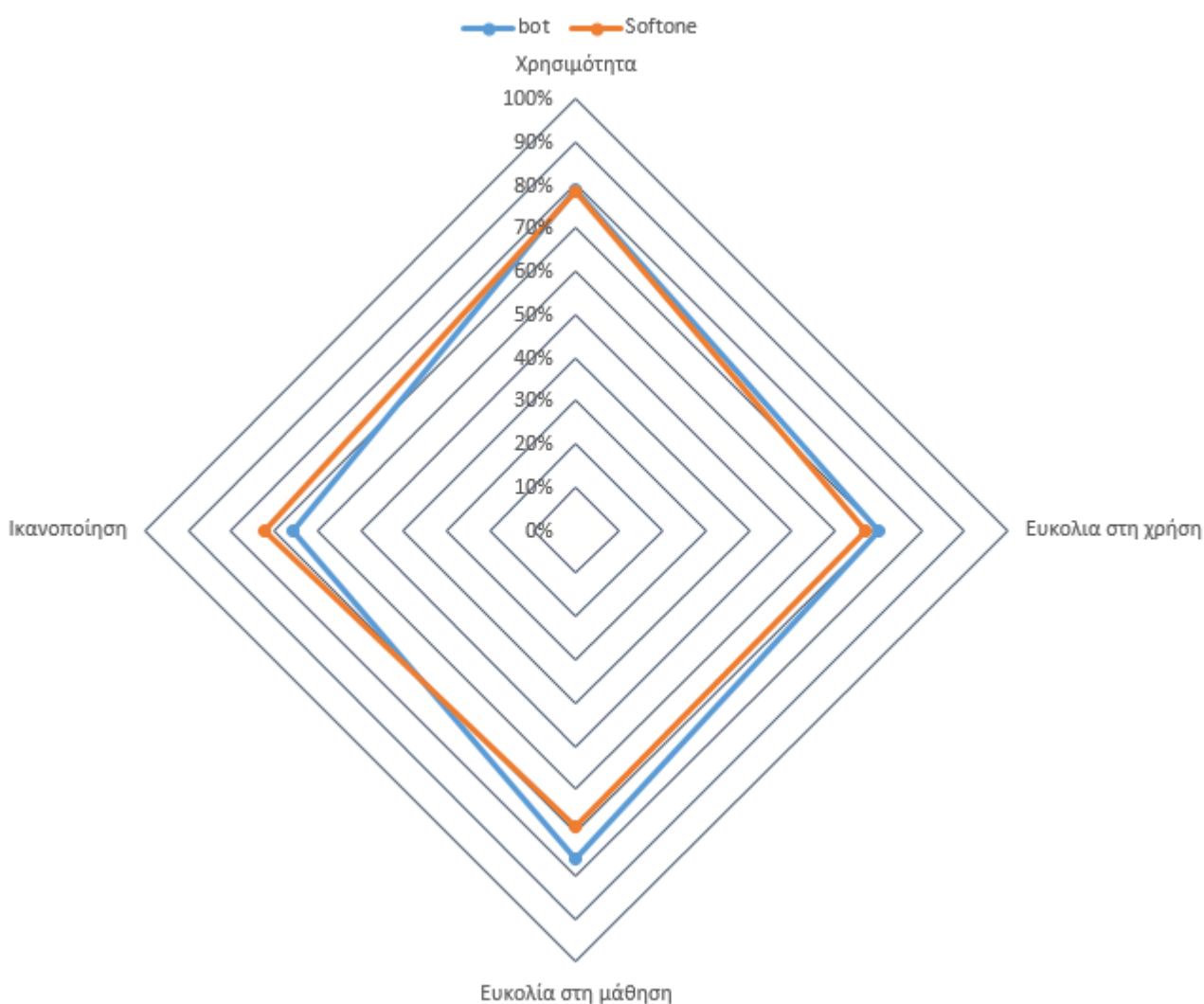
Ικανοποίηση

Όπως έχει ήδη περιγραφεί το ερωτηματολόγιο Use εξετάζει την ικανοποίηση των χρηστών, σύμφωνα με τη χρησιμότητα του προς αξιολόγηση συστήματος, την ευκολία στη χρήση του, την ευκολία στη μάθηση του και την ικανοποίηση των ατόμων όταν το χρησιμοποιούν. Για κάθε μία από αυτές τις 4 ενότητες υπολογίστηκε μέσω του τύπου που παρουσιάστηκε παραπάνω ένα ποσοστό και σχεδιάστηκε διάγραμμα ραντάρ για την απεικόνιση των αποτελεσμάτων. Τα αποτελέσματα που λάβαμε από τους 10 χρήστες που συμπλήρωσαν το ερωτηματολόγιο αυτό για χρήση του Softone χωρίς το bot και για χρήση του επιχειρησιακού βοηθού (bot) για λήψη πληροφοριών από το Softone φαίνονται παρακάτω.

Χρήση του Softone χωρίς το bot	Χρήση του Softone με το bot
Χρησιμότητα : 78.57%	Χρησιμότητα : 79%
Ευκολία στη χρήση : 66.88%	Ευκολία στη χρήση : 69.9%
Ευκολία στη μάθηση : 68.45%	Ευκολία στη μάθηση : 76.19%
Ικανοποίηση : 72.1%	Ικανοποίηση : 65.64%

Από τα παραπάνω, παρατηρούμε ότι οι χρήστες αξιολογούν αρκετά χρήσιμο, με ποσοστό σχεδόν 80% το Softone, είτε με τη συμβολή του επιχειρησιακού βοηθού, είτε χωρίς. Επιπλέον, δεν παρατηρείται μεγάλη διαφοροποίηση (περίπου 3%) στην ευκολία στη χρήση του Softone σε αυτές τις δύο περιπτώσεις. Παρόλα αυτά το bot φαίνεται να έχει υψηλότερο ποσοστό στην ευκολία χρήσης, γεγονός που ήταν αναμενόμενο λόγω της εξυπηρέτησης πιο απλών λειτουργιών σε σχέση με το ολοκληρωμένο σύστημα μηχανογράφησης. Μεγαλύτερη διαφορά μεταξύ των δύο περιπτώσεων παρατηρείται σε ότι αφορά την ευκολία στη μάθηση των δύο συστημάτων, η οποία αγγίζει το 9%. Ήταν αναμενόμενο, αλλά και ζητούμενο κατά την κατασκευή του bot, να αποτελεί ένα πιο απλό και προσιτό σύστημα για τους χειριστές του, συγκριτικά με το Softone ώστε να απλοποιεί πολύπλοκες και χρονοβόρες εργασίες. Τέλος, μια διαφορά 7% φαίνεται να υπάρχει μεταξύ των δύο περιπτώσεων και σε ότι αφορά την ικανοποίηση. Σε αυτή την ενότητα μεγαλύτερο ποσοστό έχει το ERP πρόγραμμα καθώς εξυπηρετεί περισσότερες ανάγκες και έχει πολύ πιο ολοκληρωμένη λειτουργικότητα συγκριτικά με το bot.

Παρακάτω φαίνεται το διάγραμμα με τα αποτελέσματα του ερωτηματολογίου.



Σχήμα 5.9: Διάγραμμα ικανοποίησης χρηστών για τις δύο περιπτώσεις

Έπειτα από τη συμπλήρωση του ερωτηματολογίου, ζητήθηκε από τους χρήστες να καταγράψουν τα πιο θετικά και τα πιο αρνητικά χαρακτηριστικά του επιχειρησιακού βοηθού και να κάνουν σχόλια για δυνατότητες που θα επιθυμούσαν να προστεθούν σε αυτόν.

Η πλειοψηφία των χρηστών κατέγραψε ως βασικό πλεονέκτημα του bot το γεγονός ότι προσφέρει τη δυνατότητα για λήψη πληροφοριών εύκολα και γρήγορα.

Αντίθετα ως μειονέκτημα κατέγραψαν την αδυναμία του να κατανοήσει μεγάλο εύρος λέξεων, κυρίως σε θέματα γενικού περιεχομένου και απλής συνομιλίας και όχι τόσο σε θέματα που αφορούν την εργασία τους.

Κεφάλαιο 6

Συμπεράσματα και Προοπτικές

6.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζονται αναλυτικά τα συμπεράσματα που βγάλαμε μετά την ολοκλήρωση της πρώτης έκδοσης του SoftOne bot καθώς και η προστιθέμενη αξία που προσέφερε το bot στο λογισμικό μηχανογράφησης εταιρειών, SoftOne.

Βασικό αντικείμενο του συγκεκριμένου κεφαλαίου αποτελεί και η πρόταση πιθανών θεμάτων επέκτασης της παρούσας εργασίας.

6.2 Συμπεράσματα και επιτεύγματα

Συμπεράσματα

Κάποια συμπεράσματα που έχουμε βγάλει και αφορούν τη δημιουργία bot είναι:

- Ότι είναι απαραίτητο στα bots να υπάρχει ένα σύστημα ανάλυσης γλώσσας/λέξεων, είτε αυτό είναι δημιούργημα του σχεδιαστή είτε είναι ένα open source τέτοιο σύστημα. Η αναγκαιότητα για αυτό το σύστημα υφίσταται λόγω της ανάγκης να κατανοεί το bot τα μηνύματα που του στέλνει ο χρήστης. Δεν μπορεί ο σχεδιαστής να βασίζεται μόνο στο ότι ο χρήστης θα πληκτρολογήσει αυτό που πρέπει ώστε να εκτελέσει την κάθε εργασία. Όταν αναφερόμαστε σε ένα συνομιλητικό user interface πρέπει να λαμβάνουμε υπόψη και το εύρος των πιθανών μηνυμάτων που υπάρχει περίπτωση να στείλει κάποιος και να αποφασίσουμε ποιο ποσοστό αυτού του εύρους επιθυμούμε να καλύψουμε. Το πρόβλημα της κατανόησης της ερώτησης του χρήστη, δηλ. τι ακριβώς ζητάει από το bot, είναι ένα θέμα που μπορεί να εξελίσσεται συνεχώς. Η σημασία του είναι καθοριστική στα CUI (Conversational User Interfaces), και η ακριβής και γρήγορη κατανόηση, βοηθάει στην υιοθέτηση του (adoption) από τον χρήστη. Σε αυτό το σημείο μπορεί να γίνει αρκετή δουλειά ακόμη και να δοκιμαστούν ιδέες, όπως βοήθεια στη συμπλήρωση προτάσεων (τύπου google), machine learning αλγόριθμοι κλπ.
- Ότι πρέπει να γίνεται πολύ προσεκτική επιλογή του καναλιού μέσω του οποίου θα συνομιλεί το bot με τον χρήστη. Αυτό είναι πολύ σημαντικό καθώς απαιτείται έρευνα από τον σχεδιαστή που αφορά αν το κανάλι ανταποκρίνεται τουλάχιστον στο minimum των δυνατοτήτων που επιθυμεί να προσθέσει στο bot.
- Ότι είναι πολύ σημαντικό το bot να δίνει όντως προστιθέμενη αξία στον χρήστη. Να επιλύει προβλήματα ευκολότερα/γρηγορότερα/καλύτερα συγκριτικά με όλες τις άλλες επιλογές του. Ο χρήστης είναι πολύ εύκολο να βαρεθεί μία επιπλέον επαφή στο Skype η οποία έχει βασικές συνομιλητικές ικανότητες και δεν του προσφέρει τίποτα εκτός από

ένα ιδιαίτερο και πρωτοποριακό user interface. Αντίθετα, δεν θα γίνει το ίδιο με ένα bot το οποίο του επιλύει βασικά προβλήματα της καθημερινότητας του.

- Ότι πρέπει να γίνεται προσπάθεια να ελαχιστοποιείται η εισαγωγή δεδομένων από τον χρήστη, καθώς τα συστήματα CUI μειονεκτούν όταν είναι αναγκαία η εισαγωγή πολλών στοιχείων, σε σχέση με τις οθόνες εισαγωγής σε web ή desktop interfaces, όπου είναι δυνατή η συνολική εντύπωση των στοιχείων πριν την καταχώρησή τους. Σε περίπτωση πολλών στοιχείων σε συστήματα CUI οι πιθανότητες λάθους είναι μεγαλύτερες, ο χειριστής μπορεί να ξεχάσει προηγούμενα πεδία που συμπλήρωσε, να έχει δυσκολία στην επιστροφή για διόρθωση κάποιων από αυτά κλπ. Για να μειωθεί η εισαγωγή στοιχείων πιθανόν να μπορούν να χρησιμοποιηθούν πιο εξειδικευμένες συνομιλίες, για συγκεκριμένες θέσεις εργασίας (role based) αναλόγως τον χειριστή, με προσυμπληρωμένα πεδία (default values).

Επιτεύγματα

Βασικό επίτευγμα της παρούσας εργασίας αποτελεί η προσθήκη ενός νέου πρωτοποριακού συνομιλητικού user interface σε ένα σύστημα μηχανογράφησης εταιρειών. Στο επίπεδο που βρίσκεται αυτή την στιγμή δεν βελτιώνει σε μεγάλο βαθμό τις δυνατότητες του χρήστη. Υπάρχουν, όμως, κάποιες λειτουργίες που καλύπτει το bot οι οποίες δεν ήταν εφικτές πριν από τη δημιουργία του. Μία από αυτές αποτελεί η δυνατότητα αποστολής ειδοποιήσεων. Πριν τη δημιουργία του bot δεν υπήρχε δυνατότητα ο χρήστης να ενημερώνεται αυτόματα από φορητές συσκευές μέσω ειδοποιήσεων (notifications) για εταιρικούς δείκτες που επιθυμεί να δει (όπως για παράδειγμα τις εισπράξεις ημέρας, πωλήσεις καταστημάτων) ή για γεγονότα που συμβαίνουν στην εταιρεία (όπως για παράδειγμα η ανάγκη έγκρισης του για την πραγματοποίηση κάποιας διαδικασίας). Ακόμη, παρατηρούμε ότι η χρήση του bot βελτιώνει σημαντικά τον χρόνο εκτέλεσης κάποιων βασικών εργασιών, οι οποίες προς το παρόν έχουν περιορισμένο αριθμό αλλά αναμένονται να αυξηθούν στο επόμενο διάστημα.

6.3 Προτεινόμενα θέματα επέκτασης έργου

Ενσωμάτωση συστήματος κατανόησης λέξεων

Μία πιθανή επέκταση της παρούσας εργασίας θα ήταν η ενσωμάτωση ενός έτοιμου συστήματος κατανόησης λέξεων όπως είναι το LUIS (Language Understanding Intelligent Service) της Microsoft. Το συγκεκριμένο σύστημα επιτρέπει στην εφαρμογή να κατανοήσει τι θέλει ένα άτομο ανάλογα με τα λόγια του. Το LUIS χρησιμοποιεί machine learning για να επιτρέψει στους προγραμματιστές να δημιουργήσουν εφαρμογές που μπορούν να λαμβάνουν εισερχόμενα μηνύματα από τον χρήστη στη φυσική γλώσσα και να εξάγουν την έννοια από αυτά. Μια εφαρμογή(bot) που συνομιλεί με τον χρήστη μπορεί να μεταβιβάσει την είσοδο χρήστη σε μια εφαρμογή LUIS και να λάβει λεπτομερείς πληροφορίες που αφορούν το νόημα του μηνύματος του εκάστοτε χρήστη.

Μια εφαρμογή LUIS είναι ένα γλωσσικό μοντέλο που σχεδιάζεται από τον κάθε σχεδιαστή και προσαρμόζεται στις ανάγκες του. Ο καθένας μπορεί να ξεκινήσει με ένα προεπιλεγμένο μοντέλο, να δημιουργήσει το δικό του ή να συνδυάσει κομμάτια ενός προεγκατεστημένου μοντέλου με τις δικές του επιπλέον προσαρμοσμένες πληροφορίες.

Ένα μοντέλο ξεκινάει με μια λίστα γενικών προθέσεων χρηστών όπως “Book Flight” ή “Contact Help Desk”. Μόλις προσδιοριστούν οι προθέσεις, ο σχεδιαστής παρέχει παραδείγματα φράσεων που ονομάζονται “utterances” (δηλώσεις) για τις προθέσεις.

Αφού το μοντέλο σχεδιασθεί, εκπαιδευθεί και δημοσιευθεί, είναι έτοιμο να λαμβάνει και να επεξεργάζεται τις δηλώσεις. Η εφαρμογή LUIS λαμβάνει τη φράση ως αίτημα HTTP και απαντά με εξαγόμενες προθέσεις χρηστών. Η εφαρμογή στέλνει τη δήλωση και λαμβάνει την “εκτίμηση” LUIS ως αντικείμενο JSON (JSON object)[24].

Βασικές έννοιες του LUIS

- Προθέσεις[24].

Μια πρόθεση αντιπροσωπεύει τις ενέργειες που ο χρήστης θέλει να εκτελέσει. Η πρόθεση είναι ένας σκοπός ή ένας στόχος που εκφράζεται σε μια εισήγηση ενός χρήστη, όπως η κράτηση μιας πτήσης, η πληρωμή ενός λογαριασμού ή η εύρεση ενός ειδησεογραφικού άρθρου. Ο κάθε σχεδιαστής ορίζει και αναφέρει τις προθέσεις που αντιστοιχούν σε αυτές τις ενέργειες. Μια εφαρμογή ταξιδιού μπορεί να ορίσει μια πρόθεση με την ονομασία “BookFlight”.

- Δηλώσεις[24].

Μια δήλωση είναι η το τμήμα κειμένου του χρήστη, το οποίο απαιτείται να κατανοηθεί από την εφαρμογή. Μπορεί να είναι μια φράση, όπως “Κράτηση εισιτηρίου για Παρίσι” ή ένα κομμάτι μιας φράσης, όπως “Κράτηση” ή “Πτήση Παρίσι”. Οι δηλώσεις δεν είναι πάντα καλά διαμορφωμένες και μπορεί να υπάρξουν πολλές παραλλαγές της έκφρασης για μια συγκεκριμένη πρόθεση.

- Οντότητες[24].

Μια οντότητα αντιπροσωπεύει λεπτομερείς πληροφορίες που είναι συναφείς σε μία δήλωση. Για παράδειγμα, στη δήλωση “Κράτηση εισιτηρίου για Παρίσι”, η λέξη “Παρίσι” είναι μια τοποθεσία. Αναγνωρίζοντας και επισημαίνοντας τις οντότητες που αναφέρονται στη συζήτηση του χρήστη, το LUIS βοηθά να επιλεγεί η συγκεκριμένη ενέργεια που πρέπει να ακολουθηθεί ώστε να δοθεί απάντηση στο αίτημα ενός χρήστη.

Πρόθεση	Δήλωση	Οντότητα
Intent	Sample User Utterance	Entities
BookFlight	"Book a flight to Seattle ?"	Seattle
StoreHoursAndLocation	"When does your store open ?"	open
ScheduleMeeting	"Schedule a meeting at 1pm with Bob in Distribution"	1pm, Bob

Σχήμα 6.1: Παράδειγματα βασικών εννοιών LUIS[24].

Θα ήταν, λοιπόν πολύ ενδιαφέρον να γίνει μία έρευνα ενσωμάτωσης ενός συστήματος σαν το παραπάνω στο SoftOne bot ή και σε άλλα bots που υποστηρίζουν παρόμοιες λειτουργίες.

Δημιουργία του bot και σε άλλο κανάλι

Μία ακόμη επέκταση του έργου θα ήταν η δημιουργία του bot και σε κάποιο άλλο κανάλι,

όπως για παράδειγμα το Facebook Messenger.

Κατά τη διάρκεια σχεδιασμού του bot πραγματοποιήσαμε και μία έρευνα των δυνατοτήτων του messenger ώστε να αποφασίσουμε αν θα επεκταθεί το bot στο συγκεκριμένο κανάλι, το οποίο είναι γεγονός ότι είναι πολύ πιο δημοφιλές από το Skype.

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, χρησιμοποιήσαμε το Microsoft Bot Framework διότι λειτουργεί σαν διαμεσολαβητής μεταξύ bot και καναλιού και αναλαμβάνει να δώσει συγκεκριμένη μορφοποίηση στα μηνύματα – από και προς τον χρήστη- ανεξάρτητα από το κανάλι στο οποίο λειτουργεί το bot.

Παρόλα αυτά στην προσπάθεια να ενσωματώσουμε το bot και στο messenger ήρθαμε αντιμέτωποι με κάποια προβλήματα που αφορούν περιορισμούς που έχει το συγκεκριμένο κανάλι καθώς και με κάποιες θετικές εκπλήξεις που αφορούν επιπλέον δυνατότητες του καναλιού αυτού.

Προβλήματα συγκριτικά με το Skype

- Δεν έχει τις ίδιες δυνατότητες για μορφοποίηση του προς αποστολή μηνύματος όπως το Skype. Για παράδειγμα δεν υποστηρίζει την αποστολή bold χαρακτήρων.
- Δεν επιτρέπει την ύπαρξη πάνω από τριών κουμπιών (buttons) σε κάρτες. Αυτό είναι μειονέκτημα για το συγκεκριμένο bot καθώς δεν είμαστε σε θέση να γνωρίζουμε εξ' αρχής τον ακριβή αριθμό των προτεινόμενων επιλογών (σε μορφή buttons) που θα δώσει το bot στο χρήστη και δεν επιθυμούμε να μειώσουμε τον συγκεκριμένο αριθμό, κάτι το οποίο θα αλλοιώσει την λειτουργία του.

Πλεονεκτήματα συγκριτικά με το Skype

- Είναι πιο δημοφιλές από το Skype, που υπονοεί πιο εύκολη πρόσβαση για τους χρήστες του SoftOne.
- Δίνει τη δυνατότητα ορισμού ενός μενού που ονομάζεται “persistent menu”, στο οποίο ο καθένας μπορεί να έχει πρόσβαση με το πάτημα ενός μόνο κουμπιού. Αυτό για το SoftOne bot σημαίνει ότι ο χρήστης δεν θα χρειάζεται πλέον να πληκτρολογεί “μενού” για να έχει πρόσβαση στις επιλογές του, αντίθετα θα το κάνει με το πάτημα ενός button. Ακόμη, θα μας έδινε τη δυνατότητα να επιτρέπουμε την επιλογή μίας από τις επιλογές του μενού σε οποιαδήποτε φάση της συζήτησης με το bot, κάτι το οποίο δεν μπορεί να επιτευχθεί με τις υπάρχουσες συνθήκες.

Για τους παραπάνω λόγους μία έρευνα σχετικά με αυτή την επέκταση του έργου θα ήταν αρκετά ενδιαφέρουσα.

Πιο εξειδικευμένη προσέγγιση της λειτουργικότητας

Στη διάρκεια του έργου παρακολουθήθηκε η χρήση που κάνουν εργαζόμενοι της εταιρείας στο ERP πρόγραμμα SoftOne. Διαπιστώθηκε η διαφορετικότητα των εργασιών που εκτελούν καθημερινά χρήστες με τελείως διαφορετικές ειδικότητες και διαβαθμίσεις στην εταιρεία. Ένα ERP σύστημα προσφέρει ένα μεγάλο αριθμό δυνατοτήτων στους εργαζομένους, κάποιες από τις οποίες είναι πολύ απλές ενώ άλλες πολύ πολύπλοκες. Η πρώτη προσέγγιση του bot, την οποία και πραγματοποιήσαμε σε ένα σχετικά μικρό χρονικό διάστημα αφορούσε γενικές εργασίες και ερωτήσεις οι οποίες είναι χρήσιμες, κυρίως, σε χειριστές με αρκετές αρμοδιότητες που έχουν δικαίωμα πρόσβασης σε πολλές πληροφορίες.

Μία ενδιαφέρουσα επέκταση του έργου θα αποτελούσε μία πιο λεπτομερής μελέτη των εργασιών που εκτελούν εργαζόμενοι σε διάφορες θέσεις. Η μελέτη αυτή θα μπορούσε να μας δώσει

απάντηση στο ερώτημα αν έχει νόημα η εκτέλεση αυτών των εργασιών μέσω ενός επιχειρησιακού βοηθού και φυσικά αν είναι εφικτό να εκτελεσθούν αυτές μέσω ενός bot. Με βάση αυτά τα στοιχεία θα είμαστε σε θέση να καλύψουμε και πιο εξειδικευμένες ανάγκες χρηστών, απαντώντας και σε άλλου είδους ερωτήματα τα οποία είναι πιθανό να έχουν πολύ περισσότερη πολυπλοκότητα. Κάτι τέτοιο θα απαιτούσε σίγουρα πολύ περισσότερο χρόνο και εργασία.

Κεφάλαιο 7

Βιβλιογραφία

- [1] Φιτσιλής, Π.(2015). Σύγχρονα πληροφοριακά συστήματα επιχειρήσεων. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/2256>
- [2] Microservices. (n.d). In Wikipedia. Ανακτήθηκε από <https://en.wikipedia.org/wiki/Microservices>
- [3] JAMES TEMPERTON. (2016, November 8). Apps are dying. Long live the subservient bots ready to fulfil your every desire. Ανακτήθηκε από <http://www.wired.co.uk/article/here-come-the-bots>
- [4] Microsoft. (2017, December 17). Add intelligence to bots with Cognitive Services. Ανακτήθηκε από <https://docs.microsoft.com/en-us/bot-framework/bot-service-concept-intelligence>
- [5] 10 Best Skype Bots You Should Use. (2016, December 21). Ανακτήθηκε από <https://beebom.com/best-skype-bots/>
- [6] Tom Warren. (2014, April 2). The story of Cortana, Microsoft's Siri killer. Ανακτήθηκε από <https://www.theverge.com/2014/4/2/5570866/cortana-windows-phone-8-1-digital-assistant>
- [7] Unit4. (2017, May 2). Unit4 releases Wanda bot; the first truly self-driving ERP user experience. Ανακτήθηκε από <http://www.unit4.com/us/about/news/2017/05/unit4-releases-wanda-bot-the-first-truly-self-driving-erp-user-experience>
- [8] Microsoft. (2017, December 13). About Bot Service. Ανακτήθηκε από <https://docs.microsoft.com/en-us/bot-framework/bot-service-overview-introduction>
- [9] Microsoft. (2017, December 13). Key concepts. Ανακτήθηκε από <https://docs.microsoft.com/en-us/bot-framework/dotnet/bot-builder-dotnet-concepts>
- [10] Microsoft. (2017, December 13). Design and control conversation flow. Ανακτήθηκε από <https://docs.microsoft.com/en-us/bot-framework/bot-service-design-conversation-flow>

- [11] Skype. (n.d). Implementation. Ανακτήθηκε από
<https://dev.skype.com/bots>
- [12] Transmission Control Protocol. (n.d). In Wikipedia. Ανακτήθηκε από
https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [13] Hypertext Transfer Protocol. (n.d). In Wikipedia. Ανακτήθηκε από
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [14] Softone. (n.d). Soft1 Web Services. Ανακτήθηκε από
<https://www.softone.gr/ws/services.html>
- [15] Component Object Model. (n.d). In Wikipedia. Ανακτήθηκε από
https://en.wikipedia.org/wiki/Component_Object_Model
- [16] Microsoft. (2017, December 13). Principles of bot design. Ανακτήθηκε από
<https://docs.microsoft.com/en-us/bot-framework/bot-service-design-principles>
- [17] Microsoft. (2017, December 13). Design a bot's first user interaction. Ανακτήθηκε από
<https://docs.microsoft.com/en-us/bot-framework/bot-service-design-first-interaction>
- [18] Microsoft. (2017, December 13). Design bot navigation. Ανακτήθηκε από
<https://docs.microsoft.com/en-us/bot-framework/bot-service-design-navigation>
- [19] Justin Mifsud . (2015, June 22). Usability Metrics – A Guide To Quantify The Usability Of Any System. Ανακτήθηκε από
<https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>
- [20] Αβούρης Νικόλαος, Κατσanos Χρήστος, Τσέλιος Νικόλαος, Μουστάκας Κωνσταντίνος. (Νοέμβριος 2016). ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΑΝΘΡΩΠΟΥ - ΥΠΟΛΟΓΙΣΤΗ. Εκδότης: Πανεπιστήμιο Πατρών.
- [21] Likert scale. (n.d). In wikipedia. Ανακτήθηκε από
https://en.wikipedia.org/wiki/Likert_scale
- [22] Arnold Lund . (2016, March 13). Measuring Usability with the USE Questionnaire. Ανακτήθηκε από
https://www.researchgate.net/publication/230786746_Measuring_Usability_with_the_USE_Questionnaire
- [23] Κουτσαμπάσης, Π., 2015. Αξιολόγηση διαδραστικών συστημάτων με επίκεντρο τον χρήστη. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο:
<http://hdl.handle.net/11419/2765>
- [24] Microsoft. (2017, December 13). About Language Understanding (LUIS). Ανακτήθηκε από
<https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home>

Παράρτημα Α΄

Κώδικας εργασίας

Στο παράρτημα αυτό δεν φαίνεται ολόκληρος ο κώδικας. Έχουν καταγράψει ενδεικτικά κάποια τμήματα του ώστε να γίνει πλήρως κατανοητή η δομή του.

Αρχικά παρατίθεται η main συνάρτηση του bot.

Σε αυτή, αρχικά, γίνεται φόρτωση όλων των άλλων αρχείων που περιέχουν microservices. Η βασική ιδέα της συνάρτησης είναι :

- Ελέγχεται αν το μήνυμα του χρήστη ανήκει σε κάποια από τις περιπτώσεις που απαιτούν άλλη από την βασική διαχείριση. Αυτές είναι η περίπτωση που ο χρήστης θέλει να λάβει υπερσύνδεσμο για να εκτελέσει μία εργασία από το μενού του, η περίπτωση που στέλνεται μία ειδοποίηση η οποία απαιτεί έγκριση, η περίπτωση που ο χρήστης χρησιμοποιεί ανάρμοστο λεξιλόγιο για να απευθυνθεί στο bot κλπ.
- Καλεί τη συνάρτηση που επεξεργάζεται το μήνυμα του χρήστη, όπως έχει περιγραφεί.
- Ελέγχει την ομοιότητα του με τις διαθέσιμες εργασίες και ανάλογα με το αποτέλεσμα προχωρά.

Έπειτα, βλέπουμε τη συνάρτηση menumain στην οποία μεταφέρεται ο έλεγχος όταν ο χρήστης ζητήσει link για να εκτελέσει κάποια εργασία από το προσωπικό του μενού.

Πρέπει να σημειωθεί ότι τα όλα τα λεκτικά βρίσκονται συγκεντρωμένα σε ξεχωριστό αρχείο, όπου υπάρχουν στα ελληνικά και στα αγγλικά. Στα παρακάτω τμήματα τα λεκτικά σημειώνονται με το όνομα της μεταβλητής που τα περιέχει μαζί με την παράμετρο lang η οποία ορίζει την γλώσσα της συγκεκριμένης εγκατάστασης.

```
1 include('/js/variables.js'); //include needed js scripts
2 include('/js/menu.js');
3 include('/js/receivables.js');
4 include('/js/upoloipo.js');
5 include('/js/sumamnt.js');
6 include('/js/cmpOverview.js');
7 include('/js/commands.js');
8 include('/js/gadgets.js');
9 include('/js/checkedCommands.js');
10 include('/js/typical.js');
11 include('/js/menuJobs.js');
12 include('/js/basicFunc.js');
13
14 for (var j=0;j<typical.length;j++){ //word parse typical
15     var typLen = typical[j].title.length;
16     for (var k=0; k<typLen; k++){
17         var typTitle = typical[j].title[k],
18             parseTitle = BotParse(lang, typTitle);
19         typical[j].title.push(parseTitle);
20     }
21 }
22
23 var fromJob=false;
24 function main(input) {
25
26     if (input.validate)
27         return DoValidation(input);
28     if (botstate.job== "task"){ //check if it is task
29         return menumain(input)
30     }
31
32     if (input.id=="approval01"){ //check if it needs approval
33         var appr = input.formdata.approval.split(";"),
34             obj = appr[0].split(":"),
35             Tfield = appr[1].split(":"),
36             vApprv = createApprv(obj, Tfield);
```

```

37     if (!vApprv.success){
38         return {text : apprvText[lang]}
39     }
40     else{
41         return {text : "ok!"}
42     }
43 }
44
45 if (input.id == "selectone") {
46     input.id      = input.formdata.id;
47     input.formdata = null;
48 }
49 if (input.id){
50     return WorkById(input);
51 }
52
53
54
55 if(input.text && !fromJob){                                     //Parse input
56     for (var i=0; i<checkedCommands.length; i++){
57         if (checkedCommands[i].id == input.text){
58             input.id = checkedCommands[i].id;
59             return ExecuteCommand(checkedCommands[i], input);
60         }
61     }
62     input.text = BotParse(lang,input.text);
63     input.text=input.text.trim();
64     if (input.text==""){
65         return {text : emptyText[lang]}
66     }
67     badWords(input.text);
68 }
69
70 fromJob = false;
71
72
73
74 if (botstate.bad){                                             //Check if it is a badword
75     botstate.bad = false;
76     if(botstate.badcount==1){
77         botstate.badcount=0;
78         return {text : text1[lang]}
79     }
80     else {
81         return {text : text2[lang]}
82     }
83 }
84
85
86
87 var result  = {},
88     scores  = {},
89     atext    = input.text;
90
91 if (atext==botstate.input){
92     botstate.count+=1;
93 }
94 else botstate.count=0;
95
96 if (botstate.count>=2){

```

```

97     return {text: text3[lang]}
98 }
99
100 botstate.input=atext;
101
102
103 for (var i=0; i<typical.length; i++){
104     if (Checkinit(typical[i].title, atext)==0)
105         return {text : typical[i].ans};
106 }
107
108 for (var i=0; i<checkedCommands.length; i++)
109 {
110     var score = Checkinit(checkedCommands[i].title, atext);
111     if (score==0)
112     {
113         input.id = checkedCommands[i].id;
114         return ExecuteCommand(checkedCommands[i], input);
115     }
116     else if (score==1){
117         FillScores(scores,score,i)
118     }
119     else if (score==2){
120         FillScores(scores,score,i)
121     }
122     else if (score==3){
123         FillScores(scores,score,i)
124     }
125     else if (score==4){
126         FillScores(scores,score,i)
127     }
128     else if (score==5){
129         FillScores(scores,score,i)
130     }
131     else if (score==6){
132         FillScores(scores,score,i)
133     }
134     else if (score==7){
135         FillScores(scores,score,i)
136     }
137     else if (score==8){
138         FillScores(scores,score,i)
139     }
140     else if (score==9){
141         FillScores(scores,score,i)
142     }
143 }
144
145 }
146 var count    = 0,
147     choices   = "",
148     choicesArray= [];
149 for (var i in scores)
150 {
151     for (var j=0; j<scores[i].length && count<4; j++)
152     {
153         var command = checkedCommands[scores[i][j]];
154         choicesArray.push(command.title[0]+'='+command.id);
155         count++;
156     }

```

```

157 }
158
159 if (count>0){
160     if (choicesArray.length==1){
161         return WorkById(command)
162     }
163     else{
164         choicesArray.push(cancel[lang])
165         choices = choicesArray.join("|");
166         return FillForm(choices);
167     }
168
169 }
170
171 return { text: error[lang]};
172 }
173
174 function menumain(input){
175     if (input.validate)
176         return DoValidation(input);
177
178     var scores ={},
179         choices = [];
180
181     if(input.text){
182         input.text = BotParse(lang,input.text);
183         input.text=input.text.trim();
184         var toMain1 = toMain(0, input),    //check if asking for help
185             toMain2 = toMain(5, input);    //check if asking for menu
186         if(toMain1=="question"||toMain2=="question"){
187             return main(input)
188         }
189
190         badWords(input.text);
191     }
192     if (botstate.bad){
193         botstate.bad = false;
194         if(botstate.count==1){
195             botstate.count=0;
196             return {text : text1[lang]}
197         }
198         else {
199             return {text : text2[lang]}
200         }
201     }
202     for (var i=0; i<menujobs.length; i++){
203         var score = Checkinit(menujobs[i].caption, input.text);
204         if ((menujobs[i].caption == input.text) || (score==0))
205         {
206             var menu = menujobs[i],
207                 obj = CreateObj(menu)
208             choices.push(obj);
209             return MenuForm(choices);
210         }
211         else if (score==1){
212             FillScores(scores,score,i)
213         }
214         else if (score==2){
215             FillScores(scores,score,i)
216         }

```

```

217     else if (score==3){
218         FillScores(scores,score,i)
219     }
220     else if (score==4){
221         FillScores(scores,score,i)
222     }
223     else if (score==5){
224         FillScores(scores,score,i)
225     }
226     else if (score==6){
227         FillScores(scores,score,i)
228     }
229     else if (score==7){
230         FillScores(scores,score,i)
231     }
232     else if (score==8){
233         FillScores(scores,score,i)
234     }
235     else if (score==9){
236         FillScores(scores,score,i)
237     }
238 }
239 var count=0;
240
241     for (var i in scores)
242     {
243         for (var j=0; j<scores[i].length && count<10; j++)
244         {
245             var menu = menujobs[scores[i][j]],
246                 obj = CreateObj(menu);
247             choices.push(obj);
248             count++;
249         }
250     }
251     if (count>0){
252         return MenuForm(choices);
253     }
254     return { text: error[lang]};
255 }

```

Listing 1: Main bot function

Παρακάτω βλέπουμε το microservice που αφορά το μενού του bot. Εμφανίζει τις βασικές επιλογές που είναι Ερωτήσεις, Εργασίες και Ειδοποιήσεις και προχωρά ανάλογα με την επιλογή που θα κάνει ο χρήστης.

```
1 var menu = [ {
2   title   : menuTitle[lang],
3   id      : "menu",
4   form    : [{
5     field  : "job",
6     prompt : jobPrompt[lang],
7     type   : "optiongoto",
8     options : jobOptions[lang]
9   }, {
10    field   : "questions",
11    prompt  : questionsPrompt[lang],
12    type    : "text",
13    nextfield : "last"
14  }, {
15    field   : "ergasies",
16    prompt  : ergasiesPrompt[lang],
17    type    : "text",
18    nextfield : "last"
19  }, {
20    field   : "notificationHour",
21    prompt  : notHourPrompt[lang],
22    type    : "option",
23    options : notHourOptions[lang],
24    nextfield : "notifications"
25  }, {
26    field   : "notifications",
27    prompt  : notPrompt[lang],
28    type    : "text",
29    nextfield : "last"
30  } ],
31  command : function (input) {
32    var result = {};
33    if (input.formdata.job==1) {
34      botstate.job = "question";
35      return main({text : input.formdata.questions})
36    }
37    else if (input.formdata.job==2) {
38      botstate.job = "task";
39      return menumain({text : input.formdata.ergasies})
40    }
41    else if (input.formdata.job==3)
42    {
43      var hour = input.formdata.notificationHour;
44      setTime(hour);
45      botstate.job = "notification";
46      return main({text : input.formdata.notifications})
47    }
48  }
49 }, {
50  title   : quesTitle[lang],
51  id      : "ques",
52  form    : [{
53    field  : "questions",
54    prompt : questionsPrompt[lang],
55    type   : "text"
56  } ],
57 }
```

```

58   command : function(input){
59       var result = {};
60       botstate.job = "question";
61       return main({text : input.formdata.questions})
62   }
63 },{
64   title    : tasksTitle[lang],
65   id       : "tasks",
66   form     : [{
67       field   : "ergasies",
68       prompt  : ergasiesPrompt[lang],
69       type    : "text"
70   }],
71   command : function(input){
72       var result = {};
73       botstate.job = "task";
74       return menumain({text : input.formdata.ergasies})
75   }
76 },{
77   title    : alertTitle[lang],
78   id       : "alert",
79   form     : [{
80       field   : "notificationHour",
81       prompt  : notHourPrompt[lang],
82       type    : "option",
83       options : notHourOptions[lang]
84   }],{
85       field   : "notifications",
86       prompt  : notPrompt[lang],
87       type    : "text"
88   }],
89   command : function (input){
90       var hour = input.formdata.notificationHour;
91       setTime(hour);
92       botstate.job = "notification";
93       return main({text : input.formdata.notifications})
94   }
95 }
96 ]

```

Listing 2: Bot menu microservice

Παρακάτω, παρατίθεται ενδεικτικά το microservice που αφορά τα υπόλοιπα πελατών, προμηθευτών και ειδών.

Για κάθε μία από αυτές τις επιλογές, εμφανίζεται μία φόρμα στον χρήστη ώστε να συμπληρώσει τα στοιχεία που απαιτούνται. Έπειτα, ανακτώνται αυτά τα στοιχεία, ορίζεται το query σε sql το οποίο απαιτείται για την ανάκτηση των πληροφοριών, αυτό εκτελείται και εμφανίζεται το κατάλληλο μήνυμα. Με παρόμοιο τρόπο λειτουργούν όλα τα microservices.

```
1
2 var upoloipo = [
3   {
4     title : upolCusTitle[lang],
5     id    : "upolCus",
6     access : ["CUSFINDATA"],
7     form  : [
8       {
9         prompt : customerPrompt[lang],
10        field  : "customer",
11        type   : "optiongoto",
12        options : customerOptions[lang]
13      }, {
14        field  : "cusname",
15        prompt : cusnamePrompt[lang],
16        type   : "text",
17        onvalidate : "SelectCustomer",
18        nextfield : "last"
19      }
20    ],
21    command : function(input){
22      var result = {};
23      if (input.formdata.customer==2){
24        var trdr = input.formdata.cusname;
25        var query = "SELECT ISNULL(B.LBAL,0) AS YPOL FROM TRDR A LEFT
26                   OUTER JOIN TRDFINDATA B ON B.COMPANY=:X.SYS.COMPANY "+
27                   " AND B.TRDR=A.TRDR WHERE  A.COMPANY=:X.SYS.COMPANY AND A.
28                   SODTYPE=13 AND A.TRDR="+ trdr;
29        var data = sql2json(query);
30        if (data!=null){
31          if (data[0].YPOL!=0){
32            result.text = upolCusResult1[lang]+ X.FORMATFLOAT(data[0].YPOL
33                          , 'V')}
34          else result.text = upolCusResult2[lang]
35        }
36      }
37      else if (input.formdata.customer==1){
38        var query = "SELECT sum(ISNULL(B.LBAL,0)) AS YPOL FROM TRDR A LEFT
39                   OUTER JOIN TRDFINDATA B ON B.COMPANY=:X.SYS.COMPANY AND B.TRDR
40                   =A.TRDR WHERE A.COMPANY=:X.SYS.COMPANY AND A.SODTYPE=13 "
41        var data = sql2json(query);
42        if (data!=null){
43          if (data[0].YPOL!=0){
44            result.text = upolCusResult3[lang] + X.FORMATFLOAT(data[0].
45                          YPOL, 'V')}
46          else result.text = upolCusResult4[lang]
47        }
48      }
49      else{
50        result.text = "No data";
51      }
52    }
53  }
54 ]
```

```

48     }
49     return checkResult(result, input.id, input);
50 }
51 },
52 {
53     title : upolSupTitle[lang],
54     id     : "upolSup",
55     access : ["SUPFINDATA"],
56     form  : [
57         {
58             prompt    : supplierPrompt[lang],
59             field     : "supplier",
60             type      : "optiongoto",
61             options    : supplierOptions[lang]
62         }, {
63             field     : "supname",
64             prompt    : supnamePrompt[lang],
65             type      : "text",
66             onvalidate : "SelectSup",
67             nextfield : "last"
68         }
69     ],
70     command : function(input){
71         var result = {};
72         if (input.formdata.supplier==2){
73             var trdr = input.formdata.supname;
74             var query = "SELECT ISNULL(B.LBAL,0) AS YPOL FROM TRDR A LEFT
75                 OUTER JOIN TRDFINDATA B ON B.COMPANY=:X.SYS.COMPANY"+
76                 " AND B.TRDR=A.TRDR WHERE A.COMPANY=:X.SYS.COMPANY AND A.
77                 SODTYPE=12 AND A.TRDR="+ trdr;
78             var data = sql2json(query);
79             if (data!=null){
80                 if (data[0].YPOL!=0){
81                     result.text = upolSupResult1[lang] + X.FORMATFLOAT(data[0].YPOL,
82                         'V');
83                 }
84                 else result.text = upolSupResult2[lang]
85             }
86             else{
87                 result.text = "No data";
88             }
89         }
90         else if (input.formdata.supplier==1){
91             var query = "SELECT sum(ISNULL(B.LBAL,0)) AS YPOL FROM TRDR A LEFT
92                 OUTER JOIN TRDFINDATA B ON B.COMPANY=:X.SYS.COMPANY"+
93                 " AND B.TRDR=A.TRDR WHERE A.COMPANY=:X.SYS.COMPANY AND A.
94                 SODTYPE=12 "
95             var data = sql2json(query);
96             if (data!=null){
97                 if (data[0].YPOL!=0){
98                     result.text = upolSupResult3[lang] + X.FORMATFLOAT(data[0].
99                         YPOL, 'V')
100                 }
101                 else result.text = upolSupResult2[lang]
102             }
103             else{
104                 result.text = "No data";
105             }
106         }
107     }
108     return checkResult(result, input.id, input);
109 }

```

```

102 },
103 {
104     title : upolMtrlTitle[lang],
105     id    : "upolMtrl",
106     access : ["ITEWHOUSE"],
107     form  : [
108         {
109             field    : "mtrl",
110             prompt   : mtrlPrompt[lang],
111             type     : "text",
112             onvalidate : "SelectMtrl",
113             nextfield : "last"
114         }
115     ],
116     command : function(input){
117         var result = {};
118         var d      = new Date();
119         var year   = d.getFullYear();
120         var mtrl   = input.formdata.mtrl;
121         var query  = "SELECT ISNULL(B.QTY1,0) AS QTY FROM MTRL A LEFT OUTER
122                     JOIN MTRDATA B ON B.COMPANY=:X.SYS.COMPANY"+
123                     " AND A.MTRL=B.MTRL AND B.FISCPRD="+year+" WHERE A.COMPANY=:
124                     X.SYS.COMPANY AND A.SODTYPE=51 AND A.MTRL="+mtrl;
125         var data   = sql2json(query);
126         if (data!=null){
127             result.text = mtrlResult[lang] + X.FORMATFLOAT(data[0].QTY,'Q')
128         }
129         else{
130             result.text = "No data";
131         }
132         return checkResult(result, input.id, input);
133     }
134 }
135
136 function SelectCustomer(input) {
137     var sql = "select top 5 trdr, name from trdr where sodtype=13 and name
138             like '%" + input.text + "%'";
139     var prompt0 = SelectCustomer0[lang];
140     var prompt1 = SelectCustomer1[lang];
141     var prompt2 = SelectCustomer2[lang];
142     var result = GeneralSelect(input, sql, prompt0, prompt1, prompt2);
143     return result;
144 }
145
146 function SelectSup(input) {
147     var sql = "select top 5 trdr, name from trdr where sodtype=12 and
148             name like '%" + input.text + "%'";
149     var prompt0 = SelectSup0[lang];
150     var prompt1 = SelectSup1[lang];
151     var prompt2 = SelectSup2[lang];
152     var result = GeneralSelect(input, sql, prompt0, prompt1, prompt2);
153     return result;
154 }
155
156 function SelectMtrl(input) {
157     var sql = "select top 10 mtrl, name from mtrl where sodtype=51 and

```

```

        name like '%" + input.text + "%' AND company=:X.SYS.COMPANY";
158 var prompt0 = SelectMtrl0[lang];
159 var prompt1 = SelectMtrl1[lang];
160 var prompt2 = SelectMtrl2[lang];
161 var result = GeneralSelect(input, sql, prompt0, prompt1, prompt2);
162 return result;
163 }
164
165 function GeneralSelect(input, sql, prompt0, prompt1, prompt2) {
166     var data = sql2array(sql);
167     var result = {},
168         resultArray = [];
169     var l = data.length;
170     if (l==0)
171     {
172         result.prompt = prompt0;
173         result.type = "option";
174         result.options = "";
175     }
176     else if (l==1){
177         result.prompt = prompt1;
178         result.options = data[0][1] + "=" + data[0][0];
179         result.type = "option";
180     }
181     else
182     {
183         for (var i=0; i<l; i++){
184             resultArray.push(data[i][1].replace("'", '\\\\') + "=" + data[i][0])
185         }
186         result.options = resultArray.join("|");
187         result.type = "option";
188         result.prompt = prompt2
189     }
190     return result;
191 }
192 }

```

Listing 3: Balance (upoloipo) microservice

Παρακάτω βλέπουμε τις βασικές συναρτήσεις που χρησιμοποιούνται σε όλα τα αρχεία

```
1 //Find score
2 function Checkinit(title, text) {
3     if (title.constructor === Array) {
4         tempscore = 11;
5         for (var i=0; i<title.length; i++){
6             matchscore = BotMatch(text,title[i])
7             if (matchscore==0) return 0;
8             else if (matchscore<tempscore){
9                 tempscore=matchscore
10            }
11        }
12        return tempscore;
13    }
14    else return BotMatch(text,title);
15 }
16
17 //Execute microservice command
18 function ExecuteCommand(acommand, input) {
19     botstate.count=0;
20     if (acommand.form)
21     {
22
23         if (input.formdata) {
24             return acommand.command(input);
25         }
26         else {
27             return { runform : acommand.form, text: input.text, id: acommand.id
28                 };
29         }
30     }
31     else
32     return acommand.command(input);
33 }
34
35 //Check before sending an answer
36 function checkResult(result, id, input){
37     if (botstate.job== "notification"){
38         input.id= id;
39         input = JSON.stringify(input);
40         X.SCHEDULEBOTJOB(id,input, botstate.time);
41         botstate.time = 0
42         botstate.job = "question";
43         return {text : "ok ;)"}
44     }else{
45         return result;
46     }
47 }
48
49 function ScheduleJob(input){
50     var result = {};
51     for (var i=0; i<checkedCommands.length; i++){
52         if (checkedCommands[i].id==input.id){
53             return checkedCommands[i].command(input);
54         }
55     }
56     result.text = "Something is wrong";
57     return result;
58 }
```

```

59
60 //Fill form in order to propose options
61 function FillForm(choices){
62     var optionform = {runform : [{
63         field : "id",
64         type : "option",
65         options : choices,
66         prompt : fillFormPrompt[lang]
67     }],
68     id : "selectone"
69 }
70 return optionform;
71 }
72
73 //Fill object that keeps scores
74 function FillScores(scores,score,i){
75     if (typeof(scores[score])=="undefined"){
76         scores[score] = [i];
77     }
78     else scores[score].push(i)
79 }
80
81 //Validate necessary values
82 function DoValidation(input){
83     var avalue = eval(input.validate)(input);
84     avalue.field = input.field;
85     return { validation: input.contactid, runform: [avalue] };
86 }
87
88
89 function WorkById(input) {
90     var id = input.id;
91     for (var i=0; i<checkedCommands.length; i++)
92         if (checkedCommands[i].id==id){
93             return ExecuteCommand(checkedCommands[i], input);
94         }
95 }
96
97 function getDeepLink(command){
98     var domainName = X.DOMAINNAME;
99     return "https://" + domainName + ".oncloud.gr/s360?job="+encodeURIComponent(
        command.command) + "&caption=" + encodeURIComponent(command.caption);
100 }
101
102 //Fill form in order to propose options in menu jobs
103 function MenuForm(choices){
104     var optionform = {
105         attachments: [
106             {
107                 contentType: "application/vnd.microsoft.card.hero",
108                 content: {
109                     text: menuFormtext[lang],
110                     buttons: choices
111                 }
112             }
113         ]
114     }
115 }
116 botstate.job = "question";
117 return optionform;

```

```

118 }
119
120 function CreateObj(menu){
121     var caption = menu.caption[0];
122     caption = caption.split(">");
123     var size = caption.length-1,
124     obj= {};
125     obj.type = "openUrl";
126     obj.title = caption[size];
127     obj.value = getDeepLink(menu);
128     return obj;
129 }
130 }
131
132 function sql2json(sql)
133 {
134     var ds = X.GETSQLDATASET(sql, null);
135     if (ds.RECORDCOUNT>0){
136         return eval(ds.JSON);
137     }else{
138         return null;
139     }
140 }
141
142 function sql2array(sql)
143 {
144     var ds = X.GetSqlDataSet(sql, null);
145     var reccount = ds.RecordCount;
146     var fldcount = ds.FieldCount;
147     var result = [];
148     for (var i = 0; i<reccount; i++)
149     {
150         ds.RecNo = i+1;
151         var rec = [];
152         for (var j = 0; j<fldcount; j++)
153             rec[j] = ds.Fields(j);
154         result[i] = rec;
155     }
156     return result;
157 }
158
159 //Find date
160 function Today(x){
161     var d = new Date();
162     if (x!=null){
163         d.setDate(d.getDate() + x);
164     }
165     var day = d.getDate();
166     if (day<10){
167         day= "0"+day;
168     }
169     var month = d.getMonth();
170     month+=1;
171     if (month<10){
172         month = "0"+month;
173     }
174     var year = d.getFullYear().toString();
175     var dateStr = year+month+day ;
176     return dateStr;
177 }

```

```

178
179
180
181 function formatDate(input){
182     var result = {},
183         day = "",
184         month = "",
185         year = "",
186         d = input.text,
187         regex = /^[0-9]{1,2}[/\-\-][0-9]{1,2}[/\-\-]{0,1}[0-9]{0,4}$/g,
188         res = d.match(regex);
189
190
191     if (!res){
192         result.prompt = formatDatePrompt[lang];
193         result.type = "option";
194         result.options = "";
195     }
196 }
197 else {
198     result.type = "text";
199     var fdate1 = d.split("/"),
200         fdate2 = d.split("-")
201     if (fdate1.length==1){
202         fdate = fdate2;
203     }
204     else {
205         fdate = fdate1;
206     }
207     if (fdate[0].length==1){
208         day = "0"+fdate[0];
209     }
210     else {
211         day = fdate[0];
212     }
213     if (fdate[1].length==1){
214         month = "0"+fdate[1];
215     }
216     else if(fdate[1].length==2){
217         month = fdate[1];
218     }
219     else {
220         result.prompt = formatDatePrompt[lang];
221         result.type = "option";
222         result.options = "";
223         return result;
224     }
225     if (fdate[2]){
226         if (fdate[2].length==4){
227             year = fdate[2];
228         }
229         else{
230             result.prompt = formatDatePrompt[lang];
231             //result.type = "option";
232             //result.options = "";
233             result.success = false;
234             result.type = "text"
235             return result;
236         }
237     }

```

```

238     else{
239         var d      = new Date();
240         var year1   = d.getFullYear().toString();
241         year       = year1;
242     }
243     result.success = true;
244     result.day = day;
245     result.month = month;
246     result.year = year;
247 }
248 return result;
249 }
250
251
252 function dateAdeies(input){
253
254     var result = formatDate(input);
255     if(result.day){
256         result.value = ",'" + result.year + "-" + result.month + "-" + result.day + "
257             00:00:00'";
258     }
259     return result;
260 }
261
262 function dateDb(input){
263
264     var result = formatDate(input);
265     if(result.day){
266         result.value = result.year + result.month + result.day;
267     }
268     return result;
269 }
270
271 //Set time for notification
272 function setTime(hour){
273     if (hour==1){
274         botstate.time = 1;
275     }
276     else if(hour==2){
277         botstate.time = 2;
278     }
279     else if(hour==3){
280         botstate.time = 3;
281     }
282     else if(hour==4){
283         botstate.time = 4;
284     }
285 }
286
287 //Badwords
288 function badWords(str){
289     if (str){
290         var a = str.split(" ");
291         for (var i=0; i<a.length;i++){
292             if(a[i]=="BADWORD"){
293                 botstate.bad = true;
294                 var word=a[i];
295                 if (word==botstate.input){
296                     botstate.badcount=1;

```

```

297     }
298   }
299 }
300 if (botstate.bad){
301   botstate.input=word;
302 }
303 }
304
305
306 function createApprv (obj,fld){
307   var key    = obj[1],
308       field  = fld[0].split("."),
309       value  = fld[1],
310       object,
311       checkForm = obj[0].split("."),
312       result;
313
314   var SoObj = {};
315   if (checkForm.length==2){
316     SoObj.FORM = checkForm[1];
317     object = checkForm[0];
318   }
319   else{
320     object = obj[0];
321   }
322
323   SoObj.SERVICE = "setData";
324   SoObj.KEY = key;
325   SoObj.OBJECT = object;
326
327   SoObj.DATA = {};
328   if(field[0]!=null && field[1]!=null){
329     SoObj.DATA[field[0]] = [];
330
331     var rec = {};
332     rec[field[1]] = value
333     SoObj.DATA[field[0]].push(rec);
334
335     result = X.WEBREQUEST(JSON.stringify(SoObj));
336     result = JSON.parse(result);
337     return result;
338   }
339   else return {success : false}
340 }
341 }
342
343 //check if the job is question
344 function toMain(commandNum, input){
345   for(var k=0;k<checkedCommands[commandNum].title.length;k++){
346     var score = Checkinit(checkedCommands[commandNum].title[k], input.
347                           text);
348     if(score==0){
349       botstate.job = "question";
350       fromJob= true;
351       return botstate.job
352     }
353   }
354 }
355 }

```

Listing 4: Basic functions needed