



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

**Ανάπτυξη Ευφυούς Πράκτορα Συνομιλίας (Chatbot) για
τις ανάγκες των μεταπτυχιακών φοιτητών**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Δήμητρα Καλομοίρη
Αρ. Μητρώου: 72020**

Επιβλέπων : Κασδερίδης Στάθης
Ακαδημαϊκός Υπότροφος, ΔΙ.ΠΑ.Ε.

Επιβλέπων : Σταμάτης Δημοσθένης
Καθηγητής, ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Φεβρουάριος 2022



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB
INTELLIGENCE

Ανάπτυξη Ευφυούς Πράκτορα Συνομιλίας (Chatbot) για τις ανάγκες των μεταπτυχιακών φοιτητών

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΑ ΚΑΛΟΜΟΪΡΗ

Επιβλέπων : Κασδερίδης Στάθης
Ακαδημαϊκός Υπότροφος, ΔΙ.ΠΑ.Ε.

Επιβλέπων : Σταμάτης Δημοσθένης
Καθηγητής, ΔΙ.ΠΑ.Ε.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Όνομα Επώνυμο
Choose an item. ΔΙ.ΠΑ.Ε.

.....
Όνομα Επώνυμο
Choose an item. ΔΙ.ΠΑ.Ε.

.....
Όνομα Επώνυμο
Choose an item. ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Φεβρουάριος 2022

(Υπογραφή)

.....

Δήμητρα Καλομοίρη

© 2022– Allrightsreserved

Περίληψη

Τα τελευταία χρόνια, έχει παρατηρηθεί εκθετική ανάπτυξη των chatbot. Αποτελούν εργαλεία λογισμικού τα οποία συνομιλούν με χρήστες, είτε μέσω γραπτού λόγου (δηλαδή κειμένου), είτε μέσω φωνητικών εντολών. Παρέχουν πληροφορίες απαντώντας σε ερωτήματα και εκτελούν διάφορες εργασίες. Χρησιμοποιούνται σε ποικίλους τομείς, όπως για παράδειγμα στην εκπαίδευση, στην εξυπηρέτηση, στην υποστήριξη, στην ψυχαγωγία.

Η παρούσα διπλωματική, διερευνά τις διαθέσιμες τεχνικές για την ανάπτυξη ενός chatbot. Παρουσιάζεται η ανάπτυξη, αναλύοντας την σχεδίαση και την κατασκευή ενός ψηφιακού βοηθού, του WiBot (Web Intelligence ChatBot). Το chatbot που δημιουργήθηκε, απευθύνεται σε φοιτητές του μεταπτυχιακού προγράμματος «Ευφυείς Τεχνολογίες Διαδικτύου» του Διεθνούς Πανεπιστημίου της Ελλάδος και στόχο έχει την υποστήριξή τους, παρέχοντας πληροφορίες σε οποιοδήποτε στάδιο της φοίτησής τους (από την εισαγωγή έως την αποφοίτησή τους). Το chatbot μιλάει και κατανοεί την ελληνική γλώσσα. Επιπλέον, για την εκπόνηση της εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python και το Rasa Framework.

Σε πρώτο στάδιο, έγινε η μελέτη πεδίου, όπου αναλύθηκαν έννοιες γύρω από τα chatbot. Έπειτα, αναζητήθηκαν οι διαθέσιμες τεχνικές, επιλέγοντας την καταλληλότερη για τη συγκεκριμένη ανάπτυξη. Το στάδιο της υλοποίησης περιγράφεται λεπτομερώς, καθώς και η αξιολόγηση του διαλογικού πράκτορα που έγινε σε δύο φάσεις, με τις προτάσεις βελτίωσης.

Λέξεις Κλειδιά: Chatbot, Rasa Framework, Python, WiBot

Abstract

The last few years, we have seen an exponential growth of AI chatbots. Chatbot is a computer program that communicates with users, through text or through voice commands. They provide information by answering questions and perform various tasks. They are used in fields such as education, service, support, entertainment.

This dissertation, explores the available techniques for developing a chatbot. The development analysis, design and construction process of a digital assistant, WiBot (Web Intelligence ChatBot), is presented. Chatbot, is addressed to students, of the postgraduate program "Intelligent Internet Technologies" of the Hellenic University, needs and aims to support them, by providing information at any stage of their studies (from admission to graduation). Also, WiBot can understand and converse in Greek language. For its development we use the Python programming language and the Rasa Framework.

In the first stage, we explore ideas from the literature,, where concepts around chatbots are analyzed. Then, available techniques are evaluated, selecting the most suitable for the specific development. The implementation is described in details, as well as the assessment of the interactive agent, with suggestions for improvement.

Keywords: Chatbot, Rasa Framework, Python, WiBot

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος Ευφυείς Τεχνολογίες Διαδικτύου (MSc in Web Intelligence) του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος. Ως ελάχιστη δυνατή μνεία, οφείλω να ευχαριστήσω όλους όσους συνέβαλαν στην εκπόνησή της και ιδιαίτερα τον επιβλέποντα καθηγητή μου, κύριο Στάθη Κασδερίδη, για την άψογη συνεργασία, την αμέριστη υποστήριξή του αλλά και την διάθεσή του να βοηθήσει και να με καθοδηγήσει από την αρχή μέχρι το τέλος. Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου, για την υποστήριξή τους καθώς και για την συνεχή συμπαράσταση.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Ανασκόπηση βιβλιογραφίας.....	1
1.1.1	Περιγραφή προβλήματος.....	1
1.1.2	Τεχνική προσέγγιση.....	2
1.1.3	Δομή εργασίας.....	3
1.2	Σύντομη περιγραφή.....	4
1.2.1	<i>Pandorabots</i>	5
1.2.2	<i>Syn Bot Studio</i>	6
1.2.3	<i>Rasa</i>	6
2	Βασικές Έννοιες.....	8
2.1	Έννοιες Τεχνολογιών.....	8
2.1.1	<i>Artificial Intelligence</i>	8
2.1.2	<i>Machine learning</i>	9
2.1.3	<i>Natural Language Processing</i>	10
2.2	Chatbots.....	11
2.2.1	Ορισμοί Chatbots.....	11
2.2.2	Ιστορία.....	12
2.2.3	Είδη chatbot.....	13
2.2.4	Πλεονεκτήματα και Μειονεκτήματα.....	14
3	Μεθοδολογία.....	15
3.1	Επιλογή πλατφόρμας ανάπτυξης.....	15
3.2	Rasa Framework.....	24
3.2.1	<i>Installation</i>	25
3.2.2	<i>Δομή και Βασικά Αρχεία</i>	26
3.2.3	<i>Άλλες χρήσιμες εντολές</i>	32
4	Υλοποίηση.....	33
4.1	Ανάλυση Απαιτήσεων.....	33

4.2	Μεθοδολογία σχεδίασης	35
4.3	Διαγράμματα.....	36
4.4	Ανάπτυξη	41
4.4.1	Δομή.....	42
4.4.2	Custom Actions	43
4.4.3	Fallback- Χειρισμός μηνυμάτων εκτός πεδίου εφαρμογής.....	51
4.4.4	Index.html	51
4.4.5	Socket.IO.....	55
4.4.6	Προβλήματα.....	57
4.4.7	To WiBot	58
5	Αξιολόγηση.....	74
5.1	Acceptance testing	74
5.2	Πειραματική αξιολόγηση (χρήστες)	79
5.2.1	A' Φάση Αξιολόγησης	79
5.2.2	B' Φάση Αξιολόγησης	82
5.3	Συμπεράσματα αξιολόγησης.....	83
6	Επίλογος	85
6.1	Συμπεράσματα	85
6.2	Προτάσεις βελτίωσης και μελλοντική έρευνα	86
	Βιβλιογραφία.....	87
	ΠΑΡΑΡΤΗΜΑ Α : User Stories	89
	ΠΑΡΑΡΤΗΜΑ Β : Evaluation Results (1)	97
	ΠΑΡΑΡΤΗΜΑ Γ: Evaluation Results (2).....	101

Κατάλογος Εικόνων

Εικόνα 1.1.1 User stories – Λίστα.....	2
Εικόνα 1.2.1 Παράθυρο διαλόγου - Pandorabots.....	5
Εικόνα 1.2.2 Παράθυρο διαλόγου - Syn Bot Studio	6
Εικόνα 1.2.3 Παράθυρο διαλόγου - Rasa X.....	6
Εικόνα 2.2.1 Chatbot.....	12
Εικόνα 2.2.2 Ιστορική εξέλιξη chatbots [11]	13
Εικόνα 3.1.1 User Story 1- Μαθήματα εξαμήνου	16
Εικόνα 3.1.2 User Story 3 - Στοιχεία επικοινωνίας Καθηγητή	16
Εικόνα 3.1.3 Συνδυαστικό.....	16
Εικόνα 3.1.4 Μαθήματα εξαμήνου - Pandorabots.....	17
Εικόνα 3.1.5 Στοιχεία επικοινωνίας Καθηγητή – Pandorabots	18
Εικόνα 3.1.6 Παράδειγμα AIML.....	18
Εικόνα 3.1.7 Ερώτηση χρήστη - Syn Bot Studio	19
Εικόνα 3.1.8 Επιλογή εξαμήνου - Syn Bot Studio	19
Εικόνα 3.1.9 Απάντηση chatbot - Syn Bot Studio	20
Εικόνα 3.1.10 Παράδειγμα SIML	21
Εικόνα 3.1.11 Συνδυαστικό -Rasa.....	21
Εικόνα 3.1.12 Παράδειγμα Rasa	22
Εικόνα 3.1.13 Πίνακας Αξιολόγησης - excel.....	24
Εικόνα 3.2.1 Rasa Framework	24
Εικόνα 3.2.2 Αρχιτεκτονική Rasa [16].....	25
Εικόνα 3.2.3 Φάκελος data.....	26
Εικόνα 3.2.4 Αρχείο nlu.yml	27
Εικόνα 3.2.5 Αρχείο rules.yml	27
Εικόνα 3.2.6 Αρχείο stories.yml.....	28
Εικόνα 3.2.7 Αρχείο stories.yml (2)	28
Εικόνα 3.2.8 Άλλα αρχεία	28
Εικόνα 3.2.9 Αρχείο config (Policies).....	29
Εικόνα 3.2.10 Αρχείο config (Pipeline)	30
Εικόνα 3.2.11 Αρχείο credentials.yml.....	30
Εικόνα 3.2.12 Αρχείο domain.yml	31
Εικόνα 3.2.13 slots	31
Εικόνα 3.2.14 Αρχείο endpoints.yml.....	31

Εικόνα 3.2.15 Φάκελος models	32
Εικόνα 3.2.16 Αρχείο actions.py	32
Εικόνα 4.1.1 Εκδόσεις.....	34
Εικόνα 4.3.1 Διάγραμμα ενεργειών.....	37
Εικόνα 4.3.2 Αρχιτεκτονική WiBot	38
Εικόνα 4.3.3 Διάγραμμα activity “Αποστολή Email”	38
Εικόνα 4.3.4 Διάγραμμα activity “menu”	39
Εικόνα 4.3.5 Διάγραμμα activity “Help-menu”	39
Εικόνα 4.3.6 Διάγραμμα activity “Μαθήματα”.....	40
Εικόνα 4.3.7 Διάγραμμα activity “Ωρολόγιο Πρόγραμμα”	40
Εικόνα 4.3.8 Διάγραμμα activity “Ανακοινώσεις”	41
Εικόνα 4.4.1 Intents.....	42
Εικόνα 4.4.2 Stories.....	42
Εικόνα 4.4.3 Terminal run actions	43
Εικόνα 4.4.4 Παράδειγμα custom action.....	43
Εικόνα 4.4.5 ActionSubmit	45
Εικόνα 4.4.6 ActionSubmitWithFile	46
Εικόνα 4.4.7 ActionResetEmailSlots	46
Εικόνα 4.4.8 ActionReadFile	47
Εικόνα 4.4.9 ActionProfessorInfo	47
Εικόνα 4.4.10 ActionEduPeople	48
Εικόνα 4.4.11 ActionAnnouncementsAPI	48
Εικόνα 4.4.12 ActionCategoriesAll.....	49
Εικόνα 4.4.13 ActionAnnouncementPerCat.....	50
Εικόνα 4.4.14 ActionCourses	50
Εικόνα 4.4.15 Παραδείγματα	51
Εικόνα 4.4.16 Αρχική σελίδα.....	52
Εικόνα 4.4.17 Αρχική σελίδα με ανοιχτό διάλογο	52
Εικόνα 4.4.18 Navigation menu	53
Εικόνα 4.4.19 Footer	53
Εικόνα 4.4.20 Σελίδα.....	54
Εικόνα 4.4.21 Media queries	55
Εικόνα 4.4.22 credentials.yml	55
Εικόνα 4.4.23 Δήλωση script	56
Εικόνα 4.4.24 Παραμετροποίηση Web chat.....	56
Εικόνα 4.4.25 Μεγιστοποίηση παραθύρου	57
Εικόνα 4.4.26 Κλείσιμο-Ελαχιστοποίηση παραθύρου	57

Εικόνα 4.4.27 Αρχικό μήνυμα WiBot.....	58
Εικόνα 4.4.28 Ερώτηση τύπου “help”.....	59
Εικόνα 4.4.29 Μενού επιλογών.....	59
Εικόνα 4.4.30 Μήνυμα out of scope	60
Εικόνα 4.4.31 Μήνυμα «Waiting for server».....	60
Εικόνα 4.4.32 Ερώτηση «Μαθήματα»	61
Εικόνα 4.4.33 Απάντηση WiBot για το 1 ^ο εξάμηνο.....	61
Εικόνα 4.4.34 Διάλογος «Διαδικασία ορκωμοσίας».....	62
Εικόνα 4.4.35 Διάλογος «Erasmus».....	62
Εικόνα 4.4.36 Ερώτηση «Πρόσβαση στην σχολή».....	63
Εικόνα 4.4.37 Διάλογος «Πρόγραμμα μερικής φοίτησης»	63
Εικόνα 4.4.38 Διάλογος «Έκδοση φοιτητικής ταυτότητας»	64
Εικόνα 4.4.39 Διάλογος «Επικοινωνία με την γραμματεία».....	64
Εικόνα 4.4.40 Διάλογος «Ωρολόγιο πρόγραμμα».....	65
Εικόνα 4.4.41 Διάλογος για αποστολή αρχείου σε email.....	65
Εικόνα 4.4.42 Απεσταλμένο email (1)	66
Εικόνα 4.4.43 Απεσταλμένο email (2)	66
Εικόνα 4.4.44 Διάλογος «Ανακοινώσεις Τμήματος».....	67
Εικόνα 4.4.45 Κατηγορίες ανακοινώσεων	67
Εικόνα 4.4.46 Παράδειγμα εμφάνισης 1 ^{ης} Ανακοίνωσης.....	68
Εικόνα 4.4.47 Διάλογος «Αίτηση εισαγωγής».....	68
Εικόνα 4.4.48 Διάλογος «Ηλεκτρονικές υπηρεσίες»	69
Εικόνα 4.4.49 Διάλογος «Πλατφόρμα e-learning»	69
Εικόνα 4.4.50 Διάλογος «Πληροφορίες για το Τμήμα και την σχολή»	70
Εικόνα 4.4.51 Διάλογος «Τρόποι επικοινωνίας διδάσκοντες (Λίστα Τμήματος)»	70
Εικόνα 4.4.52 Διάλογος «Υποτροφίες».....	71
Εικόνα 4.4.53 Διάλογος «Υποβολή αιτήσεων».....	71
Εικόνα 4.4.54 Διάλογος «Ωρες γραφείου».....	72
Εικόνα 4.4.55 Διάλογος «Έκδοση Πιστοποιητικού»	73
Εικόνα 5.1.1 test_stories.yml	75
Εικόνα 5.1.2 Φάκελος results	75
Εικόνα 5.1.3 Intent_errors.json	76
Εικόνα 5.1.4 DIETClassifier_errors.json	77
Εικόνα 5.1.5 story_report.json	78
Εικόνα 5.2.1 Εκτέλεση εντολής	79
Εικόνα 5.2.2 Requests	79
Εικόνα 5.2.3 Φόρμα Αξιολόγησης (A’)	81

Εικόνα 5.2.4 Φόρμα Αξιολόγησης (B').....82

Κατάλογος Πινάκων

Πίνακας 1.2.1 Εργαλεία που υποστηρίζουν τα ελληνικά.....	4
Πίνακας 3.1.1 Εργαλεία και πλεονεκτήματα-μειονεκτήματα	22
Πίνακας 3.2.1 Βασικές έννοιες Rasa.....	26
Πίνακας 4.4.1 Custom Actions.....	44
Πίνακας 5.2.1 Ερωτήσεις αξιολόγησης.....	80

Συντομογραφίες

ΔΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
ΠΜΣ	Πρόγραμμα Μεταπτυχιακών Σπουδών
ΑΙ	Artificial Intelligence
TN	Τεχνητή Νοημοσύνη
NLP	Natural Language Processing
NLU	Natural Language Understanding
US	User Story

1

Εισαγωγή

Στο κεφάλαιο αυτό καταγράφεται η ανασκόπηση της βιβλιογραφίας, περιγράφοντας το πρόβλημα, την τεχνική προσέγγιση, αλλά και την δομή της εργασίας. Στο υποκεφάλαιο της σύντομης περιγραφής, γίνεται μια εισαγωγική μελέτη σχετικά την με τα διαθέσιμα εργαλεία.

1.1 Ανασκόπηση βιβλιογραφίας

1.1.1 Περιγραφή προβλήματος

Η μελέτη της παρούσας διπλωματικής εργασίας έγινε εστιάζοντας σε προβλήματα και απαιτήσεις των φοιτητών σχετικά με θέματα που προκύπτουν στην καθημερινότητά τους. Παρέχει ποικίλες πληροφορίες, προσφέροντας την δυνατότητα στους υποψήφιους φοιτητές να ξεκινήσουν με πλήρη κάλυψη των βασικών ερωτημάτων τους, αλλά φυσικά απευθύνεται στους φοιτητές του ΠΜΣ, δίνοντας πληροφορίες για την καλύτερη περάτωση των σπουδών τους. Επιπλέον, αποτελεί ένα υποστηρικτικό εργαλείο της γραμματείας του τμήματος, μιας και μπορεί να δώσει απαντήσεις σε συχνά και κοινά ερωτήματα.

Βασικό χαρακτηριστικό της εργασίας αυτής, είναι ότι συνδυάζει διαφορετικές πληροφορίες από διαφορετικά συστήματα. Συγκεκριμένα, περιέχει πληροφορίες από τον επίσημο ισότοπο του ΠΜΣ, της σχολής αλλά και από το διαθέσιμο API ((Web) Application Programming Interface) του Τμήματος, από το οποίο συλλέξαμε πληροφορίες όπως οι ανακοινώσεις και στοιχεία επικοινωνίας καθηγητών.

Στόχο έχει την διερεύνηση των προβλημάτων, αλλά και την επίλυσή τους, στα πλαίσια της Τεχνητής Νοημοσύνης και ειδικότερα αξιοποιώντας την τεχνολογία των chatbots (διαλογικών πρακτόρων).

1.1.2 Τεχνική προσέγγιση

Η παρούσα εργασία, χωρίστηκε σε δύο υπο-έργα (projects). Το αρχικό “pilot project”, στο οποίο έγινε μια σύντομη υλοποίηση, αλλά με μεγάλη συμβολή στην τελική επιλογή του κατάλληλου εργαλείου και το “main project” που αφορά την ανάπτυξη της εφαρμογής του chatbot, δηλαδή το τελικό παραδοτέο.

Όπως αναφέρθηκε για την μελέτη και την κατανόηση των τεχνολογιών, έγινε η υλοποίηση ενός pilot project. Στόχος ήταν η έρευνα των διαθέσιμων chatbots και η επιλογή του καταλληλότερου. Για τον σκοπό αυτό, δημιουργήθηκαν τρία *user stories*, εφαρμόζοντάς τα σε τρία διαφορετικά εργαλεία, τα οποία αναλύονται παρακάτω.

Η ιστορία χρήστη (**user story**) είναι ένα εργαλείο, στην ανάπτυξη λογισμικού μεθοδολογίας Agile, που χρησιμοποιείται για την καταγραφή ενός λειτουργικού χαρακτηριστικού (feature) της εφαρμογής, από την οπτική γωνία ενός χρήστη. Κάθε user story περιγράφει τον τύπο του χρήστη, τι θα ήθελε σαν προσθήκη και γιατί. Αποτελεί μια απλοποιημένη περιγραφή μιας απαίτησης [2].

Στο ακόλουθο στιγμιότυπο (Εικόνα 1.1.1) παρουσιάζεται ο πίνακας με τα καταγεγραμμένα user stories. Αφορά, την συνολική και τελική του έκδοση, ωστόσο να σημειωθεί ότι σε αυτό το στάδιο μας ενδιαφέρουν τα τρία πρώτα (US1- US3).

Σχετικά με τις πληροφορίες που περιέχει, στην πρώτη στήλη “Story ID” ορίζεται μια αρίθμηση, στην δεύτερη με τίτλο “User Story” αποτυπώνεται η περιγραφή του κάθε story και

Story ID	User Story	Project
1	Σαν φοιτητής, θέλω να μάθω ποια είναι τα μαθήματα ανά εξάμηνο (χειμερινό, εαρινό), για να προετοιμαστώ κατάλληλα	pilot & main
2	Σαν φοιτητής, θέλω να μάθω τον τον τρόπο με τον οποίο θα διεξαχθεί η εξεταστική του Σεπτεμβρίου, για να προετοιμαστώ κατάλληλα	pilot
3	Σαν φοιτητής, θέλω να μάθω τον τον τρόπο επικοινωνίας με τον καθηγητή Καθηγητή, ώστε να επικοινωνήσω άμεσα μαζί του	pilot
4	Σαν φοιτητής, θέλω να μάθω τους διαθέσιμους τρόπους επικοινωνίας με το Τμήμα Φοιτητικής Μέριμνας, ώστε να επικοινωνήσω άμεσα μαζί τους	main
5	Σαν φοιτητής, θέλω να μάθω πόσα μαθήματα μπορώ να πάρω στο πρόγραμμα μερικής φοίτησης, ώστε να γνωρίζω τον ελάχιστο αριθμό μαθημάτων.	main
6	Σαν φοιτητής, θέλω να μάθω τους τρόπους επικοινωνίας με την γραμματεία/ διεύθυνση, για άμεση επικοινωνία.	main
7	Σαν φοιτητής, θέλω να μάθω πότε ξεκινάει το χειμερινό/εαρινό εξάμηνο, για να οργανώσω το υπόλοιπο πρόγραμμά μου.	main
8	Σαν φοιτητής, θέλω να μάθω τις τελευταίες ανακοινώσεις του τμήματος για να ενημερωθώ για τα τελευταία νέα, σεμινάρια, ημερομηνίες υποβολής (top 5)	main
9	Σαν φοιτητής, θέλω να μάθω τα απαραίτητα έγγραφα για την υποβολή αίτησης στο πρόγραμμα, ώστε να ολοκληρώσω σωστά την αίτησή μου	main
10	Σαν φοιτητής, θέλω να μάθω ποια είναι η πλατφόρμα e-learning, για να την εγκαταστήσω/ προετοιμαστώ	main
11	Σαν φοιτητής, χρειάζομαι ένα Πιστοποιητικό / Αποδεικτικό Εγγραφής, για να το αποστείλω στην εταιρία μου ή όπου μου ζητηθεί	main
12	Σαν φοιτητής, θέλω να μάθω το site για την έκδοση της φοιτητικής ταυτότητας, για να ακολουθήσω την αντίστοιχη διαδικασία έκδοσης	main
13	Σαν φοιτητής, θέλω να μάθω ποιες είναι οι ηλεκτρονικές υπηρεσίες (webmail, pithia ..), για να γνωρίζω τις ιστοσελίδες που χρειάζομαι κατά την διάρκεια των σπουδών	main
14	Σαν φοιτητής, θέλω να μάθω ποιοι είναι οι τρόποι πρόσβασης στην σχολή (διεύθυνση, λεωφορείο), για να μπορέσω να κατευθυνθώ σωστά	main
15	Σαν φοιτητής, θέλω να μάθω πληροφορίες για το erasmus, για να αποφασίσω εάν θα προβώ σε αίτηση	main
16	Σαν φοιτητής, θέλω να μάθω πληροφορίες για την διαδικασία της ορκωμοσίας (χαρτιά, προϋποθέσεις), για να υποβάλω έγκαυρα τα κατάλληλα έγγραφα	main
17	Σαν φοιτητής, θέλω να μάθω τους τρόπους επικοινωνίας με τους Διδάσκοντες, για άμεση επικοινωνία (λίστα)	main
18	Σαν φοιτητής, θέλω να μάθω πότε ολοκληρώνονται οι αιτήσεις για τους νέους υποψήφιους, για να ολοκληρώσω την υποβολή μου σωστά και έγκαιρα.	main
19	Σαν φοιτητής, θέλω να μάθω τις τελευταίες ανακοινώσεις του τμήματος, για να ενημερωθώ για τα τελευταία νέα, σεμινάρια, ημερομηνίες υποβολής (top 10)	main
20	Σαν φοιτητής, θέλω να μάθω τον τον τρόπο επικοινωνίας με τον/την καθηγητή/καθηγήτρια, ώστε να επικοινωνήσω άμεσα μαζί του	main
21	Σαν φοιτητής, θέλω να μάθω τι πληροφορίες μπορεί να προσφέρει το chatbot, ώστε να γνωρίζω τι μπορώ να ρωτήσω	main
22	Σαν φοιτητής, θέλω να μάθω το ωρολόγιο πρόγραμμα, για να σχεδιάσω το πρόγραμμά μου	main
23	Σαν φοιτητής, θέλω να δω τον οδηγό σπουδιών, για να αποκτήσω πρόσβαση σε όλες τις βασικές πληροφορίες του τμήματος	main
24	Σαν φοιτητής, θέλω να μάθω το ακαδημαϊκό ημερολόγιο, για να σχεδιάσω το πρόγραμμά μου	main
25	Σαν φοιτητής, θέλω να μάθω πληροφορίες για το τμήμα και την σχολή (γενικά), για να γνωρίζω γενικότερες πληροφορίες για το τμήμα	main
26	Σαν χρήστης του συστήματος, θέλω να μπορώ να αποστείλω email μέσα από την εφαρμογή, για να αποστείλω κάποιες από τις προσφερόμενες πληροφορίες	main
27	Σαν φοιτητής, θέλω να μάθω για τις υποτροφίες που δίνονται, για να βρω πληροφορίες	main
28	Σαν φοιτητής, θέλω να μάθω τις ώρες γραφείου των καθηγητών, για να γνωρίζω πότε μπορώ να επικοινωνήσω μαζί τους	main

Εικόνα 1.1.1 User stories – Λίστα

στην τελευταία “Project”, ο τύπος του project όπου θα υλοποιηθεί το κάθε ένα. Το πεδίο Project, περιλαμβάνει δύο τύπους pilot και main. Στο pilot project περιέχονται τα πρώτα τεστ σενάρια που υλοποιήθηκαν κατά την διερευνητική φάση, ενώ στο main project αντιστοιχούν τα σενάρια της κύριας υλοποίησης.

Στην επόμενη φάση, έγινε εκτενής μελέτη των διαθέσιμων εργαλείων και με βάση τις προδιαγραφές της διπλωματικής, επιλέχθηκαν τρία από αυτά. Εδώ, έγινε η υλοποίηση των user stories, συλλέγοντας διάφορες πληροφορίες κατά την ανάπτυξή τους. Η διαδικασία καθώς και τα αποτελέσματα περιγράφονται λεπτομερώς στα επόμενα κεφάλαια.

Ακολούθησε ο σχεδιασμός και φυσικά η υλοποίηση του main project, που περιγράφεται στην συνέχεια.

1.1.3 Δομή εργασίας

Η παρούσα εργασία έχει αναπτυχθεί σε επτά κεφάλαια. Το πρώτο κεφάλαιο αποτελεί την εισαγωγή, διευκρινίζοντας βασικές έννοιες και προετοιμάζοντας τον αναγνώστη για το κύριο μέρος της εργασίας. Περιλαμβάνει την ανασκόπηση βιβλιογραφίας και την σύντομη περιγραφή όπου παρουσιάζονται οι διαθέσιμες πλατφόρμες. Το δεύτερο κεφάλαιο, αποτελεί μια εισαγωγή στις βασικές έννοιες γύρω από τα chatbots. Επίσης, περιλαμβάνει ένα υποκεφάλαιο για τα chatbots, με ορισμούς, ιστορική αναδρομή, τα είδη, πλεονεκτήματα και μειονεκτήματα χρήσης.

Το τρίτο κεφάλαιο, περιλαμβάνει την μεθοδολογία. Εδώ, παρουσιάζεται η μελέτη και η διαδικασία για την επιλογή της πλατφόρμας ανάπτυξης, πληροφορίες για την επιλεγμένη πλατφόρμα, δηλαδή το Rasa. Το τέταρτο, αποτελεί το κύριο μέρος της εργασίας, μιας και περιγράφει την υλοποίηση της εφαρμογής και αποτελείται από την ανάλυση απαιτήσεων, μεθοδολογία σχεδίασης, τα διαγράμματα για οπτική αναπαράσταση ορισμένων διαδικασιών και την τελική ανάπτυξη. Ακολουθεί το πέμπτο κεφάλαιο με την αξιολόγηση του συστήματος αλλά και τα αποτελέσματα. Τέλος, το έκτο κεφάλαιο, αποτελεί τον επίλογο με τα συμπεράσματα και τις προτάσεις βελτίωσης. Ακόμη, μετά την παράθεση της βιβλιογραφίας ακολουθούν τα παραρτήματα. Στο Παράρτημα Α *User Stories*, γίνεται εκτενής αναφορά στα user stories που χρησιμοποιήθηκαν, καταγράφοντας δηλαδή τις απαντήσεις που αντιστοιχούν σε κάθε ένα αντίστοιχα. Το Παράρτημα Β, παρουσιάζει τα αποτελέσματα της πρώτης αξιολόγησης (Α΄ Φάση), με οπτική αναπαράσταση των απαντήσεων, μέσω γραφημάτων. Επιπλέον, το Παράρτημα Γ περιέχει τα αποτελέσματα της δεύτερης αξιολόγησης (Β΄ Φάση) πάλι σε μορφή γραφημάτων, ανά ερώτηση.

1.2 Σύντομη περιγραφή

Στην ανάπτυξη ενός chatbot, το πρώτο στάδιο περιέχει την αναζήτηση και επιλογή του εργαλείου ανάπτυξης. Σε μια σύντομη αναζήτηση, μπορεί κάποιος να παρατηρήσει ότι υπάρχει τεράστια πληθώρα εργαλείων και συστημάτων. Όμως, για την τελική επιλογή, θα πρέπει να ληφθούν υπόψη διαφορετικοί παράμετροι και κριτήρια ανάλογα με τις απαιτήσεις.

Αρχικά λοιπόν, γίνεται διερεύνηση των διαθέσιμων εργαλείων για την υλοποίηση της εφαρμογής του chatbot. Στην συγκεκριμένη διπλωματική εργασία, επιλέχθηκε η ανάπτυξη των διαλόγων στην ελληνική γλώσσα. Για το σκοπό αυτό, έγινε κατάλληλη αναζήτηση και συγκεντρώθηκαν εργαλεία που υποστηρίζουν τα ελληνικά. Η αναζήτηση αυτή, είχε ως αποτέλεσμα τον αποκλεισμό αρκετών εργαλείων, ωστόσο βρέθηκαν έξι, από τα οποία ορισμένα είναι αρκετά διαδεδομένα. Στον πίνακα που ακολουθεί, καταγράφονται συγκεντρωτικά τα εργαλεία αυτά, καθώς επίσης και ορισμένα χαρακτηριστικά τους.

Πίνακας 1.2.1 Εργαλεία που υποστηρίζουν τα ελληνικά

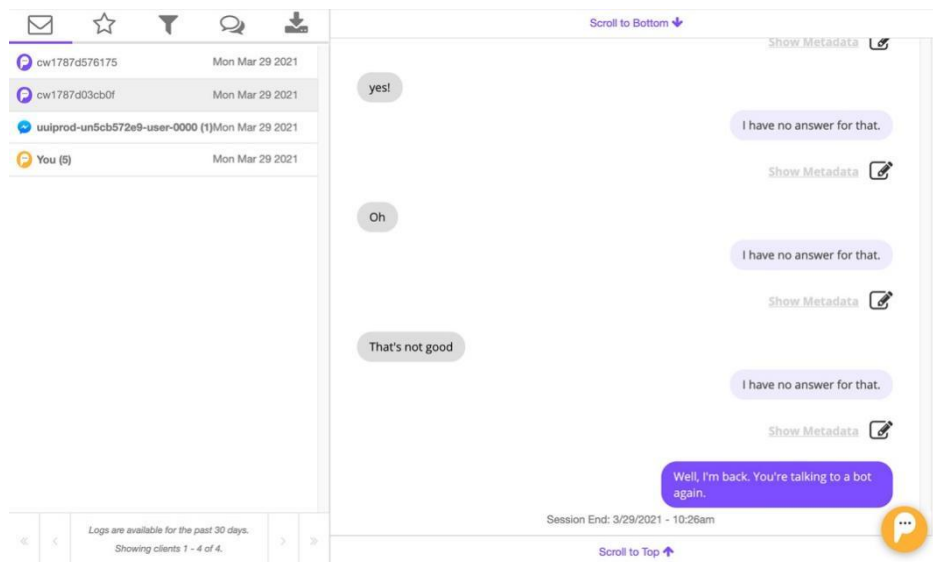
Πλατφόρμα	Γλώσσα	Χαρακτηριστικά
Pandorabots	AIML (βασίζεται σε XML)	Η δωρεάν έκδοση διαθέτει περιορισμένες δυνατότητες
Syn Bot Studio	SIML	Προσφέρει βασικές δυνατότητες Ανάπτυξη για desktops, κινητές συσκευές και ιστοσελίδες (web)
Rasa	Python	Αρκετά ισχυρό και διαδεδομένο σε προγραμματιστές παγκοσμίως
wit.ai	Python, Ruby	NLP support, Δωρεάν
Chatfuel	code-free	Δεν απαιτεί προγραμματιστικές γνώσεις Μπορεί να ενσωματωθεί μόνο στο Facebook Messenger
Flow XO	code-free	Περιορισμένο λογισμικό επεξεργασίας φυσικής γλώσσας

		Επί πληρωμή δίνει περισσότερες δυνατότητες
--	--	--

Στην συνέχεια, έγινε επιλογή των τριών επικρατέστερων, τα οποία μελετήθηκαν σε θεωρητικό αλλά κυρίως σε πρακτικό επίπεδο. Από την διαδικασία αποκλείστηκαν τα Chatfuel και Flow XO μιας και δεν χρησιμοποιούν κάποια γλώσσα προγραμματισμού για την υλοποίηση του διαλογικού πράκτορα. Αποτελούν εύκολη λύση για κάποιον που θέλει να δημιουργήσει ένα chatbot, χωρίς την συγγραφή κώδικα. Το wit.ai απορρίφθηκε, γιατί σε πρώτη φάση παρουσιάζει δυσκολία στην κατανόηση, η υποστήριξη που παρέχεται είναι περιορισμένη και επιπλέον δεν δημιούργησε κάποιο ενδιαφέρον για περαιτέρω ενασχόληση. Καταλληλότερα, κρίθηκαν τα Pandorabots, Syn Bot Studio και Rasa Framework για τα οποία παρουσιάζονται πληροφορίες παρακάτω.

1.2.1 Pandorabots

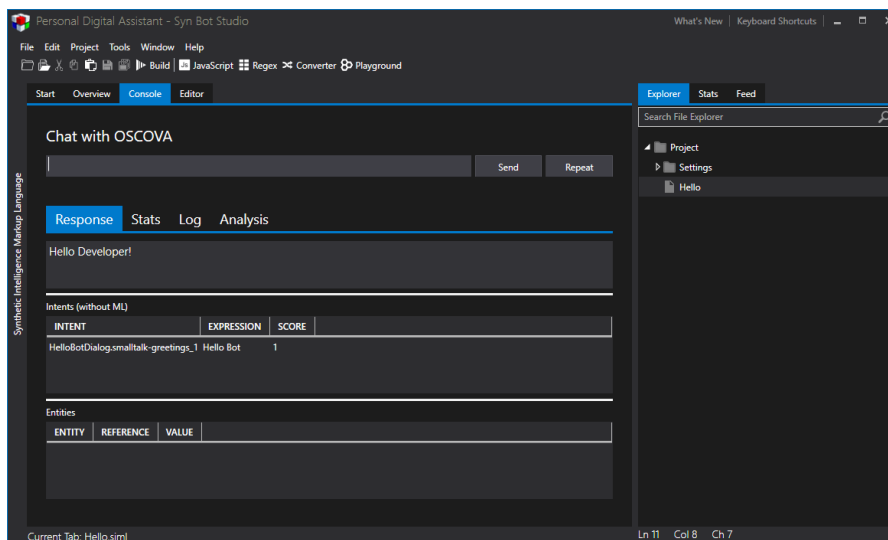
Η πλατφόρμα Pandorabots αποτελεί ένα open-source framework, που επιτρέπει στους χρήστες του να δημιουργούν chatbots που υποστηρίζουν ΑΙ σε ιστό, εφαρμογές για κινητά και εφαρμογές ανταλλαγής μηνυμάτων όπως το Messenger, WhatsApp και Telegram. Η υλοποίηση γίνεται με την χρήση της Artificial Intelligence Markup Language (AIML), μια scripting γλώσσα που βασίζεται σε XML. Αξίζει να σημειωθεί, ότι έχουν δημιουργηθεί πάνω από 300.000 chatbots στην πλατφόρμα Pandorabots, μέχρι και σήμερα [2].



Εικόνα 1.2.1 Παράθυρο διαλόγου - Pandorabots
(Πηγή: <https://chatimize.com/reviews/pandorabots>)

1.2.2 Syn Bot Studio

Το Syn Bot Studio είναι ένα IDE που προσφέρεται για ανάπτυξη SIML Bot. Αποτελεί ένα πλήρες εργαλείο ανάπτυξης σε SIML - Synthetic Intelligence Markup Language. Προσφέρει επιλογές όπως Code Analysis, Regular Expression, Tester, Script Evaluator και Converter (από AIML σε SIML) [3].

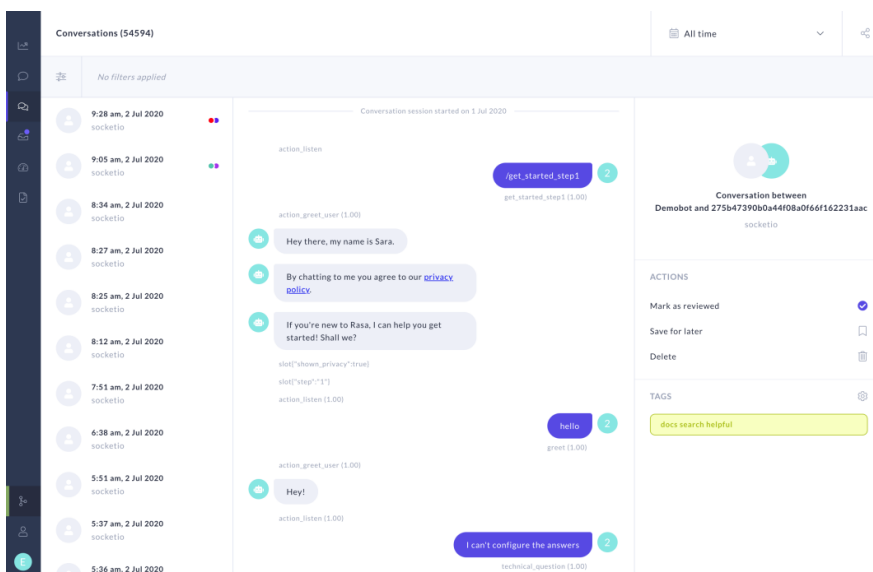


Εικόνα 1.2.2 Παράθυρο διαλόγου - Syn Bot Studio

(Πηγή: <https://developer.syn.co.in/tutorial/bot-studio/creating-project.html>)

1.2.3 Rasa

Το Rasa είναι ένα open source machine learning framework, για τη δημιουργία AI βοηθών (assistants) και chatbots. Δεν απαιτείται ιδιαίτερη εμπειρία σε γλώσσες προγραμματισμού, αν και υπάρχει ο "Rasa Action Server" όπου χρειάζεται υλοποίηση κώδικα σε Python.



Εικόνα 1.2.3 Παράθυρο διαλόγου - Rasa X

(Πηγή: <https://rasa.com/docs/rasa-x/user-guide/review-conversations/>)

Διαθέτει δύο κύρια modules, το *Rasa NLU* για την κατανόηση μηνυμάτων του χρήστη και το *Rasa Core* για τη διεξαγωγή συνομιλιών και την απόφαση για το τι θα κάνει στη συνέχεια. Επιπλέον, υπάρχει το *Rasa X* ένα συμπληρωματικό εργαλείο που βοηθά στην δημιουργία, την βελτίωση και την ανάπτυξη βοηθών AI που υποστηρίζονται από το *Rasa Core*. Το *Rasa X* περιλαμβάνει διεπαφή χρήστη (UI) όπου μπορούν οι χρήστες να συνομιλήσουν με το chatbot και με βάση την συζήτηση (Conversation-Driven Development, CDD) το bot βελτιώνεται. [4]

2

Βασικές Έννοιες

Στο Κεφάλαιο αυτό γίνεται ανάλυση των βασικότερων εννοιών που θα μελετηθούν στην παρούσα διπλωματική εργασία. Πιο συγκεκριμένα, στο πρώτο μέρος δίνονται διάφοροι ορισμοί, όπως της Τεχνητής Νοημοσύνης, απαραίτητοι για την πλήρη κάλυψη του θέματος. Στο δεύτερο μέρος, δίνονται ορισμοί για το chatbot, παρουσιάζεται σύντομα η ιστορική αναδρομή και στην συνέχεια τα είδη που συναντάμε καθώς και πλεονεκτήματα και μειονεκτήματα χρήσης.

2.1 Έννοιες Τεχνολογιών

Πρώτου αναφερθούμε εκτενέστερα στα chatbots είναι απαραίτητο να διευκρινίσουμε βασικές έννοιες που περιβάλλουν τους διαλογικούς πράκτορες. Στις υποενότητες που ακολουθούν, δίνονται σύντομοι ορισμοί και πληροφορίες σχετικά με την Τεχνητή Νοημοσύνη, την Μηχανική Μάθηση και την Επεξεργασία Φυσικής Γλώσσας.

2.1.1 Artificial Intelligence

Πρόκειται για μια από τις πιο μεγάλες τάσεις στον χώρο των ψηφιακών τεχνολογιών, με εφαρμογές σε ποικίλους τομείς. Βρίσκεται παντού γύρω μας επηρεάζοντας σε καθημερινή βάση την ζωή μας και τις αποφάσεις μας. Αναζητώντας έναν ορισμό για την Τεχνητή Νοημοσύνη, μπορεί κανείς να βρει πολλές προσεγγίσεις και εστιάζοντας σε διαφορετικές σκοπιές.

Ο Τζον Μακάρθι (1956) όρισε τον τομέα αυτόν ως «επιστήμη και μεθοδολογία της δημιουργίας νοημόνων μηχανών».

Σύμφωνα με τους Barr και Feigenbaum (1981), Artificial Intelligence (AI) ή Τεχνητή Νοημοσύνη (TN) είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζουμε με τη νοημοσύνη στην ανθρώπινη συμπεριφορά.

Ένας άλλος ορισμός που συναντάμε στο [5], είναι: TN είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφών (νοημόνων) υπολογιστικών συστημάτων, που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς τα οποία υπονοούν έστω και στοιχειώδη ευφυΐα: μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, κατανόηση από συμφραζόμενα, επίλυση προβλημάτων κλπ.

Οι περιοχές της TN αφορούν την Επίλυση προβλημάτων, Απόδειξη Θεωρημάτων, Επεξεργασία Φυσικής Γλώσσας, Τεχνητή Όραση, Μηχανική Μάθηση, Σχεδιασμός Ενεργειών και Χρονοπρογραμματισμός, Αυτόνομα Ρομπότ, Ευφυείς πράκτορες και άλλες.

Πατέρας της TN θεωρείται ο Alan Turing, ο οποίος εμπνεύστηκε το γνωστό Turing Test, περισσότερες λεπτομέρειες ακολουθούν στο Κεφάλαιο 2.2.2

2.1.2 Machine learning

Μια εξέλιξη της Τεχνητής Νοημοσύνης (TN) είναι η Μηχανική Μάθηση ή Machine learning (ML). Μηχανική Μάθηση είναι η συλλογή αλγορίθμων και μεθόδων με τις οποίες βελτιώνεται η αποδοτικότητα μιας μηχανής (για παράδειγμα υπολογιστής) στην εκτέλεση «ευφών» εργασιών.

Σύμφωνα με τον Mitchell (1997) ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του σε εργασίες από το T , όπως μετρείται από το P , βελτιώνεται μέσω της εμπειρίας E .

Γενικότερα, μπορεί να βοηθήσει στη λήψη ορθότερων αποφάσεων σε επιχειρήσεις, οργανισμούς, Κυβερνήσεις κ.α. αφού μπορεί να αναλύσει τα δεδομένα που ακόμη και οι καλύτεροι αναλυτές θα δυσκολευόταν. Μέσω δημιουργίας αλγορίθμων και χωρίς να δίνεται κάποια συγκεκριμένη λειτουργία για τον τρόπο με τον οποίο θα λειτουργήσουν, μπορούν να εξάγουν αναλυτικά αποτελέσματα. Μπορεί να χρησιμοποιηθεί για ανάλυση κειμένων, ήχου, εικόνας ή ακόμη και βίντεο.

Οι κυριότεροι τύποι Αλγορίθμων μηχανικής μάθησης είναι οι παρακάτω:

Εποπτευόμενη Μάθηση (Supervised Learning): Οι αλγόριθμοι εδώ μαθαίνουν από παραδείγματα δεδομένων ή σχετικές αποκρίσεις με ετικέτες, και αργότερα γίνεται πρόβλεψη με τη χρήση νέων παραδειγμάτων.

Μη εποπτευόμενη μάθηση (Unsupervised Learning): Οι αλγόριθμοι εδώ μαθαίνουν από παραδείγματα χωρίς όμως αυτή τη φορά να υπάρχει κάποια σχετική απόκριση. Ο αλγόριθμος βρίσκει από μόνος του και κατηγοριοποιεί τα δεδομένα.

Ενισχυμένη (Reinforcement Learning): Οι αλγόριθμοι εκπαιδεύονται συνεχώς. Μονίμως γίνονται δοκιμές και σφάλματα. Αποκτά έτσι εμπειρία, μαθαίνει από τα λάθη του και προσπαθεί στο μέλλον να πάρει τις σωστές αποφάσεις.

Βασικές κατηγορίες εργασιών:

- **Αναγνώριση προτύπων (Pattern Recognition).** Είναι η διαδικασία κατά την οποία μέσω αλγορίθμων επιτυγχάνεται η αναγνώριση των προτύπων. Σύμφωνα με τη γνώση που έχει αποκτήσει μπορεί και κατηγοριοποιεί τα δεδομένα που θα του δοθούν δίνοντας συγχρόνως και την πιθανότητα το συγκεκριμένο πρότυπο να ανήκει στη συγκεκριμένη κλάση. Καθώς τα δεδομένα που του δίνονται μπορεί να είναι σε διαφορετική μορφή από αυτήν που κατανοεί ένα υπολογιστικό σύστημα, π.χ. βίντεο, εικόνες, φωνή, κάνει τη μετατροπή σε μορφή κατανοητή από τους υπολογιστές και έπειτα γίνεται η επεξεργασία.
- **Εξαγωγή χαρακτηριστικών (Feature Extraction).** Πολλές φορές τα δεδομένα που εισέρχονται για ανάλυση στο σύστημα παρουσιάζουν πολυπλοκότητα. Υπάρχουν φορές που μπορούμε να πάρουμε τα αποτελέσματα που θέλουμε χωρίς να χρειάζεται να πάρουμε όλη την πληροφορία. Για παράδειγμα εάν θέλουμε να επεξεργαστούμε εικόνα μπορούμε να τη συμπίεσουμε και να την επεξεργαστούμε χωρίς να χάσουμε ουσιαστική πληροφορία.
- **Πρόβλεψη (Prediction) και Παλινδρόμηση (Regression).** Μία από τις πιο κοινές τεχνικές πρόβλεψης στη μηχανική μάθηση είναι η τεχνική της Παλινδρόμησης. Μπορούμε για παράδειγμα από κάποιες αλλαγές τιμών μεταβλητών που χαρακτηρίζουν την ατμόσφαιρα να προβλέψουμε ένα φαινόμενο που πιθανόν θα συμβεί.
- **Ομαδοποίηση (Clustering).** Η ομαδοποίηση είναι η διαδικασία κατά την οποία τα δεδομένα χωρίζονται σε ομάδες βάση της ομοιότητας τους ή κοινά σημεία που παρουσιάζουν. Για παράδειγμα μπορούμε να δώσουμε φωτογραφίες από γάτες και σκύλους και να χωρίσει τις εικόνες σε 2 κλάσεις (ομάδες).
- **Γενίκευση (Generalization).** Αφού εκπαιδεύσουμε ένα μοντέλο, αυτό μετά θα είναι σε θέση να γενικεύσει. Δηλαδή εφόσον έχουμε εκπαιδεύσει ένα μοντέλο πρέπει αυτό μετά να είναι σε θέση σε περίπτωση που του δώσουμε ένα νέο (και άγνωστο μέχρι εκείνη τη στιγμή) δεδομένο να μπορέσει να το βάλει στην κατάλληλη κλάση.

2.1.3 Natural Language Processing

Η επεξεργασία φυσικής γλώσσας (natural language processing-NLP), σύμφωνα με την IBM, αναφέρεται στον κλάδο της επιστήμης των υπολογιστών και πιο συγκεκριμένα, στον κλάδο της Τεχνητής Νοημοσύνης, που ασχολείται με την ικανότητα των υπολογιστών να κατανοούν το κείμενο αλλά και τον προφορικό λόγο, με τον ίδιο σχεδόν τρόπο όπως οι άνθρωποι [6].

Οι ερευνητές, στην προσπάθεια τους να αναπτύξουν καλύτερα εργαλεία και τεχνικές, συλλέγουν γνώσεις για το πώς τα ανθρώπινα όντα κατανοούν και χρησιμοποιούν τη γλώσσα.

Στόχος είναι να κάνουν τα συστήματα υπολογιστών να κατανοούν και να χειρίζονται τις φυσικές γλώσσες ώστε να εκτελούν τις επιθυμητές εργασίες. Στις εφαρμογές του NLP συναντάμε διάφορα πεδία, όπως η αυτόματη μετάφραση, η επεξεργασία και σύνοψη κειμένου σε φυσική γλώσσα, οι διεπαφές χρήστη (user interfaces), η πολυγλωσσική και διαγλωσσική ανάκτηση πληροφοριών (CLIR), η αναγνώριση ομιλίας, η τεχνητή νοημοσύνη και τα έμπειρα συστήματα.[7]

Επιπρόσθετα, πολλές φορές συμβαίνει ένας άλλος όρος, ο NLU (Natural Language Understanding) να συγχέεται λανθασμένα με τον όρο NLP. Η κατανόηση φυσικής γλώσσας (Natural Language Understanding-NLU) αποτελεί ένα συστατικό ή ορθότερα ένα υποσύνολο του NLP. Πιο συγκεκριμένα, αναλαμβάνει την κατανόηση του κειμένου, εκείνου που πληκτρολογεί ο χρήστης και προσπαθεί να το ταξινομήσει στις σωστές προθέσεις (intents). Σύμφωνα με το [8] *“η κατανόηση φυσικής γλώσσας χρησιμοποιεί συντακτική και σημασιολογική ανάλυση κειμένου (γραπτού ή προφορικού λόγου), για να προσδιορίσει το νόημα μιας πρότασης. Η σύνταξη αναφέρεται στη γραμματική δομή μιας πρότασης, ενώ η σημασιολογία παραπέμπει στο επιδιωκόμενο νόημά της.”*

Αξίζει να αναφέρουμε άλλο ένα υποσύνολο του NLP, το NLG (Natural Language Generation). Η Παραγωγή Φυσικής Γλώσσας (Natural Language Generation-NLG) επιτρέπει στους υπολογιστές να γράφουν. Αναλαμβάνει την διαδικασία παραγωγής μιας απάντησης κειμένου, σε ανθρώπινη γλώσσα, που βασίζεται σε κάποια εισαγωγή δεδομένων. Τα τελευταία χρόνια τα συστήματα παραγωγής φυσικής γλώσσας έχουν εξελιχθεί, επιτρέποντας δυναμική παραγωγή κειμένου σε πραγματικό χρόνο [8].

2.2 Chatbots

Σε αυτό το κεφάλαιο παρουσιάζονται γενικές πληροφορίες για τα chatbots - διαλογικούς πράκτορες, όπως ορισμοί, ιστορική εξέλιξη, κατηγορίες και πλεονεκτήματα - μειονεκτήματα χρήσης.

2.2.1 Ορισμοί Chatbots

Για τον όρο chatbot υπάρχουν πολλές κοινές εκφράσεις όπως διαλογικοί πράκτορες, ψηφιακοί βοηθοί, chatbots, chat bots, smart bots, interactive agents άλλα και άλλοι. Μία γενικότερη διατύπωση τι ορίζεται ως chatbot, είναι λογισμικό με το οποίο συνομιλεί κάποιος χρήστης online, μέσω γραπτού ή προφορικού λόγου, δίνοντας αυτοματοποιημένες απαντήσεις. Είναι ένα πρόγραμμα υπολογιστή που έχει σχεδιαστεί για να προσομοιώνει



Εικόνα 2.2.1 Chatbot

(Πηγή: <https://tinyurl.com/w42kku2>)

συνομιλίες με χρήστες (ανθρώπους), ειδικά μέσω διαδικτύου (Αγγλικό Λεξικό της Οξφόρδης).

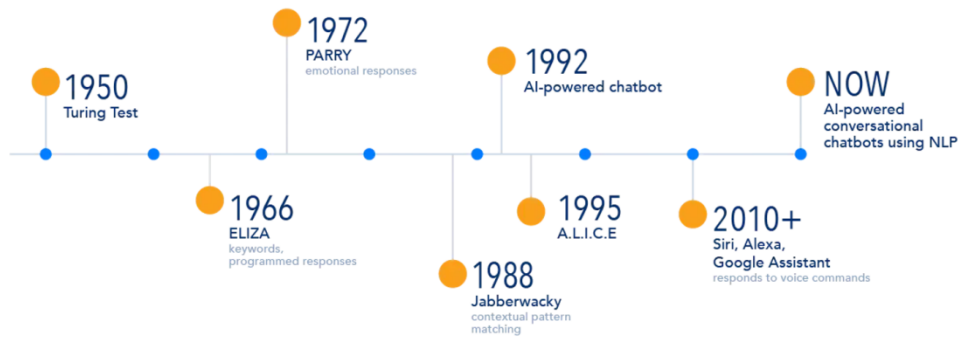
Από την άλλη, η IBM [9], ορίζει ως chatbot, ένα πρόγραμμα υπολογιστή που χρησιμοποιεί Τεχνητή Νοημοσύνη (AI) και Επεξεργασία Φυσικής Γλώσσας (NLP) για να κατανοήσει τις ερωτήσεις των πελατών και να αυτοματοποιήσει τις απαντήσεις σε αυτές, προσομοιώνοντας την ανθρώπινη συνομιλία.

Σύμφωνα με την Oracle [10], πρόκειται για ένα πρόγραμμα (υπολογιστή) που προσομοιώνει και επεξεργάζεται ανθρώπινη ομιλία είτε γραπτή είτε προφορική. Επιτρέπει στους ανθρώπους να αλληλεπιδρούν με ψηφιακές συσκευές, όπως επικοινωνούν με έναν άλλο άνθρωπο. Υπάρχουν chatbots που απαντούν σε ένα απλό ερώτημα, αλλά και πιο εξελιγμένα, όπως οι ψηφιακοί βοηθοί, που μαθαίνουν και εξελίσσονται, συλλέγοντας πληροφορίες.

Γενικά τα chatbots χρησιμοποιούνται για διασκέδαση, έχουν εφαρμογή σε επιχειρήσεις, στην εκπαίδευση, στο ηλεκτρονικό εμπόριο και σε άλλους τομείς.

2.2.2 Ιστορία

Το 1950, ο Άλαν Τούρινγκ έθεσε ένα από τα σημαντικότερα ερωτήματα για την επιστήμη των υπολογιστών, αν μια μηχανή μπορεί να μιμηθεί την ανθρώπινη σκέψη, το περίφημο Turing Test. Το τεστ επιτυγχάνει εάν ο χρήστης πειστεί ότι τα μηνύματα που δέχεται, δεν προέρχονται από υπολογιστή, αλλά από άνθρωπο. Για πολλούς αυτή η ερώτηση ήταν η βασική ιδέα που γέννησε τα chatbots. Το πρώτο γνωστό chatbot συστήθηκε στο κοινό ως Eliza και κατασκευάστηκε το 1966. Στόχος του ήταν να λειτουργεί ως ψυχοθεραπευτής, δίνοντας ως απαντήσεις τις εκφράσεις του χρήστη αλλά σε μορφή ερώτησης. Χρησιμοποίησε αντιστοίχιση προτύπων και οι απαντήσεις δίνονταν βάση ενός σχήματος από templates. Με περιορισμένες γνώσεις, ικανότητες και χωρίς δυνατότητα εκμάθησης, ήταν η αρχή για την ανάπτυξη των διαλογικών πρακτόρων. Το 1972 ήρθε ένα βελτιωμένο chatbot, το PARRY.



Εικόνα 2.2.2 Ιστορική εξέλιξη chatbots [11]

Προσομοιάζε έναν ασθενή με σχιζοφρένεια, εκφράζοντας συναισθήματα κατά την διάρκεια της συζήτησης. Αυτή ήταν και η μεγάλη διαφορά σε σχέση με το Eliza, αφού διέθετε προσωπικότητα. Παρόλα αυτά, δεν είχε πολλές ικανότητες και επίσης δεν μπορούσε να μάθει.

Το A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), δημιουργήθηκε το 1995 και κέρδισε βραβείο στο ετήσιο τεστ Turing. Χρησιμοποιεί την AIML, μια γλώσσα βασισμένη σε XML που σχεδιάστηκε για τη δημιουργία chatbots. Βασίστηκε στην αντιστοίχιση προτύπων, όπως το Eliza, αλλά έχοντας διαφορετική δυναμική αφού διαθέτει περίπου 41.000 πρότυπα και μοτίβα.

Το 2000 περίπου δημιουργήθηκε ένα διαφορετικό chatbot το SmarterChild. Ένα από τα δημοφιλέστερα μέχρι την εμφάνιση του Siri, καθώς ήταν διαθέσιμο σε MSN, Yahoo Messenger κ.α.

Ακολούθησαν οι έξυπνοι ψηφιακοί βοηθοί φωνής, όπως η Siri της Apple, αντικαθιστώντας τις γραπτές συνομιλίες με φωνητικούς διαλόγους. Η πρωτοποριακή αυτή προσέγγιση έδωσε το βήμα για να αναπτυχθούν οι πιο διαδεδομένοι ψηφιακοί βοηθοί όπως IBM Watson, Google Assistant, Microsoft Cortana και Amazon Alexa [12].

2.2.3 Είδη chatbot

Τα chatbots μπορούν να ταξινομηθούν σε ποικίλες κατηγορίες. Σε γενικότερα πλαίσια, τοποθετούνται με βάση τους στόχους σε *Task-oriented chatbots* και *Non-task-oriented chatbots* [5]. Η πρώτη κατηγορία αφορά τα chatbots που έχουν σχεδιαστεί για συγκεκριμένες διεργασίες και έχουν ρυθμιστεί για να εκτελούν μικρές και σύντομες συνομιλίες. Χρησιμοποιούνται για εξυπηρέτηση και υποστήριξη πελατών και απαντούν κυρίως κοινές ερωτήσεις ή εκτελούν βασικές και όχι περίπλοκες διαδικασίες. Η δεύτερη και πιο ανοιχτή κατηγορία, περιλαμβάνει chatbots που χαρακτηρίζονται *chit-chat bots*. Σκοπό έχουν να παρουσιάσουν ενδιαφέρον και να έχουν έναν πιο ψυχαγωγικό χαρακτήρα κατά την συνομιλία τους με τον χρήστη. Θεωρούνται πιο δημιουργικά και οι συνομιλίες αυτές δεν έχουν κάποιο συγκεκριμένο στόχο.

Πιο συχνά όμως συναντάμε την προσέγγιση που παρουσιάζεται στο [13], η οποία τα κατατάσσει σε τρεις βασικές κατηγορίες.

Menu/Button-Based Chatbots: Ένας από τους πιο διαδεδομένους τύπους chatbot στην αγορά και ο πιο απλούστερος. Βασίζεται σε μενού επιλογών, με καθορισμένες απαντήσεις και ερωτήσεις υπό την μορφή κουμπιών (buttons). Ακολουθεί τον σχεδιασμό των δένδρων αποφάσεων, με κάθε επιλογή του χρήστη να οδηγεί στην τελική απάντηση. Συνήθως, είναι πιο αργά σε σχέση με άλλα και θεωρείται περιοριστικό αφού αποτελείται από περιορισμένες ερωτήσεις-απαντήσεις.

Keyword Recognition-Based Chatbots: Αναγνωρίζει συγκεκριμένες λέξεις-κλειδιά από την είσοδο του χρήστη και με την βοήθεια της Τεχνητής Νοημοσύνης (Natural Language Processing), εξάγει- δίνει την κατάλληλη απάντηση. Αποτυγχάνουν συνήθως όταν διατυπώνονται ίδιες λέξεις-κλειδιά σε παρόμοιες ερωτήσεις. Επιπλέον, συνδυάζεται με τον παραπάνω τύπο δημιουργώντας έτσι ένα υβριδικό μοντέλο.

Contextual Chatbots: Ένα από τα πιο προηγμένα τεχνολογικά είδη chatbots, χρησιμοποιώντας τεχνολογίες Μάθησης Μηχανής και Τεχνητής Νοημοσύνης με σκοπό να αναγνωρίσουν την πρόθεση του χρήστη και τελικά να δώσουν την πιο εύστοχη απάντηση. Μαθαίνουν και βελτιώνονται συνεχώς μετά από κάθε συνομιλία και μπορούν να προτείνουν απαντήσεις με βάση την γνώση που έχουν χτίσει, είτε από τον συγκεκριμένο χρήστη είτε από το σύνολο.

2.2.4 Πλεονεκτήματα και Μειονεκτήματα

Το βασικότερο πλεονέκτημα των chatbot είναι ότι παρέχουν άμεση και βέλτιστη εμπειρία εξυπηρέτησης. Με συνεχή υποστήριξη 24/7, καθώς και εξοικονόμηση κόστους, η εφαρμογή τους σε έναν οργανισμό δημιουργεί ανταγωνιστικά οφέλη. Επιπλέον, εξελίσσονται με βάση την εμπειρία που αποκτούν, μπορούν να δώσουν πληροφορίες για τους πελάτες που εξυπηρετούν, βοηθώντας τον οργανισμό να καταλάβει περισσότερο τις ανάγκες τους [14].

Από την άλλη, όσο αφορά τα μειονεκτήματα, πολλές φορές μπορεί να δημιουργηθεί σύγχυση και τελικά να προσφέρουν λάθος πληροφορία στο χρήστη. Ακόμα, το chatbot δεν μπορεί να ταυτιστεί με την ανθρωπινή εξυπηρέτηση. Και τέλος χρειάζονται αρκετά συχνά συντήρηση και βελτιστοποίηση συμπληρώνοντας νέα δεδομένα ή διαφοροποιώντας τα υπαρκτά. [15]

3

Μεθοδολογία

Το κεφάλαιο τρία, περιέχει την μεθοδολογία, με την διαδικασία επιλογής της πλατφόρμας ανάπτυξης και αναλυτικότερες πληροφορίες για την εγκατάσταση και την δομή της επιλεγμένης πλατφόρμας.

3.1 Επιλογή πλατφόρμας ανάπτυξης

Μετά την επιλογή των επικρατέστερων εργαλείων, Pandorabots, Syn Bot Studio, και Rasa Framework δημιουργήθηκαν τρία παραδείγματα διαλόγων (user stories). Υλοποιήθηκαν και στα τρία περιβάλλοντα ώστε να αποκτήσουμε μία πλήρη εικόνα και στην συνέχεια να γίνει η καταλληλότερη επιλογή. Στις παρακάτω εικόνες αποτυπώνονται ξεχωριστά οι διάλογοι που υλοποιήθηκαν.

Το πρώτο user story *Μαθήματα εξάμηνου* (Εικόνα 3.1.1), αποτελεί μια δημόσια συζήτηση, η οποία ανήκει στην κατηγορία τύπου public. Πιο συγκεκριμένα, το chatbot μπορεί να απαντήσει σε τέτοια ερωτήματα χωρίς να χρειαστεί ταυτοποίηση του χρήστη.

User: Καλησπέρα!
ChatBot: Καλησπέρα, πως μπορώ να βοηθήσω;
User: Θα ήθελα να μάθω ποια είναι τα μαθήματα ανά εξάμηνο.
ChatBot: Για το χειμερινό ή το εαρινό εξάμηνο ενδιαφέρεστε;
User: Και για τα δύο./ Για το χειμερινό./ Για το εαρινό.
ChatBot: Τα μαθήματα του (πχ) χειμερινού εξαμήνου είναι:
1. Μεθοδολογίες Έρευνας
2. Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές
3. Ασφάλεια Διαδικτυακών Πληροφοριακών Συστημάτων
4. Συστήματα Κινητού και Διάχυτου Υπολογισμού
5. Μηχανική Μάθηση
User: Ευχαριστώ

Εικόνα 3.1.1 User Story 1- Μαθήματα εξαμήνου

User: Καλησπέρα!
ChatBot: Καλησπέρα, πως θα μπορούσα να σας εξυπηρετήσω;
User: Θα ήθελα να μάθω τους τρόπους επικοινωνίας με τον καθηγητή/καθηγήτρια Καθηγητή.
ChatBot: Για να σας προωθήσω προσωπικά στοιχεία, θα πρέπει να γνωρίζω τον αριθμό του μητρώου σας.
User: AM 72020
ChatBot: Το τηλέφωνο επικοινωνίας είναι: 2234034498 και το mail: example@ihu.gr
User: Ευχαριστώ!
ChatBot: Μπορώ να σε βοηθήσω σε κάτι άλλο;
User: Όχι
ChatBot: Στην διάθεσή σου!
User: Ευχαριστώ!

Εικόνα 3.1.2 User Story 3 - Στοιχεία επικοινωνίας Καθηγητή

User: Καλησπέρα!
ChatBot: Καλησπέρα, πως θα μπορούσα να σας εξυπηρετήσω;
User: Θα ήθελα να μάθω πως θα διεξαχθεί η εξέταση για το Μάθημα τον Σεπτέμβριο.
ChatBot: Παρακαλώ πείτε μου τον αριθμό του μητρώου σας.
User: 72020
ChatBot: Η εξέταση θα διεξαχθεί εξ αποστάσεως, μέσω της πλατφόρμας zoom.
User: Ευχαριστώ!
ChatBot: Μπορώ να σε βοηθήσω σε κάτι άλλο;
User: Θα ήθελα να μάθω τους τρόπους επικοινωνίας με τον κ. Καθηγητή
ChatBot: Το τηλέφωνο επικοινωνίας είναι: 2234034498 και το mail: example@ihu.gr
User: Ευχαριστώ!

Εικόνα 3.1.3 Συνδυαστικό

Το δεύτερο user story, αφορά το user story με Id 3 και τίτλο *Στοιχεία επικοινωνίας Καθηγητή*. Δίνει στον χρήστη πληροφορίες σχετικά με του τρόπους επικοινωνίας με έναν υποθετικό καθηγητή του Πανεπιστημίου, τον “Καθηγητή”.

Η τρίτη υλοποίηση με τίτλο *Συνδυαστικό*, αποτελεί έναν συνδυασμό του *Στοιχεία επικοινωνίας Καθηγητή (User Story 3)* με ένα ακόμη σενάριο που αφορά τα *Εξέταση Μάθημα Σεπτεμβρίου (User Story 2)*. Εδώ το chatbot πρέπει να απαντήσει σε περισσότερα ερωτήματα. Όπως παρατηρείται, οι διάλογοι στα user stories *Στοιχεία επικοινωνίας Καθηγητή* και

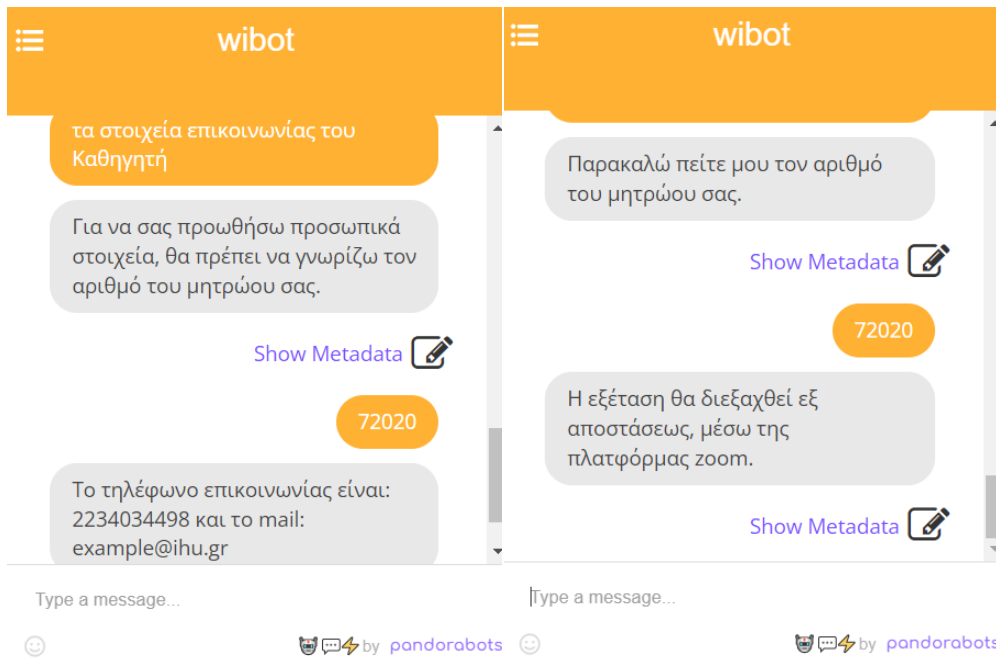
Συνδυαστικό (Εικόνα 3.1.2 και Εικόνα 3.1.3 αντίστοιχα) αποτελούνται από ιδιωτικές συζητήσεις, ανήκουν δηλαδή στην κατηγορία private. Για να απαντήσει το chatbot στα ερωτήματα που ανήκουν σε έναν τέτοιο τύπο διαλόγου, θα πρέπει να έχει προηγηθεί μια διαδικασία ταυτοποίησης. Συγκεκριμένα, ο χρήστης θα πρέπει να δώσει έναν έγκυρο αριθμό μητρώου για να λάβει την επιθυμητή απάντηση.

Ακολουθούν ορισμένα στιγμιότυπα μετά την ολοκλήρωση της εκάστοτε υλοποίησης.

Συγκεκριμένα όσο αφορά το Pandorabots, παρουσιάζουμε τα user stories 1 και 2. Επιπλέον, παραθέτουμε δείγμα κώδικα AIML.



Εικόνα 3.1.4 Μαθήματα εξαμήνου - Pandorabots



Εικόνα 3.1.5 Στοιχεία επικοινωνίας Καθηγητή – Pandorabots

Στην AIML, για να υλοποιήσουμε κάθε σενάριο, ορίσαμε πολλά category. Καθένα από αυτά αντιπροσωπεύει ένα κομμάτι γνώσης και περιλαμβάνει οπωσδήποτε ένα pattern και ένα template. Το pattern καθορίζει το μοτίβο που αντιστοιχεί με αυτό που μπορεί να εισάγει ο χρήστης. Το template αποτελεί την απάντηση του bot στο αντίστοιχο input του χρήστη.

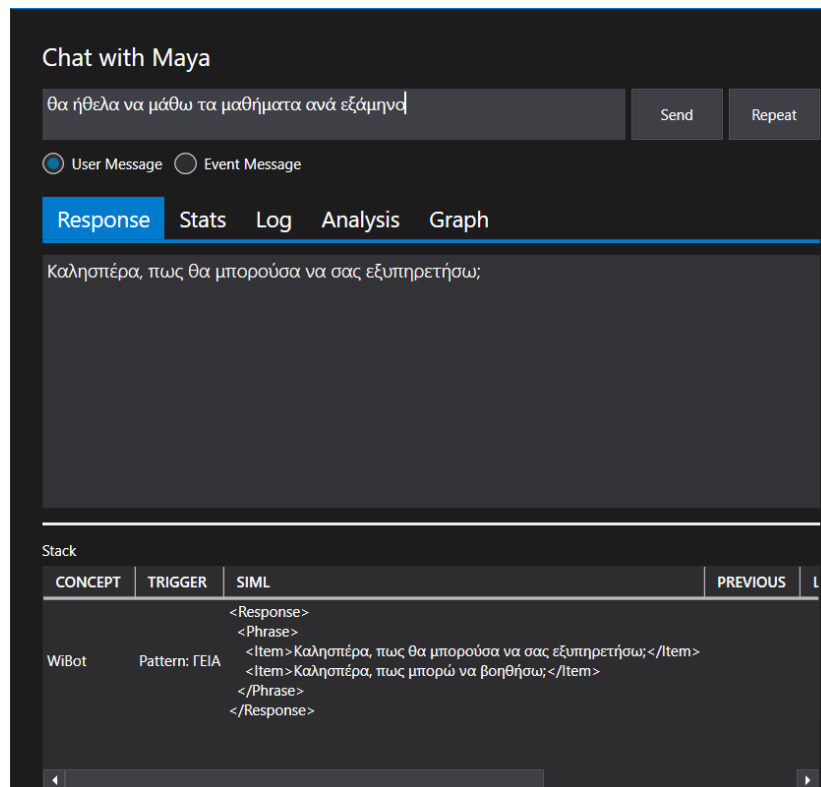
```

<?xml version="1.0" encoding="UTF-8"?>
<aiml>
  <category>
    <pattern>Γεια</pattern>
    <template>
      <random>
        <li> Καλησπέρα </li>
        <li> Γεια σου! </li>
        <li> Καλώς ήρθες </li>
        <li> Γεια σας, πως θα μπορούσα να σας εξυπηρετήσω; </li>
        <li> Καλησπέρα, πως μπορώ να βοηθήσω; </li>
      </random>
    </template>
  </category>
  <category>
    <pattern>Καλησπέρα</pattern>
    <template>
      <srail>Γεια</srail>
    </template>
  </category>
  <category>
    <pattern>Καλημέρα</pattern>
    <template>
      <srail>Γεια</srail>
    </template>
  </category>
  <category>
    <pattern>Γεια σου</pattern>
    <template>
      <srail>Γεια</srail>
    </template>
  </category>
  <category>
    <pattern>μαθήματα</pattern>
  </category>

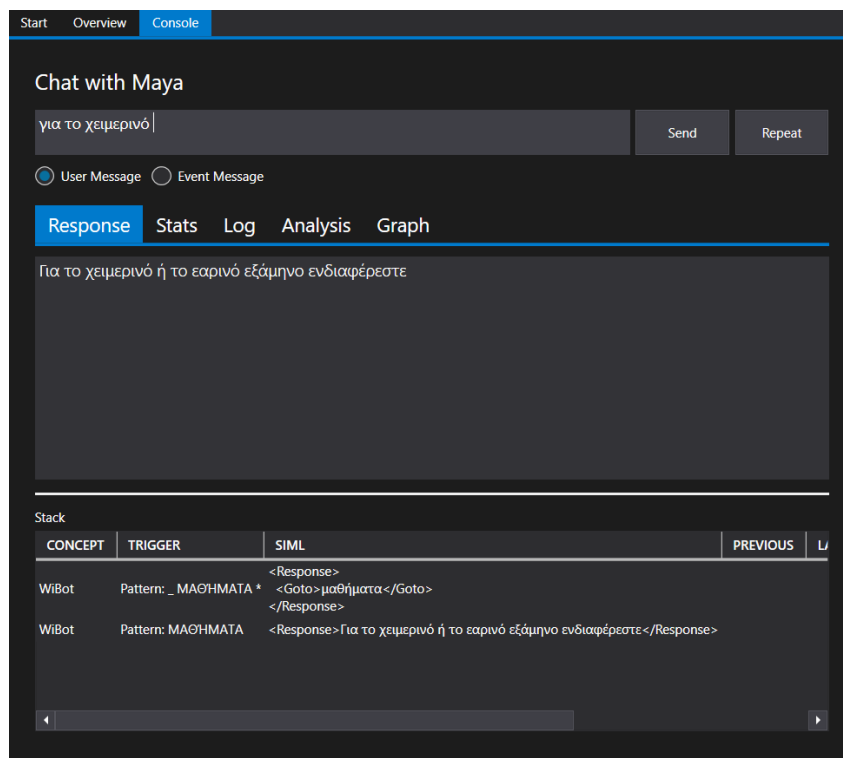
```

Εικόνα 3.1.6 Παράδειγμα AIML

Στο Syn Bot Studio, έγινε η δεύτερη υλοποίηση με την χρήση της SIML. Η απάντηση του chatbot βρίσκεται στο section “Response”. Τα ακόλουθα στιγμιότυπα αφορούν το user story 1

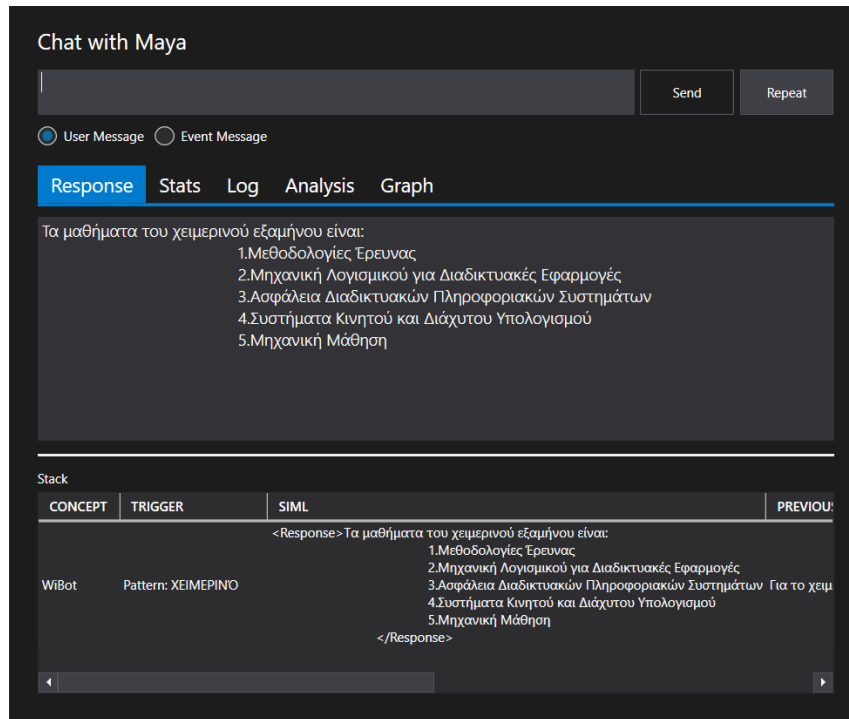


Εικόνα 3.1.7 Ερώτηση χρήστη - Syn Bot Studio



Εικόνα 3.1.8 Επιλογή εξαμήνου - Syn Bot Studio

Στο κάτω μέρος της οθόνης, μπορούμε να παρατηρήσουμε το section, Stack. Η στήλη *TRIGGER* περιέχει το keyword που βρέθηκε σε κάποιο pattern της SIML και η στήλη SIML περιέχει το Response, δηλαδή την απάντηση που έχει οριστεί για αυτό.



Εικόνα 3.1.9 Απάντηση chatbot - Syn Bot Studio

Στον κώδικα SIML, έχουμε ορίσει ένα Concept με πολλά μοντέλα- Model για διαφορετικούς σκοπούς. Κάθε μοντέλο περιέχει δύο σημαντικές οντότητες το Pattern και το Response, στα οποία πρέπει να έχουμε ορίσει τουλάχιστον ένα Item για να είναι ικανοποιητικό το μοντέλο μας. Το Pattern είναι ένα σύνολο από λέξεις που ελέγχονται στο runtime, σε σύγκριση με το input που δίνει ο χρήστης. Στην περίπτωση που ταιριάζει ή αλλιώς ικανοποιείται η ορισμένη συνθήκη, επιστρέφεται το αντίστοιχο Response στον χρήστη [3].

```

<Siml xmlns:Think="http://syn.co.in/2014/siml#think">
  <Concept Name="WiBot">

    <Model>
      <Pattern>
        <Item>ΚΑΛΗΣΠΕΡΑ</Item>
        <Item>ΓΕΙΑ</Item>
        <Item>ΓΕΙΑ ΣΟΥ</Item>
      </Pattern>
      <Response>
        <Phrase>
          <Item>Καλησπέρα, πως θα μπορούσα να σας εξυπηρετήσω;</Item>
          <Item>Καλησπέρα, πως μπορώ να βοηθήσω;</Item>
        </Phrase>
      </Response>
    </Model>

    <Model>
      <Pattern>ΜΑΘΗΜΑΤΑ</Pattern>
      <Response>Για το χειμερινό ή το εαρινό εξάμηνο ενδιαφέρεστε</Response>
    </Model>

    <Model>
      <Pattern>_ ΜΑΘΗΜΑΤΑ</Pattern>
      <Response>
        <Goto>μαθήματα</Goto>
      </Response>
    </Model>

    <Model>
      <Pattern>_ ΜΑΘΗΜΑΤΑ</Pattern>
      <Response>
        <Goto>μαθήματα</Goto>
      </Response>
    </Model>
  </Concept>
</Siml>

```

Εικόνα 3.1.10 Παράδειγμα SIML

Όσο αφορά το Rasa framework η συζήτηση έγινε μέσω του terminal. Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα με τον *Συνδυαστικό* διάλογο καθώς και δείγμα κώδικα από τα αρχεία εκπαίδευσης.

```

Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> γεια
Είμαι ο EduBot, ο αυτόματος ψηφιακός βοηθός σου!Είμαι εδώ για να σου δώσω πληροφορίες
που αφορούν την γραμματεία του τμήματος αλλά και άλλες πληροφορίες.
Πως μπορώ να σε εξυπηρετήσω;
Your input -> Πως θα διεξαχθεί η εξέταση για το Μάθημα τον Σεπτέμβριο
Παρακαλώ πείτε μου τον αριθμό του μητρώου σας
Your input -> 72020
Η εξέταση θα διεξαχθεί εξ αποστάσεως, μέσω της πλατφόρμας zoom.
Μπορώ να σε εξυπηρετήσω σε κάτι άλλο;
Your input -> Πως μπορώ να επικοινωνήσω με τον Καθηγητή
Το τηλέφωνο επικοινωνίας είναι: 2234034498 και το mail: example@ihu.gr
Μπορώ να σε εξυπηρετήσω σε κάτι άλλο;
Your input -> 

```

Εικόνα 3.1.11 Συνδυαστικό -Rasa

```



version: "2.0"
stories:
- story: greet
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_intro1
  - action: utter_help
- story: courses_story
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_intro1
  - action: utter_help
  - intent: courses
  - action: utter_courses
  - action: utter_help_new
- story: contactInfo_story
  steps:
  - intent: greet
  - action: utter_intro1
  - action: utter_help
  - intent: contactInfo
  - action: utter_ask_edu_code
  ## intent:courses
  - Θα ήθελα να μάθω ποια είναι τα μαθήματα ανά εξάμηνο
  - Ποια είναι τα μαθήματα ανά εξάμηνο
  - Ποια είναι τα μαθήματα του εξαμήνου
  - Μπορείς να μου πεις ποια είναι τα μαθήματα ανά εξάμηνο
  ## intent:contactInfo
  - Θα ήθελα να μάθω τους τρόπους επικοινωνίας με τον καθηγητή Καθηγητή
  - Θα ήθελα να μάθω τους τρόπους επικοινωνίας με τον κ Καθηγητή
  - Θα ήθελα να μάθω τους τρόπους επικοινωνίας με τον Καθηγητή
  - Ποιοι είναι οι τρόποι επικοινωνίας με τον Καθηγητή
  - Πως μπορώ να επικοινωνήσω με τον Καθηγητή
  ## intent:septemberExams
  - Θα ήθελα να μάθω πως θα διεξαχθεί η εξέταση για το Μάθημα τον Σεπτέμβριο
  - Πως θα διεξαχθεί η εξέταση για το Μάθημα τον Σεπτέμβριο
  - Εξέταση για το Μάθημα τον Σεπτέμβριο
  ## intent:thanks
  - Ευχαριστώ
  - ευχαριστώ
  - Τέλεια
  - Υπέροχα


```

Εικόνα 3.1.12 Παράδειγμα Rasa

Αφότου, υλοποιήθηκαν όλα τα παραπάνω user stories σε όλες τις πλατφόρμες, συγκεντρώθηκαν ορισμένες πληροφορίες. Ακολουθεί ένας συγκεντρωτικός πίνακας με σχόλια πάνω σε πλεονεκτήματα και μειονεκτήματα που παρατηρήθηκαν κατά την υλοποίηση του pilot project.

Πίνακας 3.1.1 Εργαλεία και πλεονεκτήματα-μειονεκτήματα

Εργαλείο	Πλεονεκτήματα	Μειονεκτήματα
	<p>Γρήγορο γιατί είναι Online, χωρίς κάποιο ιδιαίτερο setup μόνο με εγγραφή.</p> <p>Εύκολη και κατανοητή γλώσσα και δομή (AIML).</p> <p>Πολλές πληροφορίες διαθέσιμες online.</p>	<p>Η δωρεάν έκδοση έχει περιορισμένες δυνατότητες καθώς επίσης υποστηρίζει συγκεκριμένο αριθμό/πλήθος μηνυμάτων.</p>
	<p>Ξεκάθαρη λογική, μιας και κάθε αρχείο περιέχει συγκεκριμένα στοιχεία.</p> <p>Κατανοητή δομή.</p> <p>Διαθέτει automated test.</p>	<p>Αρκετά προβλήματα κατά την εγκατάσταση.</p> <p>Δυσκολία στο να επιλυθούν ορισμένα ζητήματα (εκδόσεις κ.α.)</p>

 <p>Syn Bot Studio</p>	<p>Εύκολη γλώσσα υλοποίησης (SIML). Εύκολο στήσιμο του project και γρήγορη ανάπτυξη.</p>	<p>Τα παραδείγματα που υπάρχουν είναι λίγα, χωρίς αναρτημένα error ή άλλα ζητήματα (πχ stackoverflow), πράγμα που το καθιστά δύσκολο σε ορισμένες περιπτώσεις.</p>
---	--	--

Επιπλέον, για την καλύτερη ανάλυση της απόδοσης των εργαλείων, αλλά και για την τελική επιλογή του καταλληλότερου, επιλέχθηκαν τα σπουδαιότερα κριτήρια αξιολόγησης:

1. Ease of Use
2. Multi platform deployment
3. Research interest
4. License scheme
5. Platform attractiveness
 - Tutorials / documentation
 - Number of available development tools
 - Ease of debugging
 - Platform robustness
 - Deployment ease
 - Security concerns

Ειδικότερα, το *Ease of Use* (end user), αποτελεί κριτήριο αξιολόγησης σχετικά με το πόσο εύκολη και φιλική είναι προς τον τελικό χρήστη η εφαρμογή μετά την ανάπτυξή της. Το *Multi platform deployment* αφορά την δυνατότητα ανάπτυξης σε διάφορα λειτουργικά συστήματα, όπως για παράδειγμα σε Windows ή Linux. Το *Research interest* (development), αποτελεί μια μετρική για το πόσο κερδίζει το ενδιαφέρον από πλευράς διερεύνησης του προγραμματιστή. Ένα σημαντικό κριτήριο αφορά την άδεια χρήσης, δηλαδή το *License scheme*. Για παράδειγμα, εάν προσφέρουν την πλήρη έκδοση δωρεάν ή διαθέτουν κάποιες επιλογές επί πληρωμή. Το *Platform attractiveness* από την άλλη, περιλαμβάνει έξι υποκριτήρια και συγκεντρώνει ένα σύνολο παραμέτρων, τα οποία κάνουν την πλατφόρμα “ελκυστική”. Συγκεκριμένα, το πλήθος των διαθέσιμων tutorials και documentations, το πλήθος των διαθέσιμων εργαλείων ανάπτυξης, η ευκολία ή δυσκολία που παρουσιάζεται στα πλαίσια του debugging, την ικανότητα να αντιμετωπίζει σφάλματα κατά την εκτέλεση και λανθασμένες πληροφορίες, τον βαθμό ευκολίας στο στάδιο του deployment και την πιθανότητα να παρουσιαστούν θέματα ασφάλειας.

Αφότου συγκεντρώθηκαν τα κριτήρια, καθορίστηκε ένας συντελεστής βαρύτητας που αντιστοιχεί σε κάθε ένα από αυτά. Κατατάσσει δηλαδή κάθε κριτήριο με βάση την σημασία

του, για παράδειγμα αν το κριτήριο A είναι ποιο είναι πιο σημαντικό από το κριτήριο B. Στην συνέχεια, δημιουργήθηκε κατάλληλος πίνακας, σε πρόγραμμα λογιστικών φύλλων (τύπου excel), για την ορθότερη συμπλήρωσή του.

Στην ακόλουθη εικόνα (Εικόνα 3.1.13), παρατηρούμε ένα στιγμιότυπο του πίνακα στην τελική του μορφή. Συγκεκριμένα, στη πρώτη στήλη καταγράφονται τα κριτήρια που προαναφέρθηκαν, στην δεύτερη ορίζεται ο βαθμός βαρύτητας που αναλογεί σε καθένα από αυτά. Έχει υπολογιστεί επί της εκατό, αθροίζοντας δηλαδή όλα στην μονάδα. Στις τελευταίες τρεις στήλες παρατηρούνται τα τρία εργαλεία και η βαθμολογία τους μετά από την συνολική μελέτη που έγινε. Τέλος, στην τελευταία γραμμή παρουσιάζονται τα αθροίσματα ανά στήλη (Total). Εδώ, παρατηρείται ότι το Pandorabots έχει 7,65 συνολική βαθμολογία, το Rasa 8,38 και το Syn Bot Studio 7,77. Καταλήγοντας, το Rasa θεωρείται η καλύτερη επιλογή για την ανάπτυξη της συγκεκριμένης εργασίας, καθώς διαθέτει την υψηλότερη βαθμολογία.

Κριτήρια	Βαρύτητα	Βαρύτητα (%)	Pandorabots	Rasa	Syn Bot Studio
1. Ease of Use (end user)	80,00	0,24	8,00	7,00	8,00
2. Multi platform deployment	75,00	0,23	9,00	9,00	9,00
3. Research interest (Development)	65,00	0,20	7,00	9,00	7,00
4. License scheme (Δωρεάν / Επί πληρωμή)	55,00	0,17	6,00	9,00	7,00
5. Platform attractiveness - Subtotal	55,00	0,17	7,73	8,18	7,45
5.1 Tutorials / documentation	10,00	0,18	8,00	9,00	7,00
5.2 Number of available development tools	5,00	0,09	7,00	9,00	8,00
5.3 Ease of debugging	10,00	0,18	9,00	8,00	7,00
5.4 Platform robustness	10,00	0,18	8,00	8,00	8,00
5.5 Deployment ease	10,00	0,18	7,00	6,50	7,00
5.6 Security concerns	10,00	0,18	7,00	9,00	8,00
Total	330,00	1,00	7,65	8,38	7,77

Εικόνα 3.1.13 Πίνακας Αξιολόγησης - excel

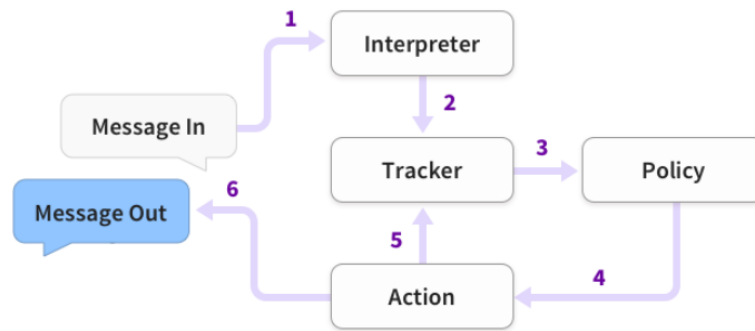
3.2 Rasa Framework

Το Rasa, όπως προαναφέρθηκε αποτελεί ένα open source framework μηχανικής μάθησης, δημιουργώντας αυτοματοποιημένες συνομιλίες κειμένου και φωνής. Κατανοεί μηνύματα, κρατάει συνομιλίες και υπάρχει δυνατότητα σύνδεσης με κανάλια μηνυμάτων και APIs.



Εικόνα 3.2.1 Rasa Framework

Όσο αφορά την αρχιτεκτονική του Rasa [16] και συγκεκριμένα την διαχείριση μηνυμάτων, υπάρχει μια σειρά βημάτων τα οποία περιγράφονται παρακάτω. Στην Εικόνα 3.2.2, αποτυπώνονται τα βασικά βήματα για το πώς ένα chatbot (που χτίζεται με το Rasa) ανταποκρίνεται σε ένα μήνυμα:



Εικόνα 3.2.2 Αρχιτεκτονική Rasa [16]

1. Το μήνυμα λαμβάνεται και μεταβιβάζεται στον Interpreter/διερμηνέα, ο οποίος το μετατρέπει σε λεξικό που περιλαμβάνει το αρχικό κείμενο, την πρόθεση και τυχόν οντότητες που βρέθηκαν. Αυτό το κομμάτι το χειρίζεται το NLU.
2. Το μήνυμα περνά από τον Interpreter/διερμηνέα στον Tracker. Ο Tracker είναι το αντικείμενο (object) που παρακολουθεί την κατάσταση της συνομιλίας.
3. Η τρέχουσα κατάσταση του Tracker/ιχνηλάτη αποστέλλεται σε κάθε πολιτική.
4. Κάθε Policy/πολιτική επιλέγει ποια ενέργεια θα ακολουθήσει.
5. Η επιλεγμένη Action/ενέργεια καταγράφεται από τον Tracker/ιχνηλάτη.
6. Η τελική απάντηση αποστέλλεται στον χρήστη.

3.2.1 Installation

Σχετικά με την εγκατάσταση, εκτελέστηκαν διάφορα παραδείγματα προκειμένου να ολοκληρωθεί σωστά. Μετά από πειραματισμούς στις εκδόσεις (downgrade, upgrade) της python, του Rasa κλπ, καταλήξαμε σε λύση, στην οποία χρειάστηκε η λήψη του Anaconda. Το Anaconda βοηθάει ιδιαίτερα στην διαχείριση των πακέτων και στην διατήρηση των διαφορετικών dependencies. Οι παρακάτω εντολές λειτουργούν σε συστήματα Windows και Linux, για την δημιουργία και ενεργοποίηση του environment που θα τρέξει το Rasa:

- `conda create -n rasa python=<version>`
- `conda activate rasa`

Ακολουθεί η εγκατάσταση του UltraJSON και του TensorFlow. Το UltraJSON είναι ένας αρκετά γρήγορος JSON κωδικοποιητής και αποκωδικοποιητής, γραμμένος σε C με συνδέσεις για Python 2.7 και 3.5+. Από την άλλη, το TensorFlow είναι μια βιβλιοθήκη ανοιχτού κώδικα για μηχανική μάθηση. Διαθέτει ένα ολοκληρωμένο, ευέλικτο σύστημα εργαλείων και βιβλιοθηκών. Με την χρήση της οι προγραμματιστές δημιουργούν και αναπτύσσουν εύκολα εφαρμογές που υποστηρίζονται από MM [17]. Η version που χρησιμοποιήθηκε είναι η 2.3.3.

- `conda install ujson`
- `conda install tensorflow`

Όπως προαναφέρθηκε, χρειάζεται η εγκατάσταση της Python, εφόσον δεν υπάρχει, και με την κατάλληλη εντολή γίνεται η λήψη του Rasa και των αντίστοιχων dependencies

➤ `pip install rasa [full]`

Μόλις, ολοκληρωθεί η εγκατάσταση του Rasa, συνέχεια έχει η δημιουργία του project. Η ακόλουθη εντολή αφότου εκτελεστεί, δημιουργεί ένα sample project, με τα απαραίτητα αρχεία που χρειάζεται για να ξεκινήσει κάποιος.

➤ `rasa init`

Τα αρχεία που απαιτούνται περιγράφονται στην συνέχεια. Σε αυτό το σημείο όμως, καλό θα ήταν να σημειωθούν τέσσερις βασικές έννοιες, απαραίτητες για την κατανόηση του επόμενου κεφαλαίου:

Πίνακας 3.2.1 Βασικές έννοιες Rasa

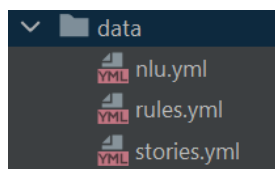
Πρόθεση / Intent	Αφορά την πρόθεση του χρήστη, για το ερώτημα που θα υποβάλει στο chatbot.
Οντότητα / Entity	Τα σημαντικά κομμάτια ή λέξεις που εντοπίζονται στο ερώτημα που κάνει ο χρήστης.
Ιστορία / Story	Η ροή ή αλλιώς εξέλιξη της συζήτησης.
Δράση / Action	Η ενέργεια που πρέπει να εκτελέσει το chatbot μετά από συγκεκριμένο ερώτημα.

3.2.2 Δομή και Βασικά Αρχεία

Εδώ παρουσιάζονται τα βασικά αρχεία, αλλά και η δομή/structure των αρχείων και φακέλων.

Οι πληροφορίες παρέχονται από τον επίσημο ιστοτόπο του Rasa [4].

Στον κατάλογο data, υπάρχουν ορισμένα από τα κύρια αρχεία. Όπως φαίνεται και στην παρακάτω εικόνα, περιέχει τα αρχεία nlu, rules και stories. Πιο συγκεκριμένα:



Εικόνα 3.2.3 Φάκελος data

NLU.YML

Αποτελεί το ακρωνύμιο του Natural language Understanding. Το αρχείο nlu περιέχει δεδομένα εκπαίδευσης, ορισμένα κάτω από συγκεκριμένες κατηγορίες όπως είναι τα intents τα entities, τα συνώνυμα. Στην εικόνα που ακολουθεί, παρουσιάζεται το intent *greet* για τον

χαιρετισμό, με τις αντίστοιχες εκφράσεις. Είναι δηλαδή ο χαιρετισμός που περιμένει το chatbot, από τον χρήστη.

```
## intent:greet
- Γειά σας
- Γεια
- γεια
- Γειά σου
- Γειά σας
- Γειά χαρά
- Χαίρετε
- Χαιρεται
- χαίρεται
- χαιρεται
- Καλησπέρα
```

Εικόνα 3.2.4 Αρχείο nlu.yml

RULES.YML

Οι κανόνες / rules αποτελούν τους περιορισμούς και συνήθως χειρίζονται συγκεκριμένα και μικρά μοτίβα συνομιλίας. Για παράδειγμα, ένας κανόνας μπορεί να οριστεί πάνω σε μια συγκεκριμένη πρόθεση (intent), όπως είναι η αποστολή email.

```
rules:
- rule: Activate Form
  steps:
  - intent: send_email
  - action: email_form
  - slot_was_set:
    - subject: "Θέμα"
  - slot_was_set:
    - email: "abc@something.com"
  - slot_was_set:
    - message: "Lorem ipsum dolor sit amet"
  - active_loop: email_form
```

Εικόνα 3.2.5 Αρχείο rules.yml

STORIES.YML

Εδώ ορίζονται οι ιστορίες, δηλαδή η ροή μιας συζήτησης μεταξύ του χρήστη και του chatbot. Το μήνυμα του χρήστη αποτυπώνεται ως πρόθεση (intent), ενώ η απάντηση του διαλογικού πράκτορα, ως ενέργεια (action). Στην παρακάτω εικόνα, παρουσιάζεται η ιστορία που αφορά τα μαθήματα ανά εξάμηνο. Αποτελεί ένα πλήρες παράδειγμα ιστορίας, ξεκινώντας με τον χαιρετισμό, έπειτα το chatbot συστήνεται και ρωτάει πως μπορεί να βοηθήσει τον χρήστη. Ακολουθεί η ερώτηση σχετικά με τα μαθήματα του εξαμήνου και τελικά το chatbot δίνει την αντίστοιχη απάντηση.

```
stories:
- story: courses_story
  steps:
  - intent: greet
  - action: utter_intro1
  - action: utter_help
  - intent: courses
  - action: utter_ask_courses
  - intent: semester
  - action: action_courses
```

Εικόνα 3.2.6 Αρχείο stories.yml

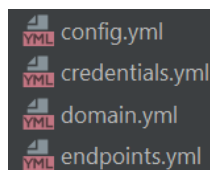
```
- story: elearning_platform
  steps:
  - intent: elearning_platform
  - action: utter_elearning_platform
  - action: utter_help_new

- story: academicID_link
  steps:
  - intent: academicID_link
  - action: utter_academicID_link
  - action: utter_help_new

- story: available_services
  steps:
  - intent: available_services
  - action: utter_available_services
  - action: utter_help_new
```

Εικόνα 3.2.7 Αρχείο stories.yml (2)

Άλλα σημαντικά αρχεία είναι το config, credential, domain, endpoint, όλα σε επέκταση αρχείου .yml, τα οποία παρουσιάζονται αναλυτικά στην συνέχεια.



Εικόνα 3.2.8 Άλλα αρχεία

CONFIG.YML

Ορίζονται τα pipelines επεξεργασίας κειμένου για την διερμηνεία μηνυμάτων του χρήστη, την πρόβλεψη πρόθεσης, και την επιλογή της σωστής έκφρασης/απάντησης. Όλα αυτά ορίζονται με τις πολιτικές (κατηγορία policies). Ακόμη, σε αυτό το σημείο ορίζεται η

γλώσσα, με το στοιχείο *language* και τον κωδικό της γλώσσας με τα δυο γράμματα, όπως *el* ή *en* (ελληνικά ή αγγλικά αντίστοιχα). Αναλυτικότερα οι πολιτικές που χρησιμοποιήθηκαν [16] policies:

- name: MemoizationPolicy

- name: TEDPolicy

max_history: 5

epochs: 10

Transformer Embedding Dialogue (TED) Policy: είναι μια αρχιτεκτονική πολλαπλών εργασιών, υπεύθυνη για την πρόβλεψη της επόμενης ενέργειας και την αναγνώριση οντοτήτων (entity recognition). Η αρχιτεκτονική αποτελείται από πολλούς κωδικοποιητές μετασχηματιστών που είναι κοινοί και για τις δύο παραπάνω εργασίες.

MemoizationPolicy: θυμάται τις ιστορίες από τα δεδομένα εκπαίδευσης. Ελέγχει εάν η τρέχουσα συνομιλία ταιριάζει με τις ιστορίες που βρίσκονται στο αρχείο *stories.yml*. Αν βρει αντίστοιχη συνομιλία, θα προβλέψει την επόμενη ενέργεια από τις ιστορίες των δεδομένων προπόνησης με confidence 1,0. Εάν δεν βρει αντίστοιχη συνομιλία, η πολιτική προβλέπει None, με confidence 0,0.

epochs: ορίζει πόσες φορές ο αλγόριθμος θα δει τα δεδομένα εκπαίδευσης. Μερικές φορές το μοντέλο χρειάζεται περισσότερες επαναλήψεις (εποχές) για να μάθει και να εκπαιδευτεί σωστά.

max_history: Ελέγχει πόσο από το ιστορικό διαλόγου εξετάζει το μοντέλο, για να αποφασίσει ποια ενέργεια θα ακολουθήσει. Το προεπιλεγμένο max_history είναι None, οπότε λαμβάνεται υπόψη το πλήρες ιστορικό διαλόγου από την επανεκκίνηση του session.

```
policies:  
- name: MemoizationPolicy  
- name: TEDPolicy  
  max_history: 5  
  epochs: 100  
  constrain_similarities: true  
  model_confidence: softmax  
- name: RulePolicy  
- name: FallbackPolicy  
  nlu_threshold: 0.4  
  core_threshold: 0.3  
  fallback_action_name: "action_default_fallback"
```

Εικόνα 3.2.9 Αρχείο config (Policies)

Τα pipelines που τελικά χρησιμοποιήθηκαν:

WhitespaceTokenizer: χωρίζει τις προτάσεις σε tokens, με βάση τα κενά διαστήματα

RegexFeaturizer: βοηθάει στο intent classification και entity extraction, με την δημιουργία

χαρακτηριστικών κοινών εκφράσεων

LexicalSyntacticFeaturizer: δημιουργεί λεκτικά και συντακτικά χαρακτηριστικά, και βοήθεια στην διαδικασία του entity extraction

CountVectorsFeaturizer: δημιουργεί μια αναπαράσταση *bag-of-words* των μηνυμάτων του χρήστη, intents και responses

DIETClassifier: είναι μετασχηματιστής διπλής πρόθεσης και οντότητας (Dual Intent and Entity Transformer). Η αρχιτεκτονική βασίζεται σε έναν μετασχηματιστή που χρησιμοποιείται και για τις δύο εργασίες.

EntitySynonymMapper: αντιστοιχεί τιμές συνώνυμων οντοτήτων

```
language: el
pipeline:
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: "char_wb"
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 100
    constrain_similarities: true
    model_confidence: softmax
  - name: EntitySynonymMapper
```

Εικόνα 3.2.10 Αρχείο config (Pipeline)

CREDENTIALS.YML

Το αρχείο αυτό περιέχει τα στοιχεία σύνδεσης με άλλες πλατφόρμες ή κανάλια που χρησιμοποιεί το chatbot, όπως για παράδειγμα facebook, slack και στην συγκεκριμένη περίπτωση με το socketIO.

```
socketio:
  user_message_evt: user_uttered
  bot_message_evt: bot_uttered
  session_persistence: false

rasa:
  url: "http://localhost:5002/api"
```

Εικόνα 3.2.11 Αρχείο credentials.yml

DOMAIN.YML

Αποτελεί τον “κόσμο” του chatbot. Περιέχει όλα όσα πρέπει να γνωρίζει το chatbot, όπως οι ενέργειες που μπορεί να κάνει, οι προθέσεις που πρέπει να κατανοήσει, οι απαντήσεις του για κάθε πρόθεση, τα slots για την αποθήκευση τιμών και άλλα.

```
responses:
  utter_greet:
    - text: "Γεια σου :)"
    - text: "Χαίρετε :)"
    - text: "Καλώς ήρθες!"
    - text: "Καλή σου ημέρα"
```

Εικόνα 3.2.12 Αρχείο domain.yml

```
slots:
  semester:
    type: text
    influence_conversation: true
  category_id:
    type: text
    influence_conversation: true
  professor_name:
    type: text
    influence_conversation: true
  subject:
    type: text
    influence_conversation: true
  email:
    type: text
    influence_conversation: true
  message:
    type: text
    influence_conversation: true
```

Εικόνα 3.2.13 slots

ENDPOINTS.YML

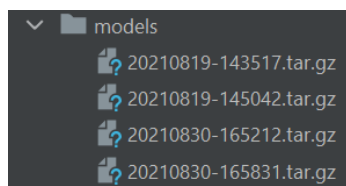
Περιέχει λεπτομέρειες για σύνδεση του Rasa με endpoints που μπορεί να συνδεθεί.

```
action_endpoint:
  url: http://localhost:5055/webhook
```

Εικόνα 3.2.14 Αρχείο endpoints.yml

MONTEΛΟ .TAR.GZ

Επιπλέον, όταν εκπαιδεύεται το chatbot (με την εντολή `rasa train`), παράγεται ένα αρχείο / μοντέλο και αποθηκεύεται στον φάκελο `models`. Ο τύπος του αρχείου είναι `.tar.gz` και το όνομα του αρχείου δημιουργείται με βάση την ημερομηνία που δημιουργήθηκε, σε μορφή `YYYYMMDD-hhmmss`. Στην περίπτωση όπου έχουμε μοντέλα NLU, προστίθεται το πρόθεμα `nlu`, πριν από την ημερομηνία.



Εικόνα 3.2.15 Φάκελος models

ACTIONS.PY

Ένα άλλο κεφάλαιο του Rasa είναι τα custom actions, υλοποιούνται με τη συγγραφή Python, εκτελώντας οποιοδήποτε custom κώδικα έχει γράψει ο χρήστης. Μπορεί να χρησιμοποιηθούν για ελέγχους ή πρόσθετες διαδικασίες, για κλήσεις API, για ερωτήματα στην βάση δεδομένων και άλλα. Το αρχείο στο οποίο υλοποιούνται τα παραπάνω είναι το “actions.py”.

```
class ActionResetEmailSlots(Action):...  
  
class ActionSubmit(Action):...  
  
class ActionSubmitWithFile(Action):...  
  
class ActionReadFile(Action):...  
  
class ActionProfessorInfo(Action):...  
  
class ActionEduPeople(Action):...  
  
class ActionAnnouncementsApi(Action):...
```

Εικόνα 3.2.16 Αρχείο actions.py

3.2.3 Άλλες χρήσιμες εντολές

Για να εκτελεστούν τα custom actions, θα πρέπει να γίνει εκκίνηση του server σε νέο τερματικό (terminal), πληκτρολογώντας την εντολή:

- `python -m rasa_sdk --actions actions`

Ακολουθούν άλλες χρήσιμες εντολές, που χρειάστηκε να εκτελέσουμε.

Για να μπορέσουμε να υλοποιήσουμε API κλήσεις, όπως για παράδειγμα κλήση σε API για την συλλογή των ανακοινώσεων του Τμήματος, απαιτείται η εγκατάσταση του πακέτου requests, στο τρέχον περιβάλλον του Rasa, με την ακόλουθη εντολή:

- `pip install requests`

Πριν από την εκπαίδευση του bot, προτείνεται ως καλή πρακτική, να γίνεται έλεγχος για τυχόν λάθη σε αρχεία όπως τα stories, rules.

- `rasa data validate`

Τέλος, για την εκπαίδευση του bot τρέχουμε

- `rasa train`

4

Υλοποίηση

Το κεφάλαιο αυτό, παρουσιάζει την ανάλυση απαιτήσεων, την μεθοδολογία σχεδίασης, βασικά διαγράμματα της εφαρμογής, και την τελική ανάπτυξη – υλοποίηση.

4.1 Ανάλυση Απαιτήσεων

Για την μοντελοποίηση των απαιτήσεων των χρηστών, αρχικά ορίστηκαν σενάρια με βάση έναν κύριο ρόλο - χρήστη, τον φοιτητή. Αναπτύχθηκαν περίπου τριάντα ιστορίες - σενάρια χρηστών (user stories) που αναφέρονται σε αυτόν τον ρόλο και περιγράφουν τις επιθυμητές λειτουργίες. Ακριβώς επειδή περιγράφουν τις επιθυμητές λειτουργικότητες, ιδανικά θα πρέπει να καταγράφονται από τον χρήστη. Στην συγκεκριμένη περίπτωση όμως, δεν ήταν δυνατή η συλλογή των στοιχείων με αυτόν τον τρόπο.

Για τις ιστορίες χρησιμοποιήθηκαν τα υπολογιστικά φύλλα, excel. Φυσικά, υπάρχουν διαθέσιμα εργαλεία, τα οποία διαθέτουν και άλλες λειτουργίες, όπως είναι το αρκετά διαδεδομένο Jira, το storiesonboard κα.

Για την εκτενέστερη μελέτη των σεναρίων έχει δημιουργηθεί στο Παράρτημα Α, η αποτύπωση τους σε αναλυτική μορφή. Έπειτα, και αφού ήταν ολοκληρωμένη η εγκατάσταση του Rasa, προχωρήσαμε στην υλοποίηση των σεναρίων ένα προς ένα, συλλέγοντας τις αντίστοιχες πληροφορίες.

Γενικότερα, το chatbot μπορεί να δώσει πληροφορίες για τα ακόλουθα:

- Ανακοινώσεις του Τμήματος, είτε επιλέγοντας συγκεκριμένη κατηγορία (π.χ. Γραμματεία) είτε τις τελευταίες από όλες τις κατηγορίες
- Στοιχεία επικοινωνίας καθηγητών, είτε λίστα με όλους τους καταχωρημένους, είτε για κάποιον συγκεκριμένο πληκτρολογώντας το ονοματεπώνυμό του
- Στοιχεία επικοινωνίας με την γραμματεία (διεύθυνση, email, τηλέφωνο)
- Τρόποι επικοινωνίας με το Τμήμα Φοιτητικής Μέριμνας
- Τρόποι πρόσβασης στην σχολή, επιστρέφοντας την διεύθυνση καθώς επίσης και τις γραμμές αστικής συγκοινωνίας που προσεγγίζουν την σχολή
- Πληροφορίες για την πλατφόρμα e-learning του Τμήματος
- Διαθέσιμες πλατφόρμες- ηλεκτρονικές υπηρεσίες που χρειάζεται να γνωρίζει ο φοιτητής
- Πληροφορίες για την έκδοση της φοιτητικής ταυτότητας
- Πληροφορίες για την διαδικασία ορκωμοσίας (έντυπα)
- Το Ωρολόγιο πρόγραμμα (εμφάνιση και αποστολή με email)
- Το Ακαδημαϊκό ημερολόγιο (εμφάνιση και αποστολή με email)
- Γενικές πληροφορίες για την αίτηση εισαγωγής στο πρόγραμμα, για την υποβολή αιτήσεων, για το Erasmus, αλλά και για το Τμήμα και την σχολή
- Πληροφορίες για τις υποτροφίες
- Δυνατότητα αποστολής email
- Ώρες γραφείου καθηγητών
- Πληροφορίες για το πρόγραμμα μερικής φοίτησης
- Έναρξη εξαμήνου (ημερομηνίες)
- Μαθήματα εξαμήνου

Ακόμη, για την υλοποίηση του project χρησιμοποιήθηκε ως IDE το PyCharm (Community Edition 2021.2). Το PyCharm [18] είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης σε γλώσσα Python, που παρέχει ένα ευρύ φάσμα βασικών εργαλείων. Διατίθεται σε τρεις εκδόσεις, την Community (δωρεάν και ανοιχτού κώδικα), την Professional (επί πληρωμή) και την Edu (δωρεάν και ανοιχτού κώδικα). Υποστηρίζει Python 2 (version 2.7) και Python 3 (version 3.6 ως 3.10).

Επιπλέον, οι υπόλοιπες εκδόσεις παρουσιάζονται στο παρακάτω στιγμιότυπο:

```
Rasa Version      : 2.8.2
Minimum Compatible Version: 2.8.0
Rasa SDK Version  : 2.8.1
Python Version    : 3.8.11
```

Εικόνα 4.1.1 Εκδόσεις

4.2 Μεθοδολογία σχεδίασης

Τα κύρια βήματα που έγιναν για την ανάπτυξη της εφαρμογής ήταν τα παρακάτω:

1. Αναζήτηση θεωρητικού υπόβαθρου
2. Μελέτη διαθέσιμων εργαλείων
3. Ανάπτυξη Pilot Project (σε 3 πλατφόρμες/εργαλεία)
4. Επιλογή πλατφόρμας ανάπτυξης
5. Υλοποίηση WiBot
6. Αξιολόγηση Χρηστών (Α' Φάση)
7. Διορθώσεις και Προσθήκες
8. Αξιολόγηση Χρηστών (Β' Φάση)

Η παρούσα διπλωματική ξεκίνησε με την ιδέα της ανάπτυξης ενός chatbot. Ήταν μια πρόκληση μιας και δεν υπήρχε καθόλου τεχνολογικό υπόβαθρο στο συγκεκριμένο πεδίο, αλλά ταυτόχρονα παρουσίαζε μεγάλο ενδιαφέρον.

Στην συνέχεια, σκεφτήκαμε την ανάπτυξη του chatbot με σκοπό την εξυπηρέτηση των φοιτητών του μεταπτυχιακού προγράμματος. Γνωρίζοντας ότι υπάρχει αυτή η ανάγκη και επιπλέον θεωρώντας ότι αποτελεί ένα έλλειμμα των Ελληνικών Πανεπιστημίων, οριστικοποιήσαμε το θέμα και περάσαμε στην αναζήτηση του θεωρητικού υπόβαθρου. Μελετήσαμε σε γενικότερο επίπεδο τον τομέα αυτόν. Αρχικά, αναζητήσαμε πως ορίζονται τα chatbots, που μπορούν να εφαρμοστούν, ποια είναι τα πλεονεκτήματά τους και άλλα. Διερευνήσαμε μελέτες που έχουν γίνει, ώστε να δούμε τα πεδία εφαρμογής τους και επιπλέον να αποκτήσουμε μια εικόνα της μελλοντικής μας βιβλιογραφίας.

Πριν ξεκινήσουμε την μελέτη των διαθέσιμων εργαλείων ανάπτυξης, σκεφτήκαμε τις προϋποθέσεις για την επιλογή αυτή, θέτονταν ως κύριο χαρακτηριστικό την υποστήριξη της ελληνικής γλώσσας. Συγκεντρώσαμε πληροφορίες από ποικίλα εργαλεία/πλατφόρμες και καταλήξαμε σε τρία επικρατέστερα. Το Pandorabots, το Syn Bot Studio και το Rasa Framework. Στόχος ήταν η μελέτη και η ανάπτυξη ενός pilot project σε κάθε ένα από αυτά ώστε να δούμε στην πράξη τις δυνατότητές τους. Έπειτα, σκεφτήκαμε τρία πιθανά σενάρια ή αλλιώς τρεις ιστορίες χρηστών (user stories) που θα μπορούσαν να υλοποιηθούν σε αυτό το στάδιο. Αφότου, προχωρήσαμε στην ανάπτυξη των παραπάνω, συλλέξαμε τα αποτελέσματα όλης αυτής της έρευνας και καταλήξαμε χαρακτηρίζοντας ως καταλληλότερη πλατφόρμα το Rasa Framework.

Σε αυτό το σημείο ξεκινήσαμε την συγγραφή του κειμένου, αφότου ολοκληρώσαμε το pilot project. Η συγγραφή προχώρησε σταδιακά, παράλληλα όμως με την υλοποίηση.

Έπειτα, έπρεπε να θέσουμε τα πιθανά σενάρια που αφορούν τους χρήστες. Συγκεκριμένα, θέλαμε να καλύψουμε ερωτήματα από τρεις τύπους φοιτητών οι οποίοι χαρακτηρίζονται ως personas.

Persona 1: Υποψήφιος φοιτητής ο οποίος αναζητά πληροφορίες πριν κάνει την αίτηση

Persona 2: Φοιτητής ο οποίος πρόσφατα ξεκίνησε τις μεταπτυχιακές του σπουδές

Persona 3: Φοιτητής ο οποίος σύντομα θα τελειώσει τις σπουδές του

Με βάση αυτούς τους τύπους χρηστών, χτίσαμε τα User Stories. Συγκεντρώσαμε όλες τις πληροφορίες που μπορεί να προσφέρει το chatbot, καταγράφοντας περίπου 30 user stories σε ένα board/πίνακα.

Αφού λοιπόν είχαμε σχεδιάσει και καταχωρήσει τα User Stories και επομένως το κύριο κομμάτι της εφαρμογής μας, προχωρήσαμε στην υλοποίηση του chatbot με όνομα WiBot. Η υλοποίηση είχε αρκετές προκλήσεις, μιας και ανακαλύπταμε ένα νέο τομέα. Πέρα από τεχνικά ζητήματα που προέκυψαν σε ποικίλα σημεία, κατά το στάδιο της ανάπτυξης του WiBot, χρειάστηκε επιπλέον η εκμάθηση της γλώσσα Python. Το Rasa δίνει την επιλογή της συγγραφής custom κώδικα σε Python, πράγμα το οποίο μας έδωσε επιπλέον κίνητρο για να συνεχίσουμε και να προσθέσουμε περισσότερες δυνατότητες στο chatbot μας.

Έχοντας ολοκληρώσει την υλοποίηση και μετά από πολλές δοκιμές, αποφασίσαμε να δώσουμε την εφαρμογή σε χρήστες για να την δοκιμάσουν αλλά και στην συνέχεια να την αξιολογήσουν. Αυτήν ήταν η πρώτη φάση αξιολόγησης του WiBot.

Μελετήσαμε τις αξιολογήσεις και σε επόμενο στάδιο διερευνήσαμε τα σχόλια και τις προτεινόμενες διορθώσεις από τους χρήστες. Τα περισσότερα από αυτά κρίθηκαν ως καλές προσθήκες οπότε προσχωρήσαμε στις αντίστοιχες ενέργειες.

Έπειτα ακολούθησε η δεύτερη φάση αξιολόγησης, στην οποία ζητήσαμε από τους ίδιους χρήστες να πειραματιστούν πάλι με το chatbot και να καταγράψουν την γνώμη τους γι' αυτό.

4.3 Διαγράμματα

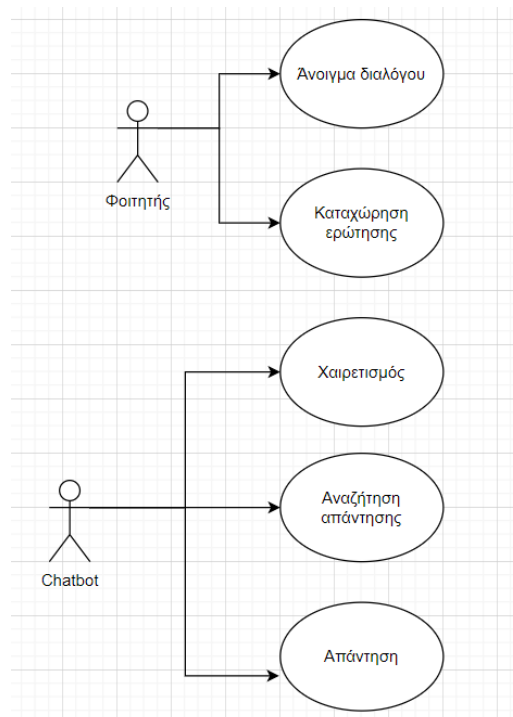
Σε αυτό το σημείο, παρουσιάζονται βασικά διαγράμματα όπως αρχιτεκτονικής του συστήματος καθώς και ορισμένα παραδείγματα πάνω σε βασικές λειτουργίες του chatbot. Για τα διαγράμματα χρησιμοποιήσαμε την γλώσσα UML.

Η Unified Modeling Language (UML) είναι μια γραφική γλώσσα για την οπτική αναπαράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Η UML στοχεύει στο σχεδιασμό αντικειμενοστραφών συστημάτων. Η UML είναι μια κοινή γλώσσα για επιχειρησιακούς αναλυτές, αρχιτέκτονες λογισμικού και προγραμματιστές που χρησιμοποιείται για την περιγραφή, τον καθορισμό, το σχεδιασμό και την τεκμηρίωση υφιστάμενων ή νέων επιχειρηματικών διαδικασιών, δομής και συμπεριφοράς

τεχνουργημάτων συστημάτων λογισμικού. [19] Μπορεί να εφαρμοστεί σε διαφορετικούς τομείς εφαρμογών όπως τραπεζικά, χρηματοοικονομικά κ.λπ.

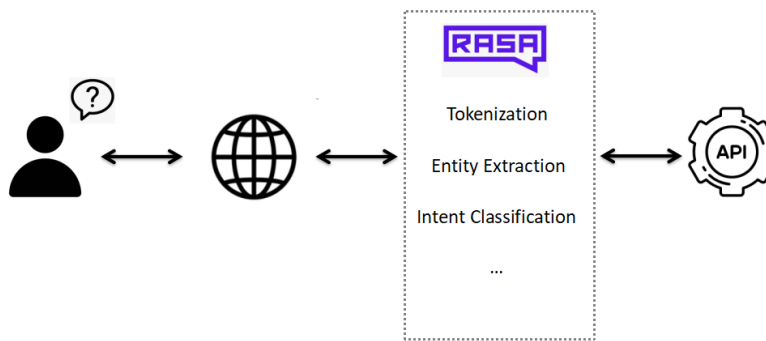
Η UML ορίζει διάγραμμα περιπτώσεων χρήσης (use case diagram), διαγράμματα δομής (class diagram και object diagram), διαγράμματα συμπεριφοράς (statechart diagram, activity diagram, sequence diagram, collaboration diagram) και διαγράμματα δομής υλοποίησης (component diagram, deployment diagram).

Στην εικόνα 4.3.1 παρουσιάζεται ένα Use case diagram (διάγραμμα ενεργειών) με τις γενικές ενέργειες από πλευράς χρήστη - φοιτητή και chatbot. Γενικά, τα διαγράμματα ενεργειών περιγράφουν τους τρόπους χρήσης ενός συστήματος.



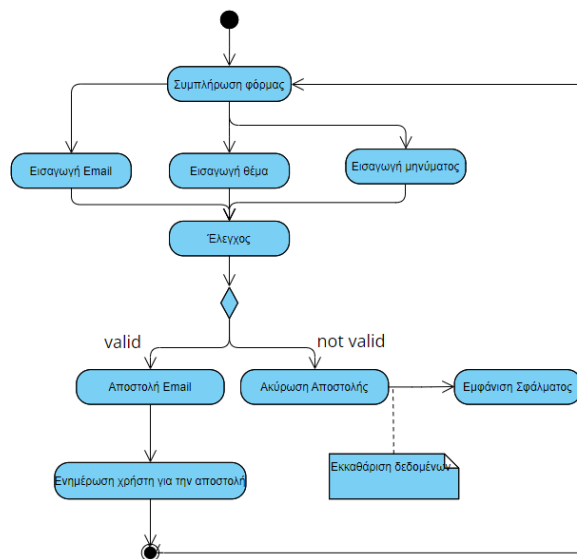
Εικόνα 4.3.1 Διάγραμμα ενεργειών

Όσο αφορά την δομή – αρχιτεκτονική του WiBot, ο χρήστης εισάγει ένα μήνυμα, κυρίως μια ερώτηση. Η ερώτηση διαβιβάζεται μέσα από ένα κανάλι του web, στο Back end της εφαρμογής που περιέχει την δομή του Rasa. Το chatbot για να απαντήσει ή θα αναζητήσει την απάντηση εσωτερικά, με βάση το μοντέλο του, ή θα ζητήσει την απάντηση από το API. Ορισμένες απαντήσεις, δίνονται δυναμικά, με κατάλληλη κλήση στο API του Τμήματος. Μόλις βρει την κατάλληλη απάντηση, επιστρέφει στον χρήστη το αντίστοιχο μήνυμα.



Εικόνα 4.3.2 Αρχιτεκτονική WiBot

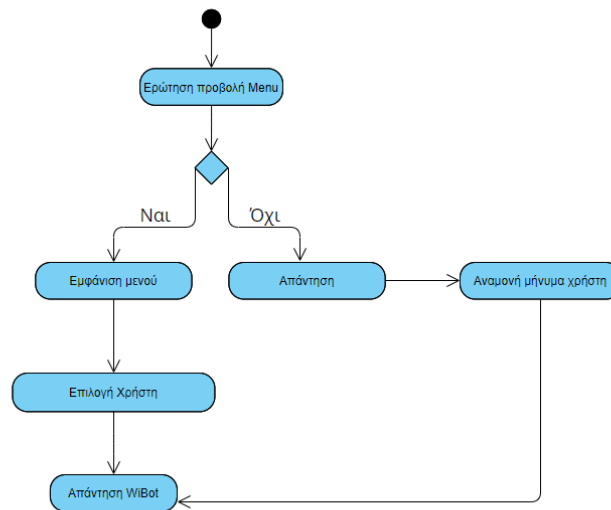
Στο παρακάτω διάγραμμα, εμφανίζεται η λειτουργία της αποστολής email. Αποτελεί ένα παράδειγμα διαγράμματος δραστηριοτήτων. Ουσιαστικά, δείχνει το workflow για την συγκεκριμένη δραστηριότητα και περιλαμβάνει διακλάδωση υπό συνθήκη (branch), για τον έλεγχο των στοιχείων της φόρμας. Ανάλογα με το αν επιτύχει ή αποτύχει ακολουθούν και οι αντίστοιχες ενέργειες.



Εικόνα 4.3.3 Διάγραμμα activity “Αποστολή Email”

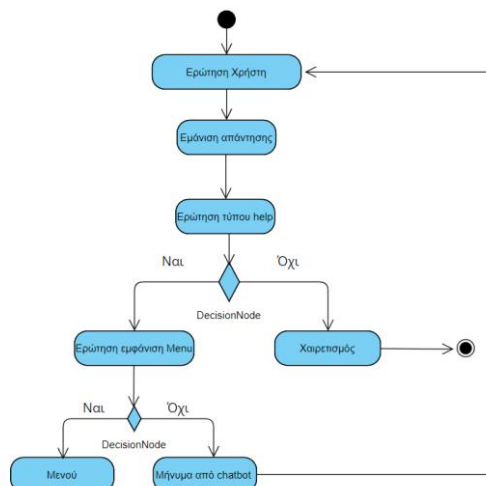
Επιπρόσθετα, ακολουθεί ένα παράδειγμα για την εμφάνιση του μενού επιλογών. Αποτελεί μια ενεργεία η οποία λαμβάνει χώρα πολύ συχνά σε ένα τυπικό διάλογο. Αρχικά, γίνεται η ερώτηση από το WiBot προς τον χρήστη, για την προβολή του μενού επιλογών. Εδώ ορίζεται ένας κόμβος μιας και ο χρήστης μπορεί να απαντήσει είτε θετικά είτε αρνητικά. Στην περίπτωση που απαντήσει με «Ναι», προβάλλεται το μενού, ο χρήστης συνεχίζει σε μια επιλογή ερώτησης και στην συνέχεια το WiBot θα δώσει την απάντησή του. Στην περίπτωση

που ο χρήστης απαντήσει με «Όχι», το WiBot εμφανίζει ένα μήνυμα, αναμένει την απάντηση του χρήστη και έπειτα συνεχίζει δίνοντας την απάντηση του.



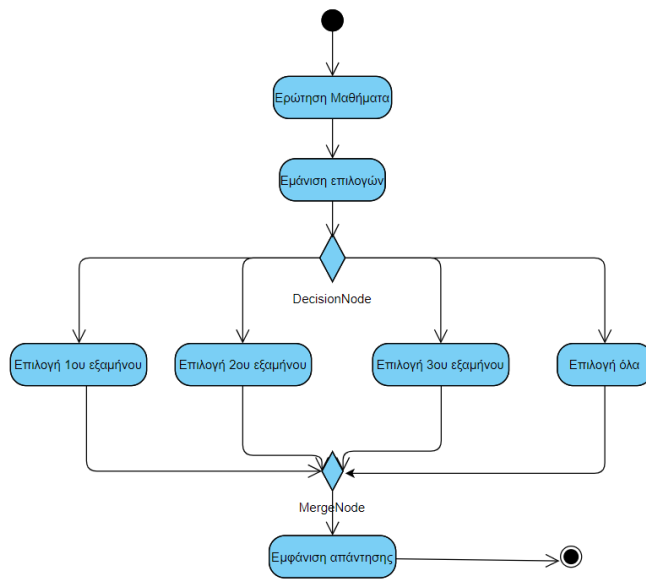
Εικόνα 4.3.4 Διάγραμμα activity “menu”

Στο ακόλουθο διάγραμμα, βλέπουμε την ροή ενεργειών σε μια γενική συζήτηση. Πιο συγκεκριμένα, ο χρήστης υποβάλλει την ερώτησή του, εμφανίζεται η απάντηση του WiBot, και ακολουθεί η ερώτηση τύπου help. Εδώ το chatbot ρωτάει τον χρήστη εάν μπορεί να βοηθήσει σε κάτι άλλο. Αν ο χρήστης απαντήσει «Ναι», ακολουθεί η ερώτηση για την εμφάνιση του μενού επιλογών και ένας αντίστοιχος κόμβος. Αν ο χρήστης απαντήσει «Όχι», το WiBot επιστρέφει έναν χαιρετισμό και κλείνει την συζήτηση.

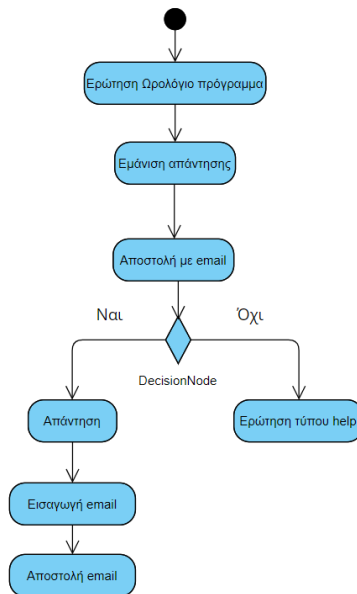


Εικόνα 4.3.5 Διάγραμμα activity “Help-menu”

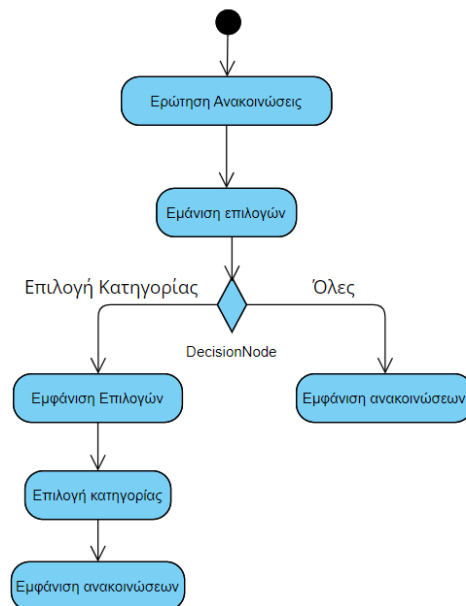
Τέλος, ακολουθούν τρία παραδείγματα διαλόγων για τα “Μαθήματα”, το “Ωρολόγιο Πρόγραμμα” και τις “Ανακοινώσεις”.



Εικόνα 4.3.6 Διάγραμμα activity “Μαθήματα”



Εικόνα 4.3.7 Διάγραμμα activity “Ωρολόγιο Πρόγραμμα”



Εικόνα 4.3.8 Διάγραμμα activity “Ανακοινώσεις”

4.4 Ανάπτυξη

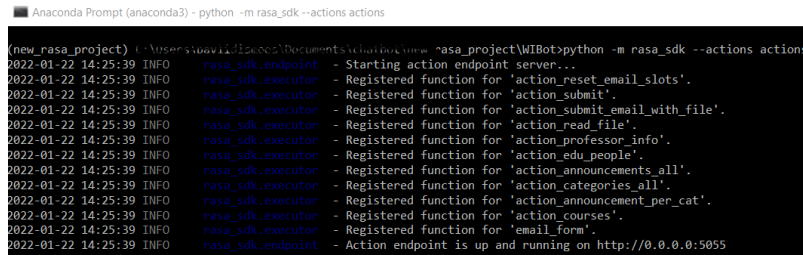
Όπως αναφέρθηκε στα πρώτα κεφάλαια, στο πειραματικό στάδιο έγινε η εγκατάσταση του Rasa Framework, και υλοποιήθηκε το pilot project. Αφότου τελικά κρίθηκε ως το καταλληλότερο εργαλείο, έγινε η ανάλυση των απαιτήσεων, έγινε καταγραφή των user stories και έπειτα προχωρήσαμε στην ανάπτυξη. Δημιουργήθηκαν intents, απαντήσεις, ιστορίες αλλά και ειδικά actions για την προσθήκη διάφορων ενεργειών που θα δούμε και αναλυτικότερα στην συνέχεια.

Πέρα από το τεχνικό κομμάτι, τέθηκε μια βασική δομή συζήτησης, την οποία ακολουθεί το chatbot σε όλες τις ιστορίες. Αρχικά, ξεκινά με το WiBot (το chatbot) να αποστέλλει ένα χαιρετισμό και να ρωτά τον χρήστη για την βοήθεια που θέλει να του προσφέρει. Ο χρήστης στέλνει το ερώτημά του και το chatbot αναζητά την απάντηση. Εφόσον, μπορεί να απαντήσει, στέλνει την απάντηση στον χρήστη και επιπλέον ρωτά αν μπορεί να τον βοηθήσει σε κάτι άλλο. Εάν ο χρήστης απαντήσει θετικά, συνεχίζεται η συζήτηση, διαφορετικά τερματίζεται η συνομιλία. Στην περίπτωση που το chatbot ερωτηθεί κάτι το οποίο δεν γνωρίζει, θα δώσει μια default απάντηση, στην οποία δηλώνει ότι δεν διαθέτει τέτοιες πληροφορίες.

Στην συνέχεια ακολουθούν λεπτομέρειες σχετικά με την ανάπτυξη του WiBot, από την υλοποίηση των custom actions έως το τελικό view της σελίδας που φιλοξενεί το chatbot.

4.4.2 Custom Actions

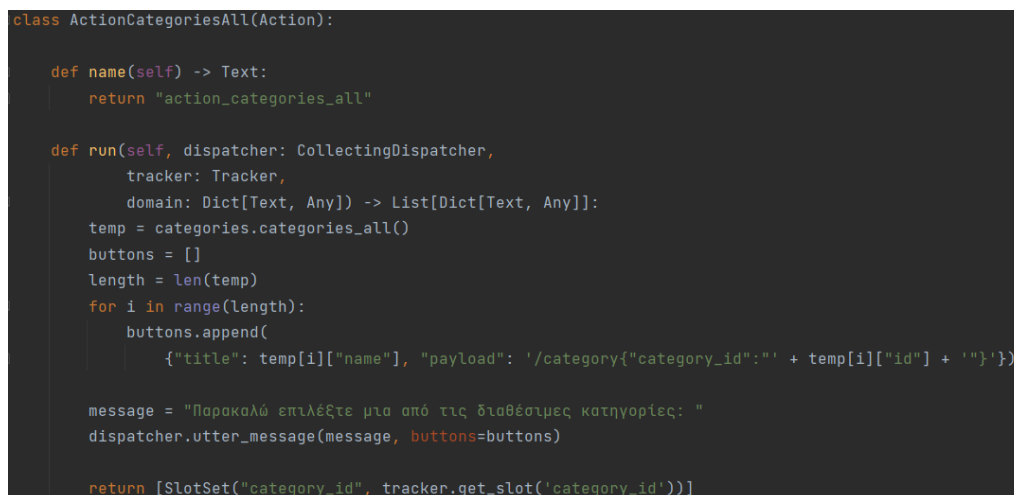
Όπως προαναφέρθηκε, τα custom actions αποτελούν μια ειδική μορφή απάντησης και χρησιμοποιούνται όταν απαιτείται προσαρμοσμένος κώδικας. Ειδικότερα, η υλοποίησή τους έχει γίνει με συγγραφή κώδικα σε python.



```
Anaconda Prompt (anaconda3) - python -m rasa_sdk --actions actions
(new_rasa_project) C:\Users\doan11\Documents\actions\new_rasa_project\WIBot>python -m rasa_sdk --actions actions
2022-01-22 14:25:39 INFO rasa_sdk.endpoint - Starting action endpoint server...
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_reset_email_slots'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_submit'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_submit_email_with_file'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_read_file'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_professor_info'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_edu_people'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_announcements_all'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_categories_all'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_announcement_per_cat'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'action_courses'.
2022-01-22 14:25:39 INFO rasa_sdk.executor - Registered function for 'email_form'.
2022-01-22 14:25:39 INFO rasa_sdk.endpoint - Action endpoint is up and running on http://0.0.0.0:5055
```

Εικόνα 4.4.3 Terminal run actions

Στην παρακάτω εικόνα (Εικόνα 4.4.4) παρουσιάζεται ένα παράδειγμα custom action, και ειδικότερα επιστρέφει μια λίστα με τις διαθέσιμες κατηγορίες ανακοινώσεων.



```
class ActionCategoriesAll(Action):
    def name(self) -> Text:
        return "action_categories_all"
    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        temp = categories.categories_all()
        buttons = []
        length = len(temp)
        for i in range(length):
            buttons.append(
                {"title": temp[i]["name"], "payload": '/category{"category_id":"' + temp[i]["id"] + '"}'})
        message = "Παρακαλώ επιλέξτε μια από τις διαθέσιμες κατηγορίες: "
        dispatcher.utter_message(message, buttons=buttons)
        return [SlotSet("category_id", tracker.get_slot('category_id'))]
```

Εικόνα 4.4.4 Παράδειγμα custom action

Αναπτύχθηκαν λοιπόν διάφορες κλάσεις, εξυπηρετώντας ποικίλους σκοπούς. Στον ακόλουθο πίνακα, παρουσιάζονται συγκεντρωτικά οι υλοποιημένες μέθοδοι, με την ονομασία τους καθώς επίσης και μια σύντομη περιγραφή που αντιστοιχεί στην κάθε μία.

Ονομασία Κλάσης	Σύντομη Περιγραφή
ActionSubmit	Αποστολή email
ActionSubmitWithFile	Αποστολή email με επισυναπτόμενο αρχείο PDF
ActionResetEmailSlots	Αφαιρεί τα περιεχόμενα στην φόρμα του email

ActionReadFile	Διαβάζει αρχεία PDF, και τα μετατρέπει σε κείμενο-text.
ActionProfessorInfo	Ανάκτηση των στοιχείων επικοινωνίας ενός συγκεκριμένου καθηγητή (Κλήση στο API)
ActionEduPeople	Ανάκτηση των στοιχείων επικοινωνίας των καθηγητών (Κλήση στο API)
ActionAnnouncementsApi	Κλήση στο API για την ανάκτηση των ανακοινώσεων
ActionCategoriesAll	Κλήση στο API για την ανάκτηση των κατηγοριών των ανακοινώσεων
ActionAnnouncementPerCat	Κλήση στο API για την ανάκτηση ανακοινώσεων ανά κατηγορία
ActionCourses	Επιστρέφει τα μαθήματα ανά εξάμηνο χειμερινό ή εαρινό

Πίνακας 4.4.1 Custom Actions

Σε ορισμένες υλοποιήσεις, όπως φαίνεται και στον πίνακα, έχουν δημιουργηθεί κλήσεις σε API(API calls). Οι συγκεκριμένες κλήσεις αφορούν στοιχεία που μας δίνονται από το Τμήμα. Με αυτόν τον τρόπο συλλέγουμε δυναμικές πληροφορίες, όπως οι ανακοινώσεις του Τμήματος, στοιχεία καθηγητών κ.α. Αναλυτικότερα, υπάρχει διαθέσιμο documentation στον εξής σύνδεσμο https://github.com/apavlidis/IT_API/wiki/API-Documentation.

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι υλοποιήσεις που αφορούν την εξαγωγή κειμένου από PDF και η αποστολή email μέσω του chatbot.

Συγκεκριμένα, για την εκτέλεση ορισμένων σεναρίων χρειάστηκε να διαβάσουμε αρχεία επέκτασης PDF. Για τον σκοπό αυτό, έγινε εκτενής αναζήτηση ώστε να βρεθεί η κατάλληλη βιβλιοθήκη. Τελικά, επιλέχθηκε το εργαλείο Apache Tika [20]. Εντοπίζει και εξάγει μεταδεδομένα και κείμενο από διαφορετικούς τύπους αρχείων (για παράδειγμα υπολογιστικά φύλλα, έγγραφα κειμένου, εικόνες, PDF). Για την εγκατάσταση, εκτελούμε την παρακάτω εντολή:

➤ `pip install tika`

Στην συνέχεια, με την δημιουργία του custom action, χρησιμοποιούμε την μέθοδο `from_file` για την εξαγωγή δεδομένων, από το αρχείο που δίνεται σαν παράμετρος.

Για την αποστολή email, χρησιμοποιήθηκε η βιβλιοθήκη `smtplib` της Python. Για τις ανάγκες, δημιουργήθηκε λογαριασμός (gmail) ηλεκτρονικού ταχυδρομείου για το chatbot. Στην συνέχεια, δημιουργήθηκε ένα αρχείο `sendEmail.py`. Στα custom actions, ορίζονται παράμετροι όπως είναι το email του παραλήπτη, το θέμα και το κείμενο του email.

Επιπλέον, δημιουργήθηκαν τέσσερα αρχεία επέκτασης .py, τα οποία περιλαμβάνουν υλοποιημένες μεθόδους για τα εξής:

categories: Η categories_all() επιστρέφει τις κατηγορίες των ανακοινώσεων.

announcements: Η μέθοδος announcements(), επιστρέφει τις ανακοινώσεις από όλες τις κατηγορίες. Η announcements_category(category_id), επιστρέφει τις ανακοινώσεις ανά κατηγορία, εισάγοντας ως παράμετρο το id της κατηγορίας.

eduPeople: Η professor_info(professor_name), επιστρέφει πληροφορίες και στοιχεία ενός συγκεκριμένου καθηγητή. Ως παράμετρο ζητείται το ονοματεπώνυμο (ή το επώνυμο) του καθηγητή.

Η edu_people(), επιστρέφει πληροφορίες / στοιχεία όλων των καθηγητών.

sendEmail: Η send_email(from_email, email, subject, message) εκτελεί την αποστολή email. Ως παράμετρο δέχεται το email του αποστολέα, το email του παραλήπτη, το θέμα και το μήνυμα που θα συμπεριληφθεί. Επιπλέον, περιέχει την send_email_with_file(from_address, to_address, subject, message, file_to_send) παρόμοια με την send_email(), με διαφορά ότι περιέχει επισυναπτόμενο αρχείο στην αποστολή email. Χρησιμοποιείται σε δύο σενάρια και συγκεκριμένα για την αποστολή του ωρολογίου ή του ακαδημαϊκού προγράμματος.

Παρακάτω θα δούμε μια προς μια τις κλάσεις που υλοποιήσαμε καθώς και αντίστοιχα στιγμιότυπα.

ActionSubmit

Η κλάση ActionSubmit αναλαμβάνει να υποβάλει, δηλαδή να στείλει τελικά το email. Καλεί με την σειρά της την send_email() που έχει υλοποιηθεί στο αρχείο sendEmail.py. Στην συνέχεια, ελέγχουμε την μεταβλητή smail. Εφόσον έχει τιμή, δηλώνει ότι έχει παρουσιαστεί λάθος κατά την εκτέλεση, όποτε επιστρέφει στον χρήστη αντίστοιχο μήνυμα λάθους. Εναλλακτικά, ενημερώνει τον χρήστη με αντίστοιχο μήνυμα επιτυχής αποστολής.

```
class ActionSubmit(Action):
    def name(self) -> Text:
        return "action_submit"

    def run(self, dispatcher, tracker: Tracker, domain: "DomainDict", ) -> List[Dict[Text, Any]]:
        smail = sendEmail.send_email("wchatbotinfo@gmail.com", tracker.get_slot("email"), tracker.get_slot("subject"),
                                     tracker.get_slot("message"))

        if smail:
            SlotSet("email", None)
            dispatcher.utter_message(smail)
        else:
            dispatcher.utter_message(
                "📧 Το email έχει αποσταλεί στην διεύθυνση: {}".format(tracker.get_slot("email")))

        return [SlotSet("email", None), SlotSet("subject", None), SlotSet("message", None)]
```

Εικόνα 4.4.5 ActionSubmit

ActionSubmitWithFile

Αντίστοιχη λογική ακολουθεί και η ActionSubmitWithFile, η οποία αναλαμβάνει την αποστολή email με πρόσθετο επισυναπτόμενο αρχείο.

```
class ActionSubmitWithFile(Action):
    def name(self) -> Text:
        return "action_submit_email_with_file"

    def run(self, dispatcher, tracker: Tracker, domain: "DomainDict" ...) -> List[Dict[Text, Any]]:
        subject = "Αρχείο Pdf"
        message = "Παρακάτω θα βρείτε το σχετικό αρχείο"
        if file_to_send:
            smail = sendEmail.send_email_with_file("wichtatbotinfo@gmail.com", tracker.get_slot("email"),
                                                  subject, message, file_to_send)

            if smail:
                SlotSet("email", None)
                dispatcher.utter_message(smail)
            else:
                dispatcher.utter_message(
                    "📧 Το email έχει αποσταλεί στην διεύθυνση: {}".format(tracker.get_slot("email")))

        return [SlotSet("email", None)]
```

Εικόνα 4.4.6 ActionSubmitWithFile

ActionResetEmailSlots

Η κλάση ActionResetEmailSlots κάνει reset, εκκαθαρίζει τρία slots, που συμπληρώνονται για την αποστολή email. Συγκεκριμένα το email, το θέμα και το μήνυμα.

```
class ActionResetEmailSlots(Action):
    def name(self) -> Text:
        return "action_reset_email_slots"

    def run(self, dispatcher, tracker, domain):
        return [SlotSet("email", None), SlotSet("subject", None), SlotSet("message", None)]
```

Εικόνα 4.4.7 ActionResetEmailSlots

ActionReadFile

Η ActionReadFile, “διαβάζει” αρχεία τύπου PDF, και τα μετατρέπει σε κείμενο text. Στην υλοποίηση της ελέγχουμε αρχικά ποιο intent προηγήθηκε, ώστε να διαβάσει και να επιστρέψει το αντίστοιχο κείμενο. Η πρώτη περίπτωση αφορά στο ακαδημαϊκό ημερολόγιο και η δεύτερη στο εβδομαδιαίο πρόγραμμα. Στην συνέχεια αφότου απαντήσει, ρωτά τον χρήστη εάν θέλει να του αποστείλει το αρχείο στο email του.

```

class ActionReadFile(Action):
    def name(self):
        return "action_read_file"

    async def run(self, dispatcher, tracker, domain):

        from tika import parser
        if tracker.latest_message['intent'].get('name') == "semester_program":
            file = "files/2021_22_Ακαδημαϊκό_Ημερολόγιο.pdf"
            parsed_pdf = parser.from_file(file)
            data = parsed_pdf['content']
            data = data.replace("\n\n", "\n\r")
        elif tracker.latest_message['intent'].get('name') == "year_program":
            file = "files/2021_22_Χειμ_Εξ_Εβδομαδιαίο_Πρόγραμμα.pdf"
            parsed_pdf = parser.from_file(file)
            data = parsed_pdf['content']
            data = data.replace("\n\n", "\n")

        dispatcher.utter_message(data)

        file_to_send = file
        message = "Θα ήθελες να σου στείλω με email το αντίστοιχο αρχείο;"
        buttons = [{ 'title': "Ναι ✓",
                    'payload': '/send_email_with_file'},
                  { 'title': "Όχι ✕",
                    'payload': '/cancel_email'}]
        dispatcher.utter_message(message, buttons=buttons)
        return []

```

Εικόνα 4.4.8 ActionReadFile

ActionProfessorInfo

Η ActionProfessorInfo επιστρέφει πληροφορίες και στοιχεία επικοινωνίας για έναν συγκεκριμένο καθηγητή. Αρχικά, βρίσκει το slot *professor_name*, που περιέχει το ονοματεπώνυμο του καθηγητή. Το εισάγει σε μια μεταβλητή και στην συνέχεια την περνάει ως παράμετρο στην *professor_info()*. Εκείνη με την σειρά της αναλαμβάνει να καλέσει το API και να επιστρέψει τα επιθυμητά αποτελέσματα στην *professor*. Ακολουθεί η σύνταξη του μηνύματος που θα επιστρέψει το chatbot με τον τίτλο του καθηγητή, το ονοματεπώνυμο, τηλέφωνο, link, την σελίδα του στο researchGate και τέλος το email. Πριν τερματίσει “καθαρίζει” το slot *professor_name*, εισάγοντας την τιμή None.

```

class ActionProfessorInfo(Action):
    def name(self) -> Text:
        return "action_professor_info"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        professor_name = tracker.get_slot('professor_name')
        print("professor: ", professor_name)
        professor = eduPeople.professor_info(professor_name)
        dispatcher.utter_message(professor["title;lang-el"] + ' - ' + professor["displayName;lang-el"] + " \n ☎ "
                                + professor["telephoneNumber"] + " \n ☎ " + professor["labeledURI"] + " \n "
                                + professor["socialMedia"]["researchGate"] + " \n ", image=professor["secondarymail"])
        return [SlotSet("professor_name", None)]

```

Εικόνα 4.4.9 ActionProfessorInfo

ActionEduPeople

Παρόμοια υλοποίηση συναντάμε στην ActionEduPeople, με την διαφορά ότι επιστρέφεται μια εκτενής λίστα και φυσικά δεν έχουμε παράμετρο (καθηγητής).

```
class ActionEduPeople(Action):

    def name(self) -> Text:
        return "action_edu_people"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        temp = eduPeople.edu_people()

        length = len(temp)
        for i in range(length):
            dispatcher.utter_message('📧 ' + temp[i]["title"] + " | " +
                                     temp[i]["name"] + ' \n 📞 ' + temp[i][
                                         "telephoneNumber"] + " \n ☎ " + temp[i][
                                         "Uri"] + " \n ", image=temp[i]["mail"])

        return []
```

Εικόνα 4.4.10 ActionEduPeople

ActionAnnouncementsApi

Η ActionAnnouncementsApi επιστρέφει τις ανακοινώσεις. Καλεί την μέθοδο announcements() και τυπώνει μέσα σε ένα loop, για κάθε εγγραφή τον τίτλο της ανακοίνωσης, τον συντάκτη, την ημερομηνία και τον link προς την αντίστοιχη ανακοίνωση.

```
class ActionAnnouncementsApi(Action):

    def name(self) -> Text:
        return "action_announcements_all"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        temp = announcements.announcements()

        length = len(temp)

        for i in range(length):
            print(temp[i]["publisher"] + "τίτλος: " + temp[i]["title"])

            dispatcher.utter_message(
                format(temp[i]["title"]) + " \n" + temp[i]["publisher"] + " \n" + dateutil.parser.parse(
                    temp[i]["date"]).strftime(
                        '%m/%d/%Y') + " \n ☎" + "https://apps.iee.ihu.gr/announcements/announcement/" +
                    temp[i][
                        "id"])

        return []
```

Εικόνα 4.4.11 ActionAnnouncementsAPI

ActionCategoriesAll

Η ActionCategoriesAll αναλαμβάνει να επιστρέψει όλες τις κατηγορίες ανακοινώσεων. Όταν ο χρήστης ζητήσει να επιλέξει μία από τις κατηγορίες των ανακοινώσεων, καλείται αυτή η κλάση και επιστρέφει τις κατηγορίες σε μορφή buttons.

```
class ActionCategoriesAll(Action):

    def name(self) -> Text:
        return "action_categories_all"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        temp = categories.categories_all()
        buttons = []
        length = len(temp)
        for i in range(length):
            buttons.append(
                {"title": temp[i]["name"], "payload": '/category{"category_id":"' + temp[i]["id"] + '"}')

        message = "Παρακαλώ επιλέξτε μια από τις διαθέσιμες κατηγορίες: "
        dispatcher.utter_message(message, buttons=buttons)

        print("category_id: ", tracker.get_slot('category_id'))
        return [SlotSet("category_id", tracker.get_slot('category_id'))]
```

Εικόνα 4.4.12 ActionCategoriesAll

ActionAnnouncementPerCat

Η ActionAnnouncementPerCat, επιστρέφει τις ανακοινώσεις ανά κατηγορία. Συγκεκριμένα, βρίσκει το category_id, από την κατηγορία που έχει επιλέξει ο χρήστης και το περνάει ως παράμετρο στην announcements_category() η οποία με την σειρά της, επικοινωνεί με το API και στέλνει πίσω τα αποτελέσματα. Με μια επανάληψη for, για κάθε ένα Item τυπώνει τον τίτλο της ανακοίνωσης, τον συντάκτη, την ημερομηνία και τον link προς την αντίστοιχη ανακοίνωση.

```

class ActionAnnouncementPerCat(Action):

    def name(self) -> Text:
        return "action_announcement_per_cat"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        category_id = tracker.get_slot('category_id')

        if category_id:
            temp = announcements.announcements_category(category_id)
            length = len(temp)
            for i in range(length):
                print(temp[i]["publisher"], temp[i]["title"])

                dispatcher.utter_message(
                    temp[i]["publisher"] + " \n" + dateutil.parser.parse(temp[i]["date"]).strftime(
                        '%m/%d/%Y') + " \n " + format(
                            temp[i]["title"]) + " \n" + "https://apps.iee.ihu.gr/announcements/announcement/" +
                            temp[i][
                                "id"])
            else:
                print("No category id")
        return []

```

Εικόνα 4.4.13 ActionAnnouncementPerCat

ActionCourses

Η κλάση ActionCourses αναλαμβάνει να επιστρέψει τα μαθήματα ανάλογα με την επιλογή του χρήστη. Ειδικότερα, ελέγχει τέσσερις περιπτώσεις, εφόσον ο χρήστης επιλέξει να μάθει τα μαθήματα είτε για το πρώτο εξάμηνο είτε για το δεύτερο είτε για το τρίτο. Μια τελευταία περίπτωση αφορά την επιστροφή όλων των μαθημάτων ανά εξάμηνο.

```

class ActionCourses(Action):

    def name(self) -> Text:
        return "action_courses"

    def run(self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        semester = next(tracker.get_latest_entity_values('semester'), None)
        if str(semester) == '1' or str(semester) == '1o' or str(semester) == 'χειμερινό':
            dispatcher.utter_message(response="utter_courses_1")
            dispatcher.utter_message(response="utter_help_new")
        elif str(semester) == '2' or str(semester) == '2o' or str(semester) == 'εαρινό':
            dispatcher.utter_message(response="utter_courses_2")
            dispatcher.utter_message(response="utter_help_new")
        elif str(semester) == '3' or str(semester) == '3o' or str(semester) == 'τρίτο':
            dispatcher.utter_message(response="utter_courses_3")
            dispatcher.utter_message(response="utter_help_new")
        else:
            dispatcher.utter_message(text="Τα μεταπτυχιακά περιλαμβάνει τρία (3) διδακτικά εξάμηνα")
            dispatcher.utter_message(response="utter_ask_courses")

        return []

```

Εικόνα 4.4.14 ActionCourses

4.4.3 Fallback- Χειρισμός μηνυμάτων εκτός πεδίου εφαρμογής

Κατά την υλοποίηση παρατηρήθηκε ότι το chatbot, όταν δέχεται κάποιο μήνυμα άγνωστο για εκείνο, βρίσκει μια απάντηση από τις ιστορίες και την επιστρέφει. Αυτό αποτελεί μια γνωστή περίπτωση στην κοινότητα του Rasa. Μετά από αναζήτηση, βρήκαμε και χρησιμοποιήσαμε τα Fallbacks.

Το chatbot όταν δεχθεί μηνύματα με χαμηλό confidence, είτε θα απαντήσει με την καθορισμένη απάντηση για αυτές τις περιπτώσεις (πχ Συγγνώμη δεν κατάλαβα), είτε θα προσπαθήσει να αποσαφηνίσει την ερώτηση του χρήστη με βάση τα training data.

Χρησιμοποιώντας το FallbackClassifier το chatbot διαχειρίζεται πλέον μηνύματα με χαμηλό confidence. Το intent nlu_fallback θα προβλεφθεί όταν όλες οι άλλες προβλέψεις πρόθεσης πέφτουν κάτω από το διαμορφωμένο όριο confidence (threshold: 0.5) .

```
Next message:
κάτιπουδενέχεινόημα
{
  "text": "κάτιπουδενέχεινόημα",
  "intent": {
    "id": -1323183611925809070,
    "name": "bye",
    "confidence": 0.9847265481948853
  },
  "entities": [],
  "intent_ranking": [
    {
      "id": -1323183611925809070,
      "name": "bye",
      "confidence": 0.9847265481948853
    },
    {
      "id": -936042157051802265,
      "name": "contact_secretariat",
      "confidence": 0.004667667206376791
    },
    {
      "id": 4817568659726336991,
      "name": "semester_program",
      "confidence": 0.0022921727504581213
    }
  ],
},
},

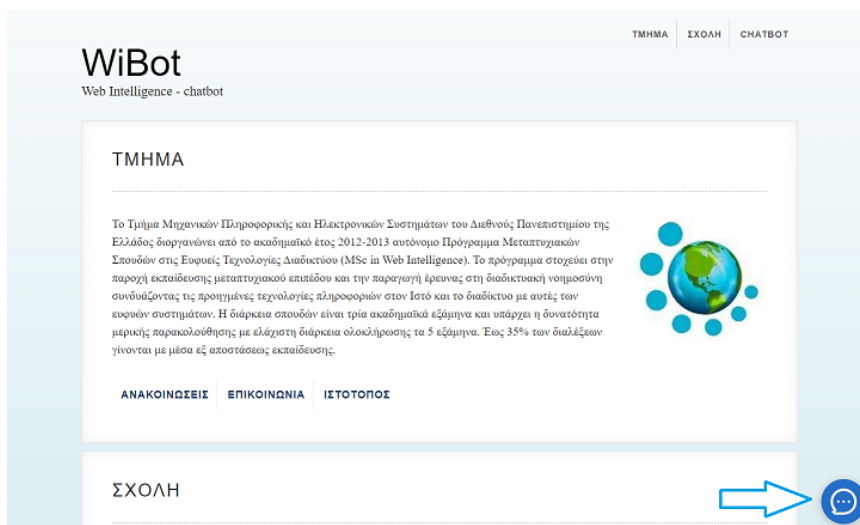
Next message:
κάτιπουδενέχεινόημα
{
  "text": "κάτιπουδενέχεινόημα",
  "intent": {
    "id": 5796948174827969644,
    "name": "out_of_scope",
    "confidence": 0.9880097508430481
  },
},
},
```

Εικόνα 4.4.15 Παραδείγματα

4.4.4 Index.html

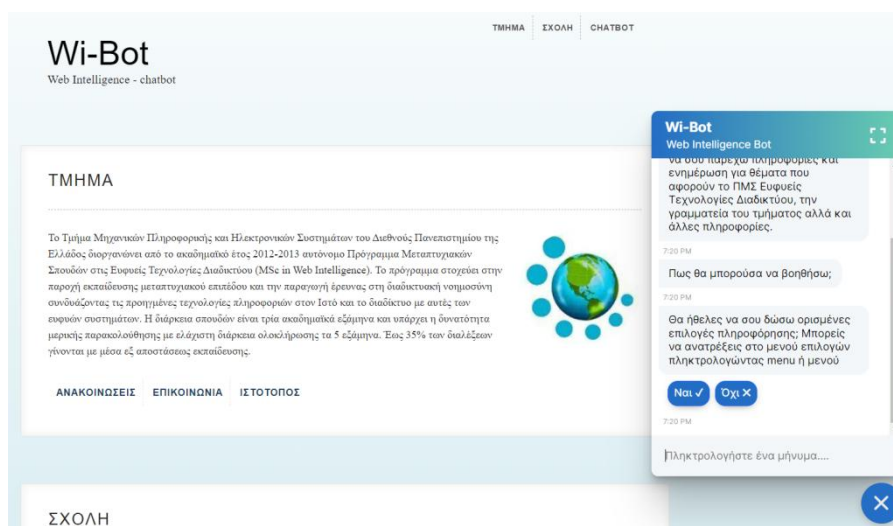
Ακόμη, δημιουργήθηκε η σελίδα index.html. Η σελίδα αυτή φιλοξενεί το chatbot και περιέχει γενικές πληροφορίες όπως για το Τμήμα, την σχολή και το chatbot. Κάθε section, διαθέτει συνδέσμους, όπως για παράδειγμα δίνεται η επίσημη ιστοσελίδα του Τμήματος, οι ανακοινώσεις, και οι τρόποι επικοινωνίας.

Πιο συγκεκριμένα, το παράθυρο διαλόγου βρίσκεται κάτω δεξιά στην οθόνη. Από προεπιλογή είναι "κλειστό".



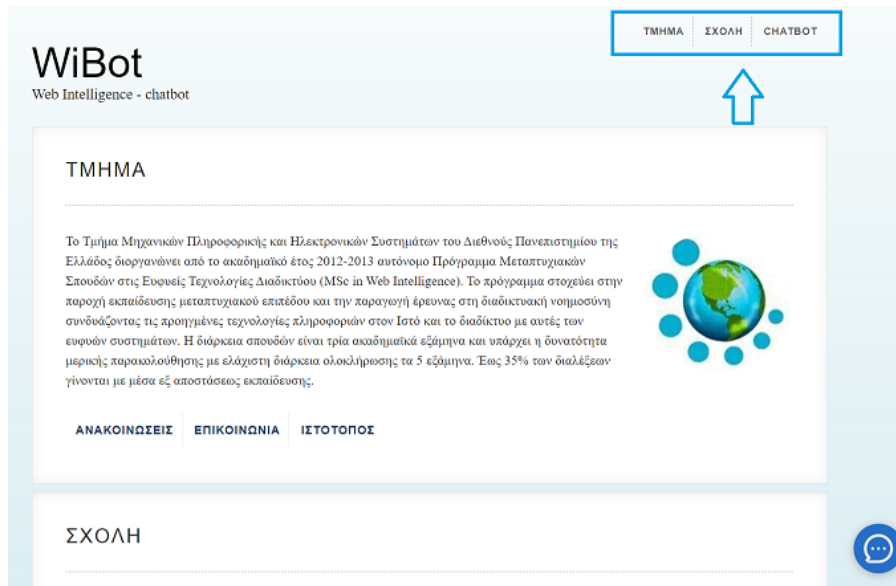
Εικόνα 4.4.16 Αρχική σελίδα

Ακολουθεί στιγμιότυπο με ανοιχτό το παράθυρο – chat.



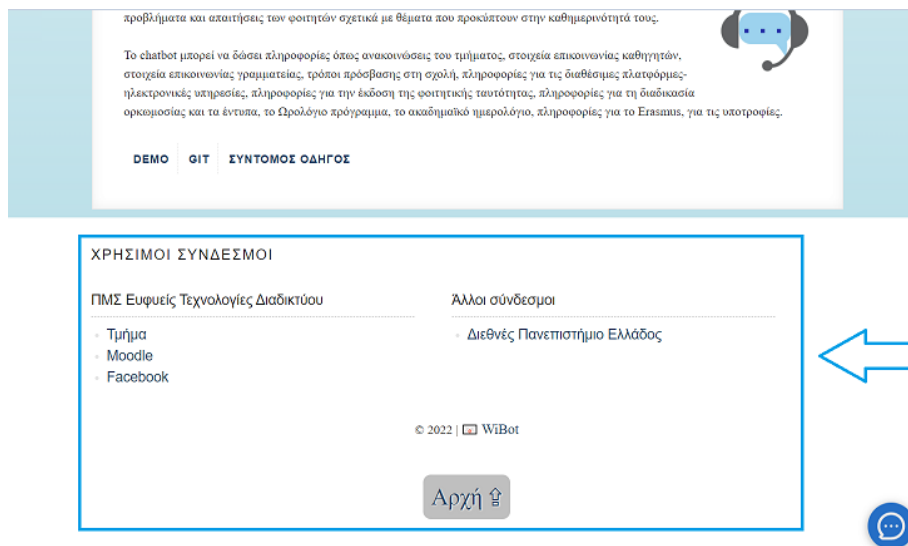
Εικόνα 4.4.17 Αρχική σελίδα με ανοιχτό διάλογο

Η επικεφαλίδα (header) της σελίδας, περιέχει το μενού πλοήγησης του εγγράφου (navigation menu). Οι διαθέσιμες επιλογές είναι Τμήμα, Σχολή, Chatbot. Κάθε επιλογή στέλνει τον χρήστη στο αντίστοιχο σημείο της σελίδας.



Εικόνα 4.4.18 Navigation menu

Το footer, ή υποσέλιδο, χρησιμοποιείται για την παροχή πληροφοριών και συγκεκριμένα συνδέσμων, που αφορούν το Τμήμα, την σελίδα του Τμήματος στο Moodle και την σελίδα του Τμήματος στο Facebook. Στην ενότητα «Άλλοι σύνδεσμοι» υπάρχει η επίσημη ιστοσελίδα του Πανεπιστημίου. Επιπλέον υπάρχει ένα κουμπί “Αρχή”, το οποίο κατευθύνει τον χρήστη, στο επάνω μέρος της σελίδας.



Εικόνα 4.4.19 Footer

Κάθε Τμήμα (section) της σελίδας παρουσιάζει ορισμένες γενικές πληροφορίες, δίνοντας όμως κατάλληλους συνδέσμους για περισσότερες λεπτομέρειες και πληροφορίες.

Στην ενότητα “Τμήμα” παρέχονται τρεις βασικοί σύνδεσμοι για τις ανακοινώσεις του Τμήματος, τους τρόπους επικοινωνίας και τον επίσημο ιστότοπο του Τμήματος.

Στην ενότητα “Σχολή”, έχουμε προσθέσει βασικές πληροφορίες, όπως έτος ίδρυσης, παραρτήματα και άλλα.

Ενώ η ενότητα “Chatbot” περιέχει πληροφορίες για το chatbot, με τους εξής συνδέσμους:

- Demo: Στο demo παρουσιάζουμε σε βίντεο, τις λειτουργικότητες του WiBot, μέσα από 3 χαρακτήρες (personas). Ο χρήστης μπορεί να παρακολουθήσει αυτό το βίντεο και για να καταλάβει πως χρησιμοποιείται ένα chatbot, αλλά και για να δει τις λειτουργίες του συγκεκριμένου.
- Git: Ορίζεται το αποθετήριο του κώδικα μας. Αυτή τη στιγμή είναι ορισμένο ως private αλλά μπορεί να διατεθεί σε όποιον το ζητήσει μετά από αίτημα.
- Σύντομος Οδηγός: Εδώ παρουσιάζουμε ένα οδηγό με χρήσιμες πληροφορίες για τον χρήστη. Παραθέτουμε τις πληροφορίες που μπορεί να προσφέρει το WiBot μαζί με ορισμένα στιγμιότυπα.



Εικόνα 4.4.20 Σελίδα

Για την αλλαγή της εμφάνισης, δημιουργήθηκε ένα αρχείο css (site.css) και ειδικό section (βλέπε <style>) όπου παραμετροποιήθηκε το layout του διαλόγου με την βοήθεια της css.

Για την απεικόνιση σε διαφορετικές συσκευές (responsive design) και φυσικά το καλύτερο user experience, δημιουργήθηκαν ερωτήματα μέσω (media queries). Τα ερωτήματα media queries, επιτρέπουν την εφαρμογή προτάσεων υπό όρους (conditional statements), ώστε να προσαρμόζουν τον ιστότοπο στην κάθε οθόνη όπως σε κινητή συσκευή, tablet ή οθόνη υπολογιστή. Στην συγκεκριμένη περίπτωση, έγιναν προσθήκες στην σχεδίαση, χρησιμοποιώντας τα Media queries για κινητές συσκευές και tablets.

```
/* responsive css screen mobile */
@media only screen and (max-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 601px) {...}
```

Εικόνα 4.4.21 Media queries

4.4.5 Socket.IO

Έπειτα, για να γίνει expose το widget, διερευνήθηκαν οι διαθέσιμες βιβλιοθήκες και τελικά επιλέχθηκε το socket io. Το Socket.IO είναι μια βιβλιοθήκη που επιτρέπει την επικοινωνία σε πραγματικό χρόνο, αμφίδρομη και βασισμένη σε συμβάντα μεταξύ του προγράμματος περιήγησης και του διακομιστή.

Στο αρχείο του credentials, ορίζεται ο παρακάτω κώδικας για την ενεργοποίηση του Socket.IO server:

```
socketio:
  user_message_evt: user_uttered
  bot_message_evt: bot_uttered
  session_persistence: true/false
```

Ο server του Rasa επεξεργάζεται τα μηνύματα και στέλνει μια απάντηση στο bot_uttered. Το user_uttered είναι το event όνομα, για τα μηνύματα του χρήστη, ενώ το bot_uttered είναι το event όνομα, για τα μηνύματα του bot.

```
socketio:
  user_message_evt: user_uttered
  bot_message_evt: bot_uttered
  session_persistence: true
```

Εικόνα 4.4.22 credentials.yml

Για την σύνδεση απαιτείται ο client του Socket.IO, συμπεριλαμβάνοντας ένα script από το CDN, πριν από το </body>. Η version που χρησιμοποιήθηκε ήταν η 1.0.1.

```

<script>!(function () {
  let e = document.createElement("script"),
      t = document.head || document.getElementsByTagName("head")[0];
  (e.src =
    "https://cdn.jsdelivr.net/npm/rasa-webchat@1.0.1/lib/index.js"),
    (e.async = !0),
    (e.onload = () => {
      window.WebChat.default(

```

Εικόνα 4.4.23 Δήλωση script

Έγινε παραμετροποίηση, ορίζοντας διάφορες λειτουργίες, όπως φαίνεται και στην εικόνα 4.4.24. Εδώ, ορίζονται ο τίτλος που φαίνεται στο chat, η γλώσσα που χρησιμοποιείται, οι εικόνες και τα εικονίδια του διαλόγου, κάποια μηνύματα ή λεκτικά αλλά απαραίτητος το url που τρέχει το Rasa (socketUrl).

```

window.WebChat.default(
  {
    selector: "#webchat",
    initPayload: "/intro",
    customData: {"language": "el"},
    socketUrl: "http://localhost:5051",
    socketPath: "/socket.io/",
    title: "Wi-Bot",
    subtitle: "Web Intelligence Bot",
    hideWhenNotConnected: false,
    profileAvatar: "/images/chatbot_round.jpg",
    openLauncherImage: "/images/comment.svg",
    closeImage: "/images/close.svg",
    showFullScreenButton: true,
    showMessageDate: true,
    connectingText: "Waiting for server...",
    inputTextFieldHint: "Πληκτρολογήστε ένα μήνυμα...",
    params: {"storage": "session"}
  },
  null
);
t.insertBefore(e, t.firstChild);

```

Εικόνα 4.4.24 Παραμετροποίηση Web chat

Για παράδειγμα με τις παραπάνω παραμέτρους, δώσαμε ορισμένες επιλογές για την προσαρμογή της οθόνης του διαλόγου. Παρακάτω φαίνονται τα εικονίδια, με τις αντίστοιχες λειτουργίες.



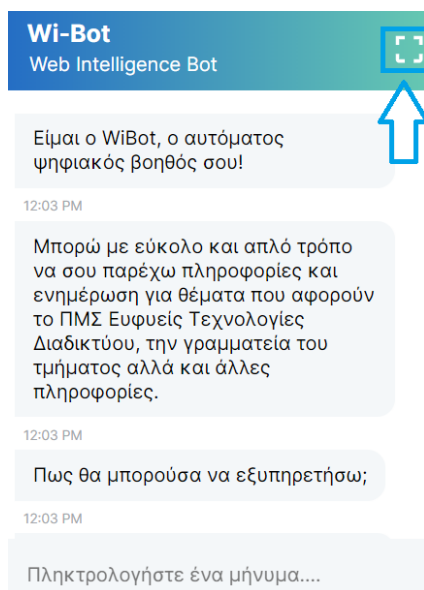
Μεγιστοποίηση: Μεγέθυνση σε πλήρη οθόνη



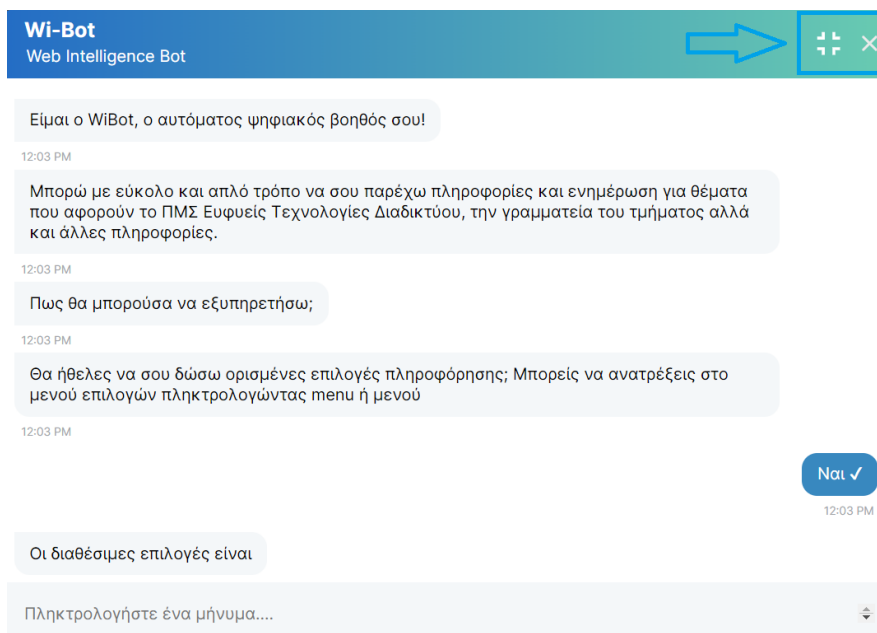
Ελαχιστοποίηση: Μικρότερη οθόνη



Κλείσιμο: Κλείσιμο διάλογου



Εικόνα 4.4.25 Μεγιστοποίηση παραθύρου



Εικόνα 4.4.26 Κλείσιμο-Ελαχιστοποίηση παραθύρου

4.4.6 Προβλήματα

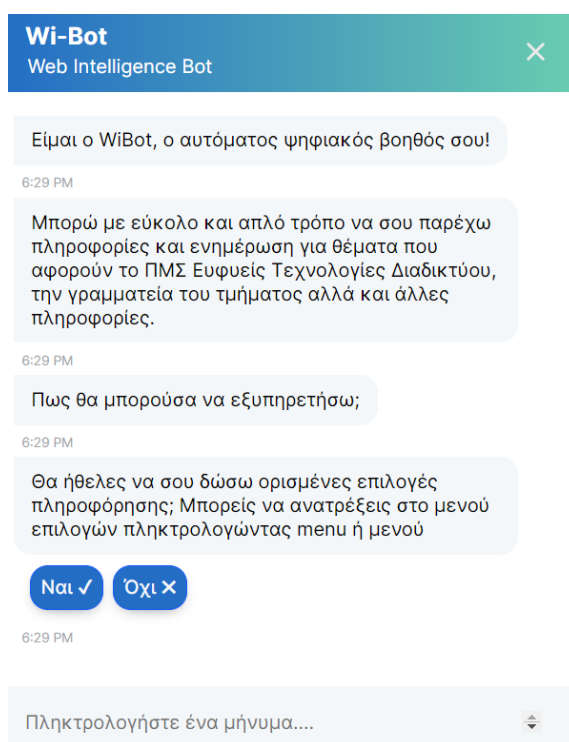
Κατά την ανάπτυξη του chatbot, παρατηρήσαμε σε διάφορα σημεία λάθος απαντήσεις. Ψάχνοντας και επαναλαμβάνοντας τους διάλογους, υπήρξαν περιπτώσεις που δεν βρέθηκε κάποιο δομικό ή τεχνικό λάθος. Εν ολίγοις, καταλήγουμε ότι το chatbot μπορεί να αντιληφθεί

σε ορισμένες περιπτώσεις λάθος Intent, με αποτέλεσμα να δώσει και λάθος απάντηση ή να απαντήσει με μήνυμα τύπου out of scope. Αυτό φυσικά είναι κάτι που αφορά το τι μαθαίνει το nlu μοντέλο και απαιτεί σίγουρα μεγαλύτερη μελέτη η δομή του και ο τρόπος που δημιουργείται τελικά το κάθε μοντέλο.

4.4.7 To WiBot

Σε αυτή την ενότητα παρουσιάζονται στιγμιότυπα από ορισμένα παραδείγματα διαλόγων.

Αρχικά, η εικόνα 4.4.27, εμφανίζει το αρχικό μήνυμα του chatbot, μόλις δηλαδή ο χρήστης ανοίξει το παράθυρο διαλόγου. Το chatbot συστήνεται, δίνει βασικές πληροφορίες και ρωτά τον χρήστη εάν θέλει να του εμφανίσει το μενού ή εάν θέλει ο ίδιος να πληκτρολογήσει την ερώτησή του. Εδώ, προστέθηκε μια σημείωση προς τον χρήστη που αφορά την μετάβαση του στο κεντρικό μενού.



Εικόνα 4.4.27 Αρχικό μήνυμα WiBot

Σε κάθε περίπτωση, αφού ο χρήστης ρωτήσει κάτι και δώσει την απάντησή του το WiBot, ρωτά τον χρήστη εάν θέλει να συνεχίσει στον διάλογο, με μία ερώτηση όπως στην παρακάτω εικόνα. Εάν ο χρήστης επιλέξει να συνεχίσει, δηλαδή απαντήσει στην ερώτηση «Ναι», το WiBot θα του δώσει ως επιλογή την εμφάνιση του κεντρικού μενού. Στην περίπτωση που ο χρήστης θέλει να τερματίσει τον διάλογο, θα απαντήσει όχι και το chatbot θα τερματίσει την συζήτηση.

Μπορώ να σε βοηθήσω σε κάτι άλλο;

6:32 PM

ναι

6:33 PM

Θα ήθελες να σου δώσω ορισμένες επιλογές πληροφόρησης; Μπορείς να ανατρέξεις στο μενού επιλογών πληκτρολογώντας menu ή μενού

Ναι ✓ Όχι ✕

6:33 PM

Πληκτρολογήστε ένα μήνυμα....

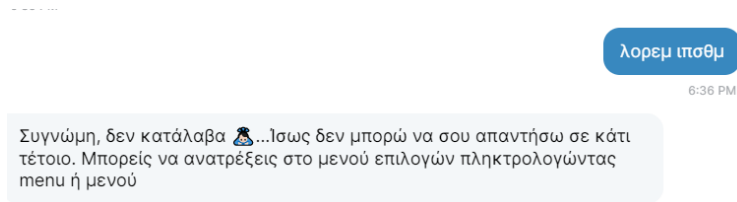
Εικόνα 4.4.28 Ερώτηση τύπου “help”

Ως κεντρικό μενού, θεωρούμε το μενού επιλογών που φαίνεται στη παρακάτω εικόνα (4.4.29). Ουσιαστικά, πρόκειται για buttons / κουμπιά, με τις διαθέσιμες πληροφορίες που μπορεί να δώσει το chatbot. Έτσι, ο χρήστης έχει πάντα καθοδήγηση στις βασικές επιλογές, δεν χρειάζεται να πληκτρολογήσει το αίτημά του και ο διάλογος γίνεται άμεσα και γρήγορα.



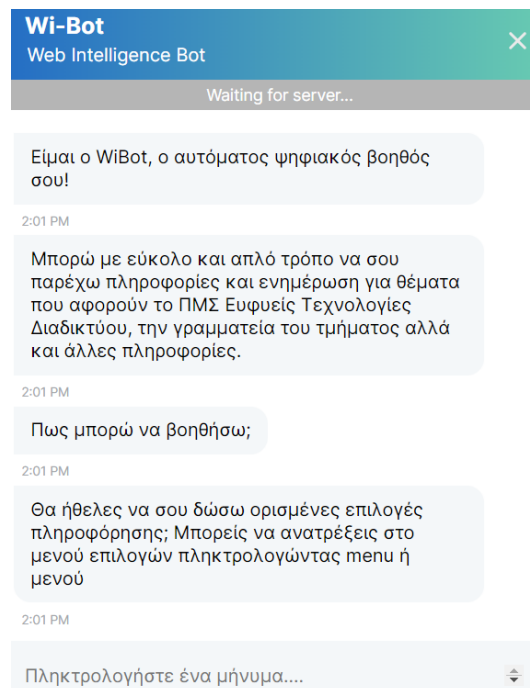
Εικόνα 4.4.29 Μενού επιλογών

Στην περίπτωση που ο χρήστης εισάγει κάποιο μήνυμα που δεν μπορεί να αναγνωρίσει ή να απαντήσει το chatbot (τύπου out of scope), επιστρέφει μια γενικού τύπου απάντηση «Συγγνώμη δεν κατάλαβα...».



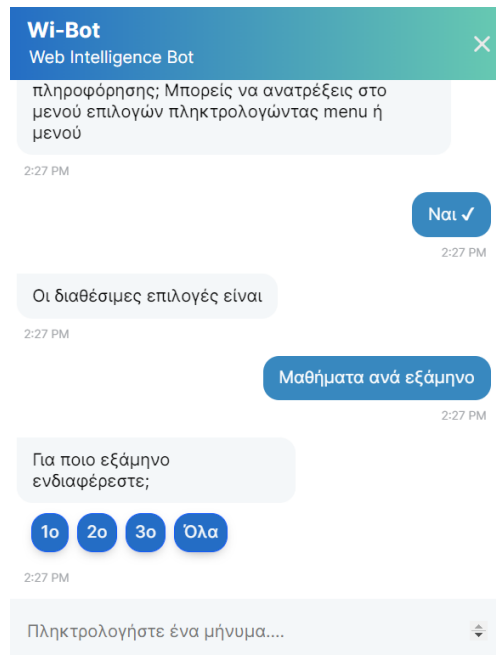
Εικόνα 4.4.30 Μήνυμα out of scope

Επιπλέον, όταν υπάρχουν προβλήματα σύνδεσης, εμφανίζεται ένα μήνυμα στην οθόνη, «Waiting for server...». Σε αυτή τη περίπτωση ο χρήστης δεν μπορεί να εισάγει κάποιο μήνυμα στο αντίστοιχο πλαίσιο μέχρι να αποκατασταθεί το πρόβλημα.



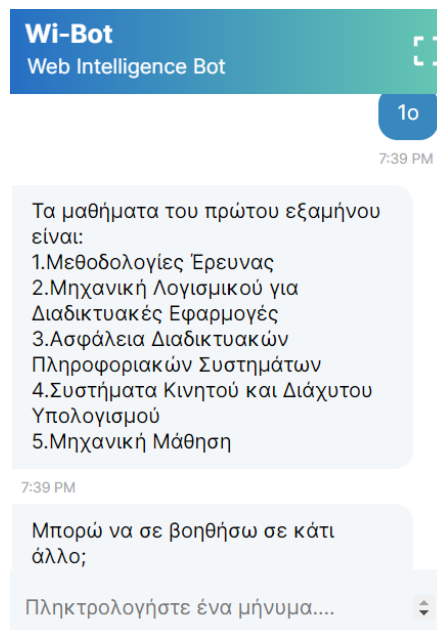
Εικόνα 4.4.31 Μήνυμα «Waiting for server»

Ακολουθούν ορισμένα παραδείγματα ερωτημάτων - απαντήσεων ή αλλιώς ιστοριών. Μία από τις πιο κοινές ερωτήσεις ενός φοιτητή ή υποψήφιου φοιτητή, αφορά τα μαθήματα που περιλαμβάνει το κάθε εξάμηνο. Εάν λοιπόν ο φοιτητής ρωτήσει το WiBot για τα μαθήματα, τότε το WiBot επιστρέφει ορισμένες επιλογές για τον χρήστη, όπως φαίνεται και στην παρακάτω εικόνα. Συγκεκριμένα, μπορεί να επιλέξει να δει, τα μαθήματα ανά εξάμηνο, αλλά επίσης να δει και όλα τα μαθήματα συγκεντρωτικά.



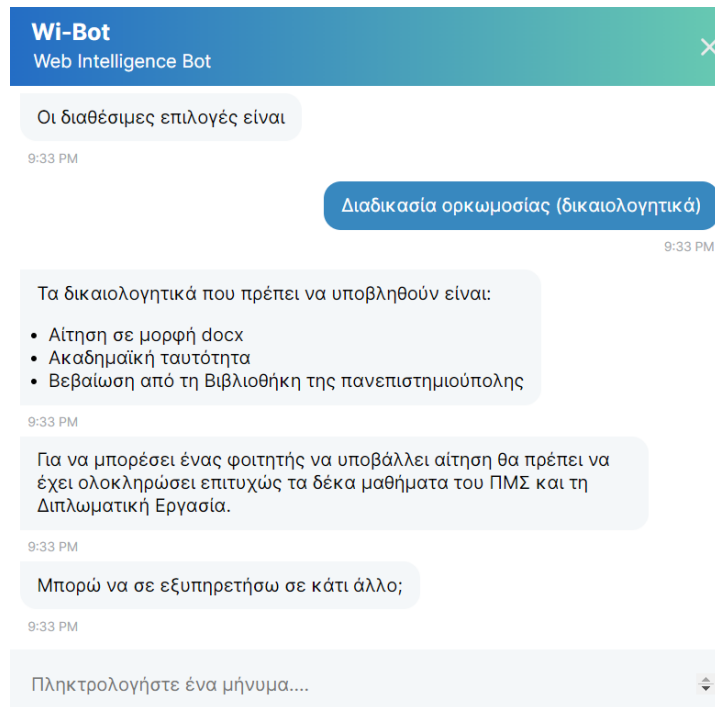
Εικόνα 4.4.32 Ερώτηση «Μαθήματα»

Στο ακόλουθο στιγμιότυπο, ο χρήστης επιθυμεί να πληροφορηθεί για τα μαθήματα του πρώτου εξαμήνου.



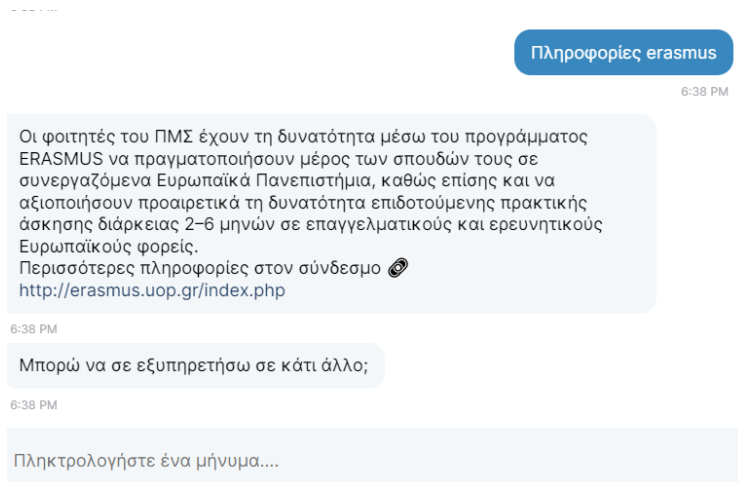
Εικόνα 4.4.33 Απάντηση WiBot για το 1^ο εξάμηνο

Στην εικόνα 4.4.34, ο χρήστης ρωτά το chatbot για να μάθει πληροφορίες για την διαδικασία της ορκωμοσίας και ειδικότερα για τα απαραίτητα δικαιολογητικά που πρέπει να υποβάλει. Το chatbot επιστρέφει μια λίστα με τα απαραίτητα έγγραφα.



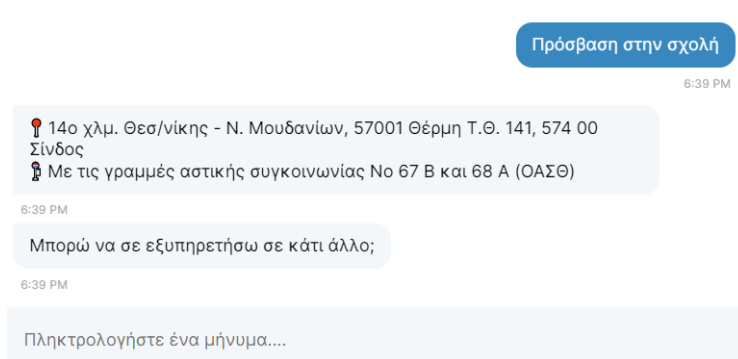
Εικόνα 4.4.34 Διάλογος «Διαδικασία ορκωμοσίας»

Η παρακάτω εικόνα, περιέχει ένα δείγμα της ιστορίας Erasmus. Εδώ ο χρήστης ζητά να μάθει πληροφορίες για το πρόγραμμα ανταλλαγής φοιτητών Erasmus. Το WiBot απαντά με τις σημαντικότερες πληροφορίες και παραπέμπει τον χρήστη να μεταβεί στον επίσημο σύνδεσμο για περισσότερες λεπτομέρειες.



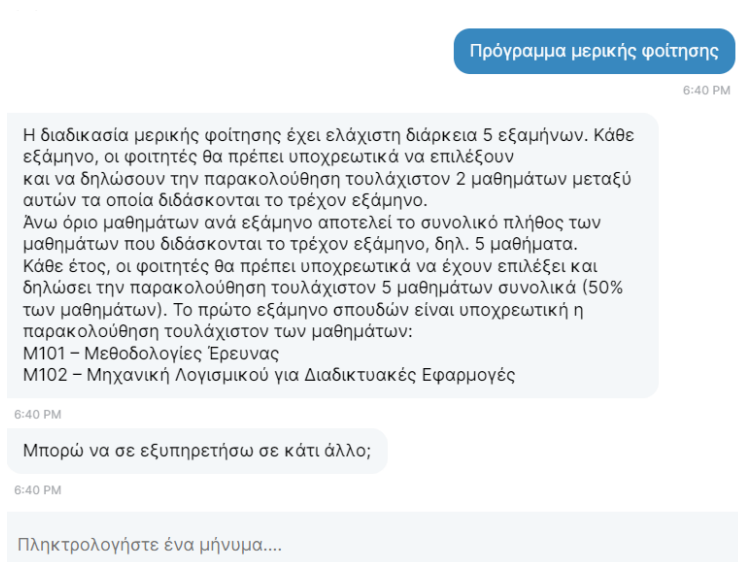
Εικόνα 4.4.35 Διάλογος «Erasmus»

Η επόμενη εικόνα, παρουσιάζει ένα διάλογο, με τον χρήστη να ρωτά πληροφορίες για τους τρόπους πρόσβασης στην σχολή. Το chatbot επιστρέφει την διεύθυνση καθώς και τις αστικές συγκοινωνίες που προσεγγίζουν την διεύθυνση αυτή.



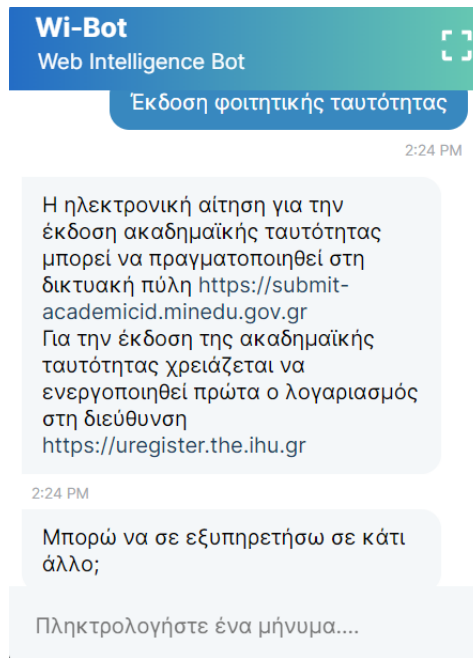
Εικόνα 4.4.36 Ερώτηση «Πρόσβαση στην σχολή»

Στην εικόνα 4.4.37, παρουσιάζεται ο διάλογος με τις πληροφορίες για το πρόγραμμα μερικής φοίτησης. Το chatbot απαντά με πληροφορίες όπως η ελάχιστη διάρκεια, επιλογή και πλήθος μαθημάτων ανά εξάμηνο και άλλα.



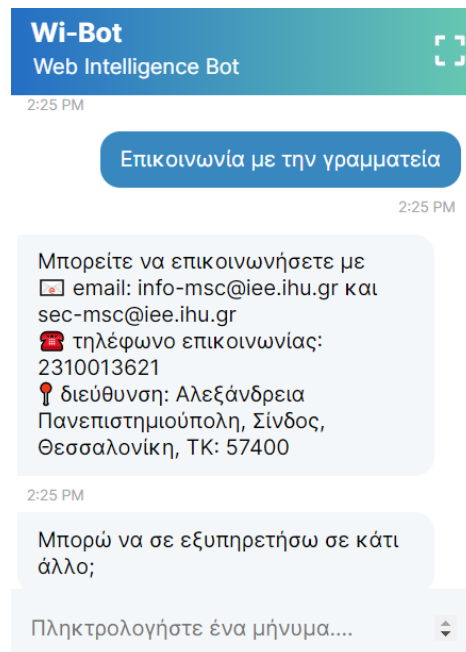
Εικόνα 4.4.37 Διάλογος «Πρόγραμμα μερικής φοίτησης»

Μια άλλη ερώτηση που μπορεί να θέσει ο χρήστης είναι πληροφορίες για την έκδοσης της φοιτητικής ταυτότητας. Ειδικότερα, το WiBot απαντά δίνοντας οδηγίες για τους ιστότοπους και την εγγραφή που απαιτείται προκειμένου να ολοκληρώσει σωστά την διαδικασία ο φοιτητής.



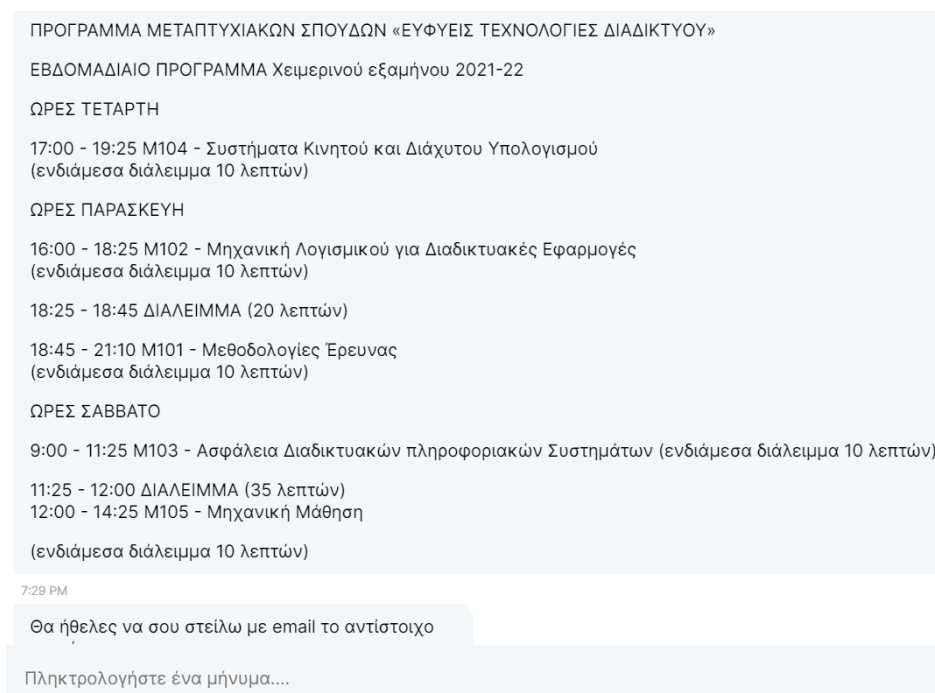
Εικόνα 4.4.38 Διάλογος «Έκδοση φοιτητικής ταυτότητας»

Επειδή το chatbot δεν μπορεί να διαθέσει όλες τις πληροφορίες στον φοιτητή, μιας και μπορεί ο χρήστης να θέσει πιο εξειδικευμένα ζητήματα, προστέθηκε μια επιλογή με τους τρόπους επικοινωνίας με την γραμματεία του Τμήματος. Δίνεται το email, το τηλέφωνο και η διεύθυνση.



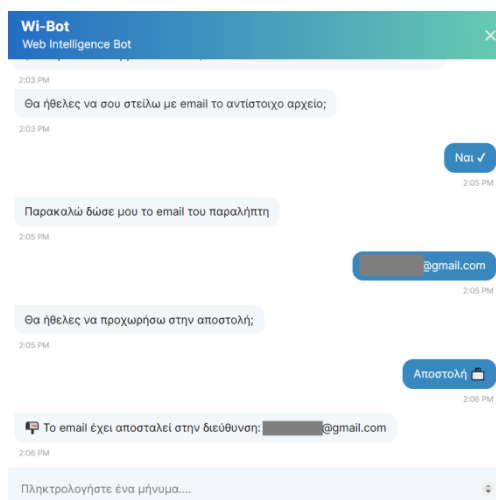
Εικόνα 4.4.39 Διάλογος «Επικοινωνία με την γραμματεία»

Επιπλέον, όπως και αναφέρθηκε, το chatbot μπορεί να “διαβάσει” αρχεία (PDF). Τα διαθέσιμα αρχεία είναι το ωρολόγιο πρόγραμμα και το ακαδημαϊκό ημερολόγιο. Συγκεκριμένα εδώ παρουσιάζεται η ιστορία με το ωρολόγιο πρόγραμμα.

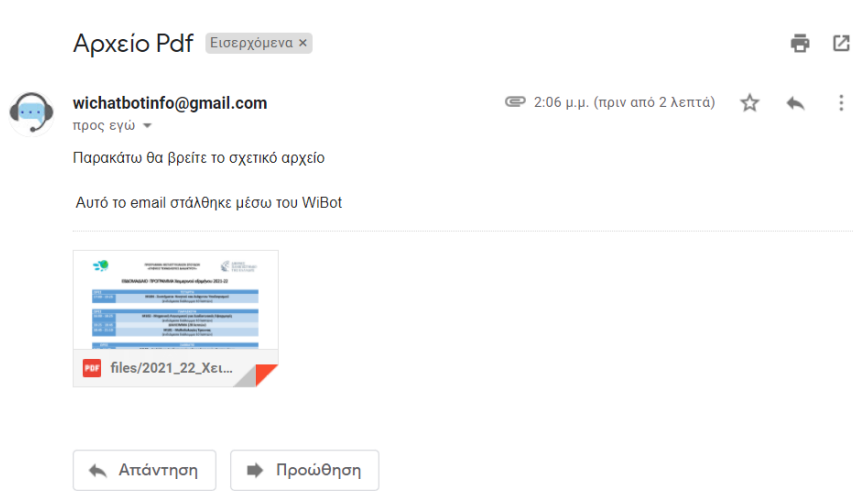


Εικόνα 4.4.40 Διάλογος «Ωρολόγιο πρόγραμμα»

Ο φοιτητής ζητάει να δει το πρόγραμμα, το WiBot επιστρέφει τις πληροφορίες και στην συνέχεια ρωτά τον χρήστη εάν επιθυμεί να του αποστείλει το αντίστοιχο αρχείο στο email του. Ακολουθεί η πληκτρολόγηση του email του και η τελική αποστολή. Στην παρακάτω εικόνα φαίνεται ο διάλογος αφού ο χρήστης επιλέξει να λάβει με email το αρχείο.

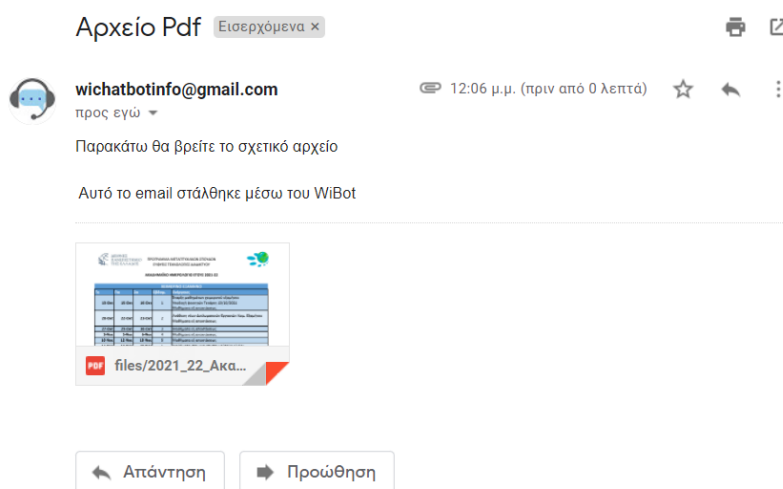


Εικόνα 4.4.41 Διάλογος για αποστολή αρχείου σε email



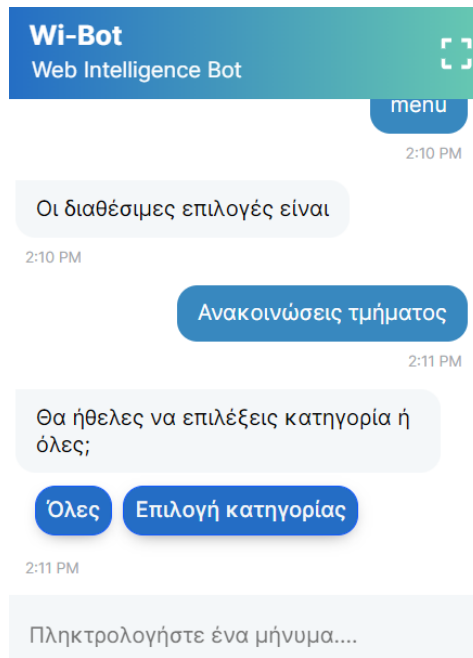
Εικόνα 4.4.42 Απεσταλμένο email (1)

Σημειώνεται ότι υπάρχει ίδια διαδικασία και για το ακαδημαϊκό ημερολόγιο.



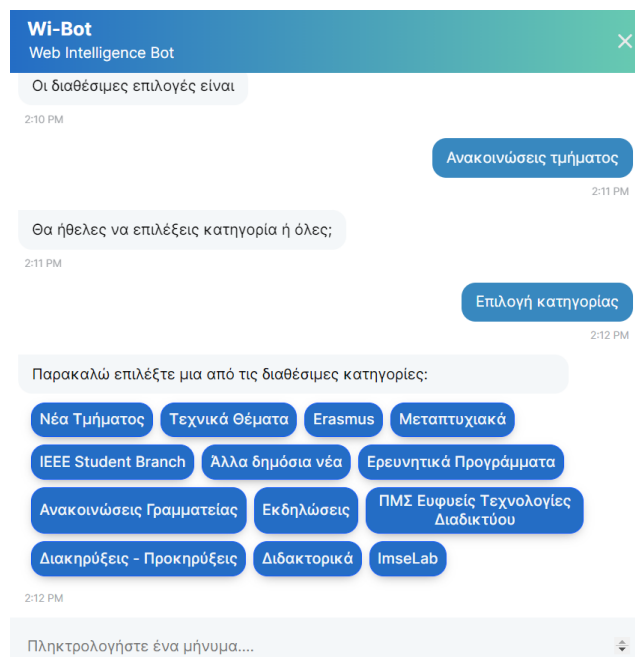
Εικόνα 4.4.43 Απεσταλμένο email (2)

Μια άλλη λειτουργία του chatbot, είναι να «διαβάξει» δυναμικά τις ανακοινώσεις. Για την ακρίβεια, συλλέγει τις ανακοινώσεις με κατάλληλες κλήσεις στο διαθέσιμο API της σχολής. Εάν λοιπόν ο χρήστης επιλέξει να ενημερωθεί για τις τελευταίες ανακοινώσεις, για αρχή το WiBot θα τον ρωτήσει εάν θέλει να επιλέξει κάποια κατηγορία (όπως *Νέα Τμήματος*) ή να του εμφανίσει από όλες τις κατηγορίες.



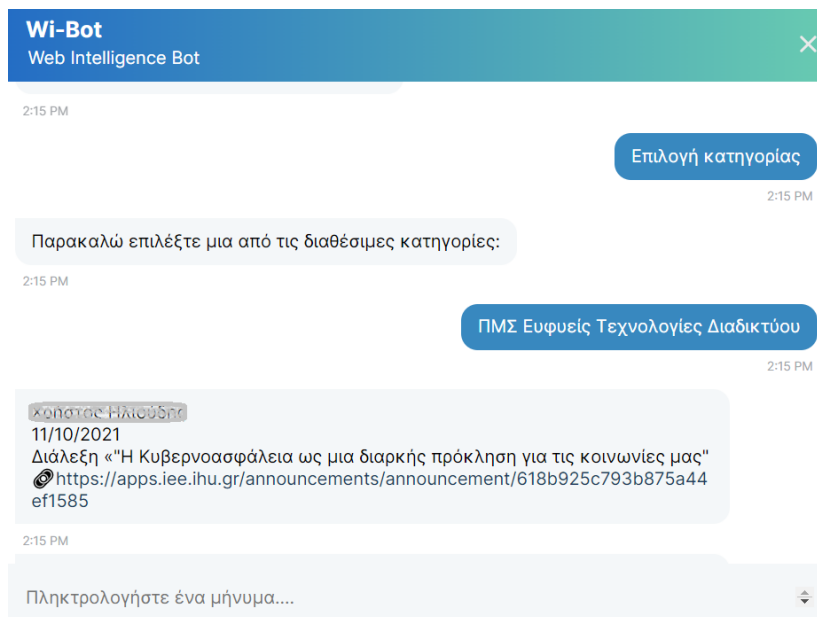
Εικόνα 4.4.44 Διάλογος «Ανακοινώσεις Τμήματος»

Στην εικόνα 4.4.45 φαίνεται η ροή του διαλόγου, εμφανίζοντας τις διαθέσιμες κατηγορίες ανακοινώσεων.



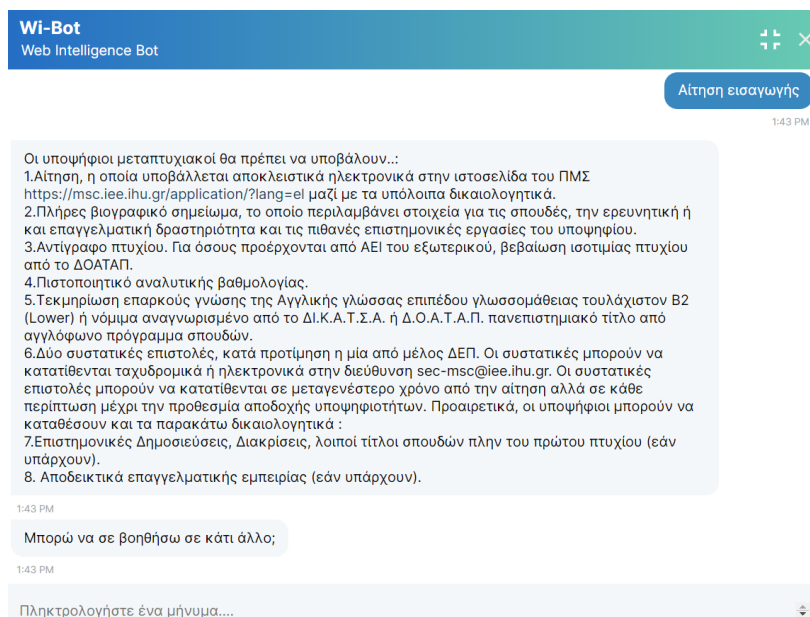
Εικόνα 4.4.45 Κατηγορίες ανακοινώσεων

Με την επιλογή μιας συγκεκριμένης κατηγορίας, εμφανίζονται οι τελευταίες (με βάση την ημερομηνία) δέκα ανακοινώσεις. Στο παράδειγμα που ακολουθεί, εμφανίζεται η πρώτη, και όπως παρατηρούμε, στο πλαίσιο παρουσιάζονται οι εξής πληροφορίες: το ονοματεπώνυμο, η ημερομηνία, ο τίτλος και το link για την προβολή της ανακοίνωσης.



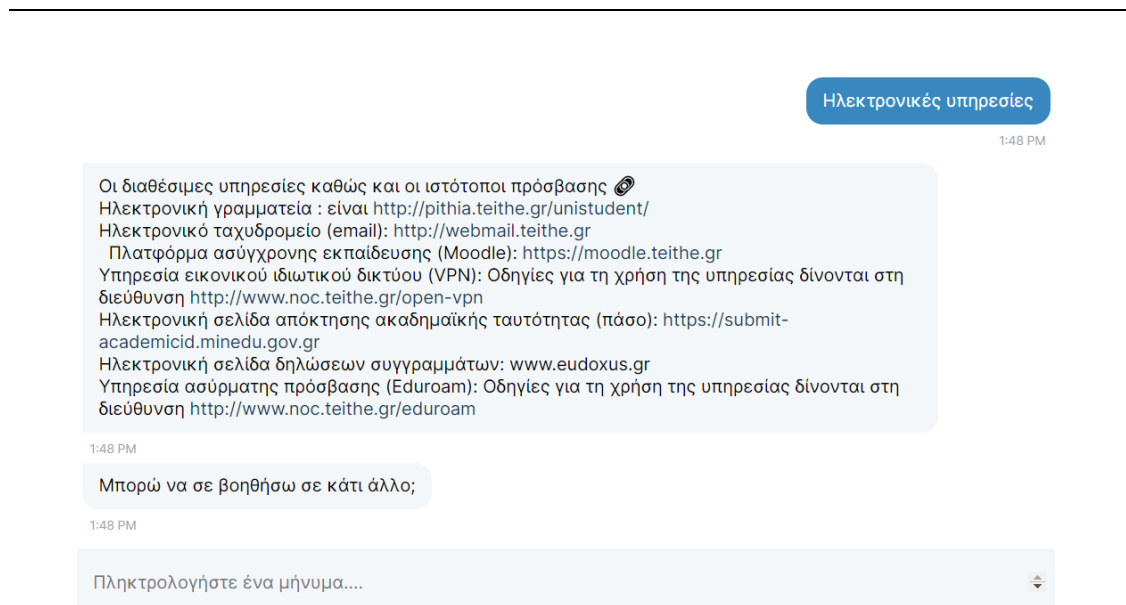
Εικόνα 4.4.46 Παράδειγμα εμφάνισης 1^{ης} Ανακοίνωσης

Σχετικά με την αίτηση εισαγωγής, το chatbot επιστρέφει τα έγγραφα που πρέπει να υποβάλλει ο υποψήφιος φοιτητής όπως ηλεκτρονική αίτηση, αντίγραφα πτυχίου, πιστοποιητικά και άλλα.



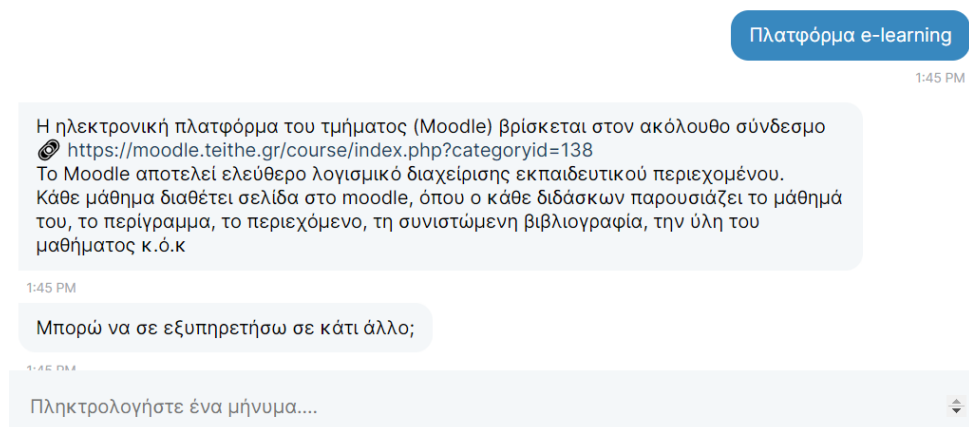
Εικόνα 4.4.47 Διάλογος «Αίτηση εισαγωγής»

Μία άλλη ερώτηση αφορά τις διαθέσιμες ηλεκτρονικές υπηρεσίες του Τμήματος. Συγκεκριμένα το chatbot επιστρέφει συγκεντρωτικά τους συνδέσμους που χρειάζεται ο φοιτητής, μετά την απόκτηση των προσωπικών του κωδικών, όπως το ηλεκτρονικό ταχυδρομείο, το VPN, τον ιστότοπο δήλωσης συγγραμμάτων κ.α.



Εικόνα 4.4.48 Διάλογος «Ηλεκτρονικές υπηρεσίες»

Ακόμη, το chatbot ενημερώνει τον χρήστη για την πλατφόρμα e-learning (Moodle) που χρησιμοποιείται από το Τμήμα.



Εικόνα 4.4.49 Διάλογος «Πλατφόρμα e-learning»

Οι υποψήφιοι φοιτητές κατά την αναζήτηση τους, συνήθως ψάχνουν πληροφορίες για το Τμήμα αλλά και την σχολή. Γι' αυτό το chatbot προσφέρει γενικές πληροφορίες για τα παραπάνω, όπως φαίνεται και στο ακόλουθο στιγμιότυπο.

Το Διεθνές Πανεπιστήμιο της Ελλάδος (ΔΙ.ΠΑ.Ε.) ιδρύθηκε με το νόμο 3391 του 2005 και άρχισε να λειτουργεί το 2008 όταν έλαβε χρηματοδότηση από το ΕΣΠΑ 2007-2013 μέσω του επιχειρησιακού προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και εδρεύει στη Θέρμη του νομού Θεσσαλονίκης. Στο Διεθνές Πανεπιστήμιο της Ελλάδας λειτουργούν 34 Τμήματα τα οποία υπάγονται σε επτά (7) Σχολές. Για συγκεκριμένα:

1:50 PM

Σχολή Οικονομίας και Διοίκησης, με έδρα τη Θεσσαλονίκη
 Σχολή Κοινωνικών Επιστημών, με έδρα τη Θεσσαλονίκη
 Σχολή Επιστημών Υγείας, με έδρα τη Θεσσαλονίκη
 Σχολή Μηχανικών, με έδρα τις Σέρρες
 Σχολή Επιστημών Σχεδιασμού, με έδρα τις Σέρρες
 Σχολή Θετικών Επιστημών, με έδρα την Καβάλα
 Σχολή Γεωτεχνικών Επιστημών, με έδρα τη Δράμα
 Περισσότερες πληροφορίες θα βρείτε, στην επίσημη ιστοσελίδα <https://www.ihu.gr/>
 Όσο αφορά το Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ) Ευφυείς Τεχνολογίες Διαδικτύου, ανήκει στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων (του Διεθνούς Πανεπιστημίου της Ελλάδος) ιδρύθηκε και λειτουργεί από το Ακαδημαϊκό εξάμηνο 2012-2013 και εγκρίθηκε πρόσφατα η συνέχιση λειτουργίας του από το Υπουργείο Παιδείας (ΦΕΚ 2121/21-5-2021). Οδηγεί σε Μεταπτυχιακό Δίπλωμα Ειδικότητας (Μ.Δ.Ε.) επιπέδου εκπαίδευσης 7, στο Εθνικό Πλαίσιο Προσόντων.
 Περισσότερες πληροφορίες θα βρείτε, στην επίσημη ιστοσελίδα <https://msc.iee.ihu.gr/>

1:50 PM

Πληκτρολογήστε ένα μήνυμα....

Εικόνα 4.4.50 Διάλογος «Πληροφορίες για το Τμήμα και την σχολή»

Όπως αναφέρθηκε παραπάνω, υπάρχει η δυνατότητα αναζήτησης των στοιχείων επικοινωνίας ενός καθηγητή ή η επιστροφή των στοιχείων από όλους (μέσω του API). Ωστόσο, θεωρήσαμε απαραίτητο για την ευκολία του χρήστη να επιστρέφουμε και την επίσημη λίστα (μέσω του site του Τμήματος). Εκεί ο χρήστης μπορεί να βρει το όνομα και το email επικοινωνίας του καθηγητή, αλλά και την σελίδα του στο people.iee.ihu.gr, όπου εκεί παρουσιάζεται το προφίλ του με το βιογραφικό του σημείωμα τρόπους επικοινωνίας, ώρες γραφείου κ.α.

Wi-Bot
Web Intelligence Bot

2:00 PM

Τρόποι επικοινωνίας διδάσκοντες
(Λίστα τμήματος)

2:02 PM

Περισσότερες πληροφορίες στον σύνδεσμο <https://msc.iee.ihu.gr/faculty/?lang=el>

2:02 PM

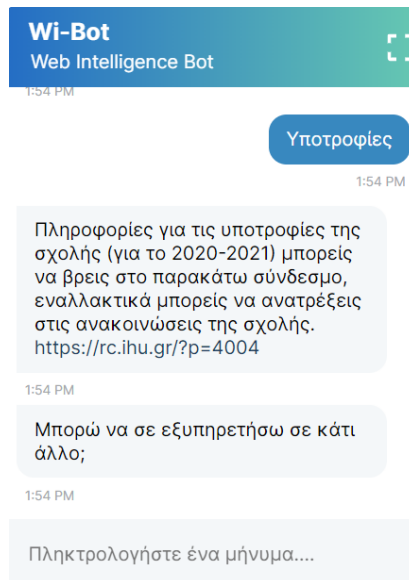
Μπορώ να σε εξυπηρετήσω σε κάτι άλλο;

2:02 PM

Πληκτρολογήστε ένα μήνυμα....

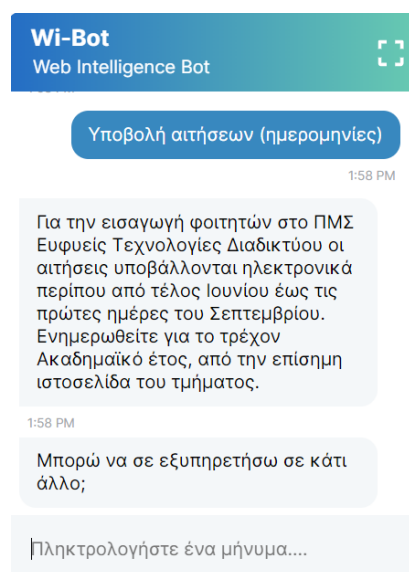
Εικόνα 4.4.51 Διάλογος «Τρόποι επικοινωνίας διδάσκοντες (Λίστα Τμήματος)»

Ένα άλλο αίτημα που δεχθήκαμε κατά την διαδικασία του testing, ήταν σχετικά με πληροφορίες για τις υποτροφίες. Το συγκεκριμένο θέμα διαφοροποιείται κάθε ακαδημαϊκό έτος, και δεν υπάρχει συγκεκριμένη ενότητα που να δίνει περισσότερες πληροφορίες. Οπότε η απάντηση που τελικά δίνει το chatbot είναι ένα παράδειγμα πρόσκλησης υποβολής στην οποία ο φοιτητής μπορεί να δει τι απαιτείται για να έχει μια γενικότερη εικόνα.



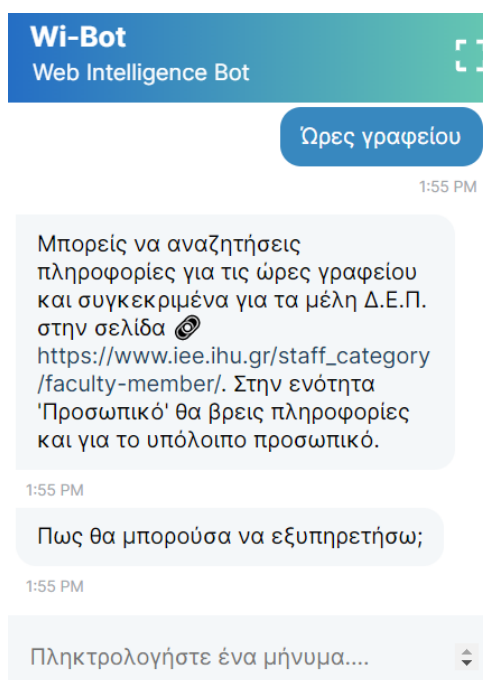
Εικόνα 4.4.52 Διάλογος «Υποτροφίες»

Παρακάτω ακολουθεί ένα στιγμιότυπο από την “Υποβολή αιτήσεων” και συγκεκριμένα αφορά τις ημερομηνίες υποβολής. Αυτό το ερώτημα δεν μπορεί να απαντηθεί απόλυτα, μιας και σε κάθε ακαδημαϊκό έτος οι ημερομηνίες αλλάζουν. Παρ’ όλα αυτά το chatbot πληροφορεί τον χρήστη δίνοντας ένα γενικότερο εύρος που συνηθίζεται να υποβάλλονται οι σχετικές αιτήσεις.



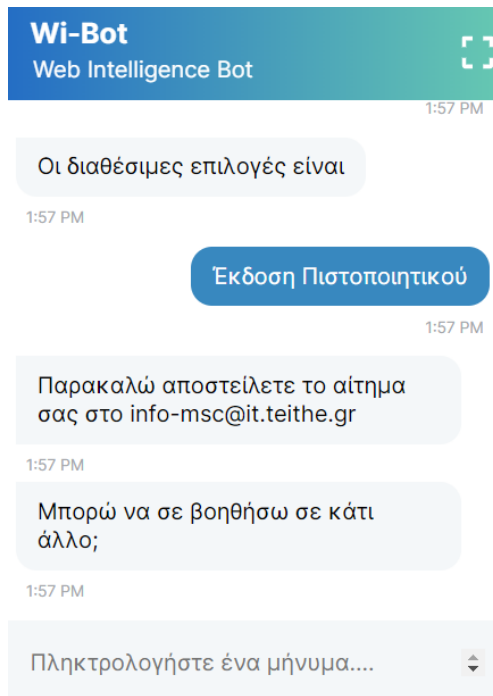
Εικόνα 4.4.53 Διάλογος «Υποβολή αιτήσεων»

Παρακάτω ακολουθεί η ιστορία με τις ώρες γραφείου των καθηγητών. Αυτή η πληροφορία σε πρώτο επίπεδο αναζητήθηκε μέσα από τα στοιχεία που δίνει το API του Τμήματος. Επειδή όμως δεν υπάρχει καταγεγραμμένη σε όλους τους καθηγητές, θεωρήθηκε ως καλύτερη λύση να δίνουμε τον σύνδεσμο που υπάρχει στον επίσημο ιστότοπο του πανεπιστημίου. Ο σύνδεσμος περιέχει τις ώρες γραφείου από τα μέλη ΔΕΠ, αλλά αντίστοιχα μπορεί κάνεις να βρει και για τα Μέλη Δ.Ε.Π. άλλων τμημάτων, Προσωπικό Ε.ΔΙ.Π. κ.α.



Εικόνα 4.4.54 Διάλογος «Ωρες γραφείου»

Μια άλλη ιδέα που είχαμε ήταν να μπορεί το chatbot να εκδώσει πιστοποιητικά, όπως για την βεβαίωση παρακολούθησης του προγράμματος. Στην πορεία όμως δεν βρήκαμε κάποια εύκολη λύση, και δεν μπορέσαμε να συλλέξουμε τα στοιχεία του φοιτητή από το σύστημα ΡΙΤΗΙΑ, για την διασταύρωση. Τελικά το chatbot δίνει απάντηση σχετικά με την έκδοση πιστοποιητικών αλλά ουσιαστικά παραπέμπει τον χρήστη στην Γραμματεία.



Εικόνα 4.4.55 Διάλογος «Έκδοση Πιστοποιητικού»

5

Αξιολόγηση

Στο κεφάλαιο αυτό αναλύεται η διαδικασία της αξιολόγησης (testing), με τα acceptance test που έτρεξαν στο development environment, και την πειραματική αξιολόγηση από τους χρήστες.

5.1 Acceptance testing

Για να υπάρξει τελικά εμπιστοσύνη στο πώς θα ενεργήσει το chatbot σε ορισμένες περιπτώσεις ή σενάρια, το Rasa προτείνει ως καλύτερο τρόπο την δημιουργία test ιστοριών (test stories ή δοκιμαστικές ιστορίες). Οι δοκιμαστικές ιστορίες είναι σαν τις ιστορίες χρήστη (αρχείο stories), αλλά περιλαμβάνουν επιπλέον πιθανά μηνύματα του χρήστη. Τρέχουν στο εκπαιδευμένο μοντέλο και ο χρήστης αποκτά μια εικόνα στο πως ανταποκρίνεται τελικά το chatbot. Για να υπάρχουν καλύτερα αποτελέσματα, οι δοκιμαστικές ιστορίες θα πρέπει να είναι αντιπροσωπευτικές ή τουλάχιστον να πλησιάζουν, όσο γίνεται τις πραγματικές.

Στην συγκεκριμένη περίπτωση, δημιουργήθηκαν δώδεκα δοκιμαστικές ιστορίες, στον φάκελο test και ειδικότερα στο αρχείο test_stories.yml. Στην εικόνα 5.1.1 εμφανίζεται ένα παράδειγμα δοκιμαστικής ιστορίας για πληροφορίες σχετικά με το Erasmus.

```
stories:
- story: erasmus
  steps:
  - user: |
      γεια
      intent: greet
      action: utter_help
  - user: |
      θέλω να μου δώσεις πληροφορίες για το erasmus
      intent: erasmus_info
      action: utter_erasmus_info
  - user: |
      ευχαριστώ
      intent: thanks
      action: utter_help_new
  - user: |
      όχι
      intent: deny
      action: utter_bye
```

Εικόνα 5.1.1 test_stories.yml

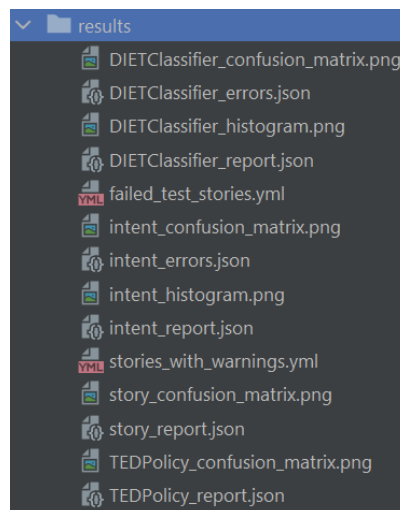
Η δοκιμή γίνεται με την παρακάτω εντολή:

- `rasa test`

Επιπλέον, για να αξιολογήσει κάποιος τα μοντέλα διαλόγου και NLU ξεχωριστά, μπορεί να χρησιμοποιήσει την παρακάτω εντολή:

- `rasa test core --model <path-to-model.tar.gz>`

Από προεπιλογή, η εντολή Rasa test, θα εκτελέσει δοκιμές σε ιστορίες από οποιαδήποτε αρχεία με ονόματα που ξεκινούν με `test_`. Τα αποτελέσματα καταγράφονται σε ένα νέο φάκελο “results”.



Εικόνα 5.1.2 Φάκελος results

Το Rasa test παράγει ορισμένα αρχεία σχετικά με το intent classification του μοντέλου:

intent_report.json μια αναφορά (report), που καταγράφει την ακρίβεια, ανάκληση και f1-score για κάθε intent, και παρέχει έναν συνολικό μέσο όρο.

intent_confusion_matrix.png, ένα πίνακα σύγχυσης σε μορφή εικόνας png, δείχνει ποιες προθέσεις μπερδεύονται με άλλες.

intent_histogram.png ένα ιστόγραμμα με βάση την εμπιστοσύνη (confidence)

errors.json καταγράφονται και αποθηκεύονται δείγματα που έχουν προβλεφθεί λανθασμένα, για ευκολότερο εντοπισμό σφαλμάτων.[21]

Να σημειωθεί ότι μόνο οι εκπαιδευσιμοι εξαγωγείς οντοτήτων (entity extractors), όπως ο DIETClassifier και ο CRFEntityExtractor αξιολογούνται με την εντολή Rasa test. Ενώ, οι προ-εκπαιδευμένοι εξαγωγείς όπως το DucklingHTTPExtractor δεν αξιολογούνται.[21]

Άλλα αρχεία που δημιουργούνται:

DIETClassifier_errors.json, σφάλματα ταξινόμησης οντοτήτων

DIETClassifier_report.json, εξαγωγή οντοτήτων f1, ακρίβεια, σύνοψη ανάκλησης

intent_errors.json, λεπτομέρειες σφάλματος πρόθεσης

Στις παρακάτω εικόνες, εμφανίζονται αρχεία json με λάθη, όπως το αρχείο Intent_errors.json και το DIETClassifier_errors.json.

```
[
  {
    "text": "3432",
    "intent": "my_edu_code",
    "intent_prediction": {
      "name": "semester",
      "confidence": 0.9462183713912964
    }
  }
]
```

Εικόνα 5.1.3 Intent_errors.json

```
[
  {
    "text": "3432",
    "entities": [
      {
        "start": 0,
        "end": 5,
        "value": "3432",
        "entity": "am"
      }
    ],
    "predicted_entities": [
      {
        "entity": "semester",
        "start": 0,
        "end": 5,
        "confidence_entity": 0.7952980995178223,
        "value": "3432",
        "extractor": "DIETClassifier"
      }
    ]
  },
  {
    "text": "το αμ 3432",
    "entities": [
      {
        "start": 6,
        "end": 11,
        "value": "3432",
        "entity": "am"
      }
    ],
    "predicted_entities": [
      {
        "entity": "semester",
        "start": 6,
        "end": 11,
        "confidence_entity": 0.682672381401062,
        "value": "3432",
        "extractor": "DIETClassifier"
      }
    ]
  }
],
{
```

Εικόνα 5.1.4 DIETClassifier_errors.json

Στα αποτελέσματα παρατηρήθηκαν προβλήματα που αφορούν κατά βάση τις ιστορίες και τα intents από το πιλοτικό project. Γι' αυτό, σχολιάστηκαν όλα όσα δεν αποτελούν πληροφορία του Main project και δημιουργήσαμε πάλι νέο μοντέλο.

Στο αρχείο *story_report* καταγράφονται ορισμένες μετρικές, η ακρίβεια (precision), η ανάκληση (recall), το F-measure και η υποστήριξη (support). Οι τιμές αυτές αποτελούν μέτρα αξιολόγησης της αποτελεσματικότητας του μοντέλου. Σύμφωνα με το scikit-learn [22] δίνονται οι βασικοί ορισμοί.

Ακρίβεια είναι ο λόγος $tp / (tp + fp)$ όπου tp είναι ο αριθμός των αληθινών θετικών (true positives) και fp ο αριθμός των ψευδώς θετικών (false positives). Η ακρίβεια είναι η ικανότητα του ταξινομητή να μην επισημαίνει ως θετικό ένα δείγμα που είναι αρνητικό.

Ανάκληση είναι ο λόγος $tp / (tp + fn)$ όπου tp είναι ο αριθμός των true positives και fn ο αριθμός των false negatives. Η ανάκληση αποτελεί την ικανότητα του ταξινομητή να βρίσκει όλα τα θετικά δείγματα.

Η βαθμολογία *F1 score* μπορεί να ερμηνευθεί ως το αρμονικό μέσο (harmonic mean) της ακρίβειας και της ανάκλησης, όπου η καλύτερη τιμή είναι το 1 και η χειρότερη το 0.

Η *υποστήριξη* είναι ο αριθμός των εμφανίσεων κάθε κλάσης στο y_true .

```
"accuracy": 0.9834710743801653,
"macro avg": {
  "precision": 0.9704134366925063,
  "recall": 0.9728682170542634,
  "f1-score": 0.9710464607250602,
  "support": 121
},
"weighted avg": {
  "precision": 0.9774563820018365,
  "recall": 0.9834710743801653,
  "f1-score": 0.9795999339132406,
  "support": 121
},
"conversation_accuracy": {
  "accuracy": 0.9487179487179487,
  "correct": 37,
  "with_warnings": 0,
  "total": 39
}
```

Εικόνα 5.1.5 story_report.json

Ο *macro-average* θα υπολογίσει το μέτρο ανεξάρτητα για κάθε τάξη και στη συνέχεια θα λάβει τον μέσο όρο.

Η *macro* εκτελεί υπολογισμούς για κάθε κλάση και βρίσκει τους μη σταθμισμένους μέσους. Όμως δεν λαμβάνει υπόψη πιθανές ανισορροπίες στα πεδία.

Ο *weighted* υπολογίζει τη μετρική για κάθε κλάση και βρίσκει τον μέσο, με βάση την υποστήριξη (support).

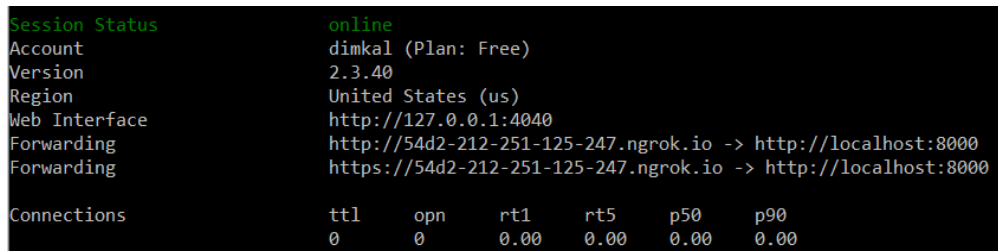
Συμπερασματικά, η αξιολόγηση δίνει πολλά δεδομένα και αποτελεί ένα μεγάλο κομμάτι ανάλυσης. Στην συγκεκριμένη περίπτωση έγινε μια προσπάθεια να αξιοποιήσουμε όσες περισσότερες πληροφορίες γίνεται ώστε να γίνει καλύτερο το μοντέλο μας και φυσικά το chatbot. Σίγουρα θα μπορούσαμε μελλοντικά να παρατηρήσουμε και να αναλύσουμε τις τιμές με σκοπό να κάνουμε καλύτερο το μοντέλο μας.

5.2 Πειραματική αξιολόγηση (χρήστες)

Για τις ανάγκες της αξιολόγησης, αναζητήσαμε μια λύση ώστε να κάνουμε διαθέσιμη δημοσίως την εφαρμογή. Μια εύκολη λύση η οποία αναφερόταν αρκετά συχνά σε αναζητήσεις αλλά και στο φόρουμ του Rasa, είναι το ngrok. Με το ngrok μπορεί κάποιος να κάνει δημοσίως διαθέσιμο στο Διαδίκτυο έναν διακομιστή web που εκτελείται σε τοπικό υπολογιστή (localhost).

Αρχικά, πρέπει να γίνει η εγκατάσταση, να δημιουργηθεί λογαριασμός στον επίσημο ιστότοπο (<https://ngrok.com>) του ngrok, ώστε τελικά να εμφανιστεί ένα authToken. Έπειτα, πρέπει να οριστεί σε ποια θύρα - port ακούει ο διακομιστής. Για παράδειγμα, με την παρακάτω εντολή έχουμε επιλέξει την θύρα 8000:

➤ `ngrok http 8000`



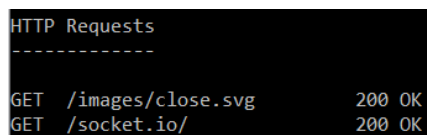
```
Session Status      online
Account            dimkal (Plan: Free)
Version            2.3.40
Region             United States (us)
Web Interface      http://127.0.0.1:4040
Forwarding         http://54d2-212-251-125-247.ngrok.io -> http://localhost:8000
Forwarding         https://54d2-212-251-125-247.ngrok.io -> http://localhost:8000

Connections
  ttl    opn    rt1    rt5    p50    p90
   0     0     0.00  0.00  0.00  0.00
```

Εικόνα 5.2.1 Εκτέλεση εντολής

Όπως φαίνεται και στην εικόνα το localhost με το port 8000, αντιστοιχεί με τη δημόσια διεύθυνση URL <https://xxxxx.ngrok.io>, δίνοντας και το αντίστοιχο url για το http.

Ακόμη, μπορούμε να δούμε τα requests που εκτελούνται.



```
HTTP Requests
-----
GET /images/close.svg      200 OK
GET /socket.io/           200 OK
```

Εικόνα 5.2.2 Requests

Αφότου, έγιναν οι παραπάνω ενέργειες, δοκιμάστηκε σε δύο συστήματα, στον υπολογιστή όπου έγινε η υλοποίηση και σε έναν ακόμη. Με τον επιτυχημένο διαμοιρασμό και αφού έγινε ένας έλεγχος στην ροή των συζητήσεων με το chatbot, ξεκίνησε η διαδικασία του testing (της αξιολόγησης).

5.2.1 Α' Φάση Αξιολόγησης

Σε αυτό το σημείο, θεωρήθηκε αρκετά σημαντικό κομμάτι, η αποτύπωση της γνώμης και της εμπειρίας των χρηστών. Γι' αυτό, δημιουργήθηκε ένα ερωτηματολόγιο μέσω των Google forms. Οι Φόρμες Google (Google Forms) επιτρέπουν την δημιουργία και την επεξεργασία φορμών. Διαθέτουν διαφορετικούς τύπους απαντήσεων όπως πολλαπλής επιλογής, σύντομη

απάντηση, ελεύθερο κείμενο και άλλα. Αποτελεί μια εύκολη και γρήγορη λύση για την δημιουργία ερωτηματολογίου που μάλιστα παράγει γραφήματα από τις απαντήσεις, οπτικοποιώντας τα αποτελέσματα.

Οι ερωτήσεις που δημιουργήθηκαν, μαζί με τον τύπο των απαντήσεων, παρουσιάζονται στον παρακάτω πίνακα.

Πίνακας 5.2.1 Ερωτήσεις αξιολόγησης

Ερώτηση	Τύπος Απάντησης
Πόσο εξοικειωμένοι είστε με τα chatbots; (1 δεν έχω χρησιμοποιήσει ποτέ, 5 χρησιμοποιώ πολύ συχνά)	Επιλογή 1 έως 5
Πως αξιολογείτε την εμπειρία σας συγκεκριμένα με το WiBot; (1 καθόλου καλή, 5 άριστη)	Επιλογή 1 έως 5
Το WiBot κάλυψε τις ανάγκες σας; (1 καθόλου, 5 άριστα)	Επιλογή 1 έως 5
Το WiBot ήταν εύκολο στην χρήση; (1 καθόλου, 5 άριστα)	Επιλογή 1 έως 5
Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος;	Ναι, Όχι
Πως αξιολογείτε την χρησιμότητα του WiBot; (1 καθόλου χρήσιμο, 5 άριστα)	Επιλογή 1 έως 5
Ποιες άλλες πληροφορίες θα μπορούσε να δώσει;	Ελεύθερο κείμενο
Αφήστε κάποια σχόλια	Ελεύθερο κείμενο

Αφού λοιπόν κοινοποιήθηκε η εφαρμογή, δόθηκε στους χρήστες και ο σύνδεσμος για το αντίστοιχο ερωτηματολόγιο. Αρχικά, ζητήθηκε από τους χρήστες να πειραματιστούν με την εφαρμογή, στην συνέχεια να απαντήσουν στις ερωτήσεις και επιπλέον να αφήσουν κάποια σχόλια ή προτάσεις βελτίωσης.

<p>Wi-Bot eval</p> <p>Αξιολόγηση chatbot Wi-Bot</p>	<p>Το Wi-Bot ήταν εύκολο στην χρήση; (1 καθόλου, 5 άριστα) *</p> <p><input type="radio"/> 1</p> <p><input type="radio"/> 2</p> <p><input type="radio"/> 3</p> <p><input type="radio"/> 4</p> <p><input type="radio"/> 5</p>
<p>Πόσο εξοικειωμένοι είστε με τα chatbots; (1 δεν έχω χρησιμοποιήσει ποτέ, 5 χρησιμοποίησα πολύ συχνά) *</p> <p><input type="radio"/> 1</p> <p><input type="radio"/> 2</p> <p><input type="radio"/> 3</p> <p><input type="radio"/> 4</p> <p><input type="radio"/> 5</p>	<p>Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος; *</p> <p><input type="radio"/> Ναι</p> <p><input type="radio"/> Όχι</p>
<p>Πως αξιολογείτε την εμπειρία σας συγκεκριμένα με το Wi-Bot; (1 καθόλου καλή, 5 άριστη) *</p> <p><input type="radio"/> 1</p> <p><input type="radio"/> 2</p> <p><input type="radio"/> 3</p> <p><input type="radio"/> 4</p> <p><input type="radio"/> 5</p>	<p>Πως αξιολογείτε την χρησιμότητα του Wi-Bot; (1 καθόλου χρήσιμο, 5 άριστα) *</p> <p><input type="radio"/> 1</p> <p><input type="radio"/> 2</p> <p><input type="radio"/> 3</p> <p><input type="radio"/> 4</p> <p><input type="radio"/> 5</p>
<p>Το Wi-Bot κάλυψε τις ανάγκες σας; (1 καθόλου, 5 άριστα) *</p> <p><input type="radio"/> 1</p> <p><input type="radio"/> 2</p> <p><input type="radio"/> 3</p> <p><input type="radio"/> 4</p> <p><input type="radio"/> 5</p>	<p>Ποιες άλλες πληροφορίες θα μπορούσε να δώσει:</p> <p>Κείμενο μακροσκελούς απάντησης</p> <hr/> <p>Αφήστε κάποια σχόλια</p> <p>Κείμενο μακροσκελούς απάντησης</p> <hr/>

Εικόνα 5.2.3 Φόρμα Αξιολόγησης (Α')

5.2.1.1 Αποτελέσματα

Οι συμμετέχοντες στην διαδικασία της αξιολόγησης ήταν συνολικά πέντε, και απάντησαν όλοι στο ερωτηματολόγιο. Τα αποτελέσματα της αξιολόγησης επισυνάπτονται αναλυτικά στο Παράρτημα Β.

Κάποιες γενικότερες παρατηρήσεις αφορούν τα εξής:

Αρχικά το μεγαλύτερο ποσοστό (60%) δηλώσαν ότι είναι εξοικειωμένοι με τα chatbots. Οι χρήστες αξιολόγησαν θετικά την εμπειρία συνομιλίας με το WiBot, αφού το 60% ψήφισε την βαθμίδα 3 (μέτριο), το 20% την βαθμίδα 4 (πολύ καλή) και το 20% την βαθμίδα 5 (άριστη). Θεώρησαν ότι το WiBot κάλυψε τις ανάγκες τους με το 60% να ψήφισε την βαθμίδα 4 (πολύ καλή). Όσο αφορά την ευκολία στην χρήση, το 40% δήλωσε πολύ εύκολο (βαθμίδα 5) και επίσης το υπόλοιπο 40% δήλωσε αρκετά εύκολο (βαθμίδα 4). Ακόμη, το 60% δήλωσε ότι το chatbot απάντησε σε κάποιο σημείο λάθος, ενώ το 40% δεν αντιμετώπισε κάποιο αντίστοιχο θέμα. Σχετικά με την χρησιμότητα του WiBot, το 60% θεωρεί πολύ χρήσιμο και 40% αρκετά χρήσιμο.

5.2.2 Β' Φάση Αξιολόγησης

Επιπλέον, αφότου έγιναν οι διορθώσεις και οι τελευταίες προσθήκες, δημιουργήθηκε άλλο ένα ερωτηματολόγιο, ίδιο με το αρχικό με σκοπό να συλλέξουμε την γνώμη των ίδιων χρηστών αλλά αυτή τη φορά στην τελική έκδοση.

Wi-Bot eval v2
Αξιολόγηση chatbot Wi-Bot v2

Πόσο εξοικειωμένοι είστε με τα chatbots; (1 Δεν έχω χρησιμοποιήσει ποτέ, 5 χρησιμοποιώ πολύ συχνά) *

1
 2
 3
 4
 5

Πώς αξιολογείτε την εμπειρία σας συγκεκριμένα με το Wi-Bot; (1 καθόλου καλή, 5 άριστη) *

1
 2
 3
 4
 5

Το Wi-Bot κάλυψε τις ανάγκες σας; (1 καθόλου, 5 άριστα) *

1
 2
 3
 4
 5

Το Wi-Bot ήταν εύκολο στην χρήση; (1 καθόλου, 5 άριστα) *

1
 2
 3
 4
 5

Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος; *

Ναι
 Όχι

Πως αξιολογείτε την χρησιμότητα του Wi-Bot; (1 καθόλου χρήσιμο, 5 άριστα) *

1
 2
 3
 4
 5

Ποιες άλλες πληροφορίες θα μπορούσε να δώσει:
Κείμενο μακροσκελούς απάντησης

Αφήστε κάποια σκόλια
Κείμενο μακροσκελούς απάντησης

Εικόνα 5.2.4 Φόρμα Αξιολόγησης (Β')

5.2.2.1 Αποτελέσματα

Οι συμμετέχοντες στην διαδικασία της δεύτερης αξιολόγησης ήταν πέντε, και απάντησαν όλοι στο ερωτηματολόγιο. Επιλέχθηκαν οι ίδιοι που συμμετείχαν στην προηγούμενη φάση, διότι θέλαμε την άποψη, από εκείνους που είχαν πειραματιστεί στην προηγούμενη έκδοση. Τα αποτελέσματα της αξιολόγησης επισυνάπτονται αναλυτικά στο Παράρτημα Γ.

Οι παρατηρήσεις που συγκεντρώσαμε αναλύοντας τα αποτελέσματα:

Αρχικά το μεγαλύτερο ποσοστό (60%) δηλώσαν ότι είναι εξοικειωμένοι με τα chatbots (ίδιοι οι χρήστες, χωρίς νέα εμπειρία σε αυτό το διάστημα). Οι χρήστες αξιολόγησαν θετικά την εμπειρία συνομιλίας με το WiBot, αφού το 60% ψήφισε την βαθμίδα 5 (άριστη) και το 40% την βαθμίδα 4 (πολύ καλή). Θεώρησαν ότι το WiBot κάλυψε τις ανάγκες τους αφού το 80% ψήφισε την βαθμίδα 5 (άριστα). Όσο αφορά την ευκολία στην χρήση, όλοι οι χρήστες το αξιολόγησαν ως εύχρηστο, βαθμίδα 5. Αυτή την φορά δεν παρατηρήθηκε κάποια λάθος απάντηση από το chatbot. Σχετικά με την χρησιμότητα του WiBot, το 40% θεωρεί αρκετά χρήσιμο και το 60% πολύ χρήσιμο.

Όλες οι παραπάνω τιμές δηλώνουν ότι το chatbot, όντως βελτιώθηκε στην τελική του έκδοση, παρουσιάζει αρκετά καλύτερες κριτικές στην ευχρηστία του, στην χρησιμότητά του και δεν παρατηρήθηκαν λάθη κατά την διαδικασία της αξιολόγησης.

5.3 Συμπεράσματα αξιολόγησης

Το σημαντικότερο μέρος της αξιολόγησης αφορά τα σχόλια και τις προτάσεις βελτίωσης που έθεσαν οι χρήστες. Θέτοντας προτεραιότητες, υλοποιήθηκαν τα περισσότερα των αιτημάτων.

Από την αξιολόγηση, εντοπίστηκαν ορισμένα ζητήματα:

- Το chatbot θα πρέπει να έχει καλύτερη ροή διαλόγου
- Να βοηθάει τον χρήστη, δίνοντας το μενού σε διάφορα σημεία ώστε να τον καθοδηγεί και να τον διευκολύνει
- Τα προβλήματα (bugs) θα πρέπει να διορθωθούν
- Το User Interface (UI) να δομηθεί ομοιόμορφα

Έπειτα, έγιναν αρκετές διορθώσεις και φυσικά test, ώστε να προσφέρουμε μια καλύτερη version και μια καλύτερη εμπειρία στους χρήστες.

1. Αρχικά, προστέθηκε μια σημείωση για το πώς μπορεί ο χρήστης να επανέλθει ή να ζητήσει το κεντρικό μενού επιλογών.
2. Το σενάριο που αφορά τα μαθήματα ανά εξάμηνο, δομήθηκε διαφορετικά. Συγκεκριμένα, προστέθηκαν buttons, δίνοντας τρεις επιλογές (1^ο εξάμηνο, 2^ο και 3^ο). Επιπλέον, εάν ο χρήστης εισάγει κάποια διαφορετική επιλογή (όπως 5^ο εξάμηνο), το chatbot διευκρινίζει ποιες είναι οι διαθέσιμες επιλογές.
3. Επιπρόσθετα, δημιουργήθηκε ένας ακόμα έλεγχος (custom action) που αφορά την αποστολή email. Στο σενάριο όπου ο χρήστης, ενώ εισάγει όλα τα στοιχεία, αποφασίζει τελικά ότι δεν θα αποστείλει το email, καθαρίζεται σωστά η φόρμα και δεν κρατάει κανένα στοιχείο.
4. Όσο αφορά διορθώσεις στο UI (ακρωνύμιο του User Interface), στο κομμάτι της διεπαφής του χρήστη, έγιναν αρκετές αλλαγές, ώστε να υπάρχει ομοιομορφία και αρμονία στον ιστότοπο.
5. Αρχεία PDF (όπως το ωρολόγιο πρόγραμμα), απεικονίζονται πλέον με καλύτερη μορφή και σε πιο σύντομο χρόνο και προστέθηκε η δυνατότητα αποστολής του αρχείου με email.
6. Τέλος, οι χρήστες ζήτησαν δύο πρόσθετες πληροφορίες, οι οποίες προστέθηκαν στα user stories και στην συνέχεια υλοποιήθηκαν. Η πρώτη αφορά πληροφορίες για υποτροφίες και η δεύτερη, τις ώρες γραφείου των καθηγητών. Η συγκεκριμένη

πληροφορία επειδή δεν υπάρχει αναρτημένη για όλους τους καθηγητές του Τμήματος, δόθηκε από το επίσημο site της σχολής.

6

Επίλογος

Στο κεφάλαιο αυτό, συνοψίζουμε τα συμπεράσματα από το σύνολο της εκπόνησης αυτής της εργασίας αλλά και τις προτάσεις βελτίωσης της υπάρχουσας υλοποίησης.

6.1 Συμπεράσματα

Η έρευνα αλλά και η υλοποίηση της παρούσας διπλωματικής εργασίας, μας έκανε να συνειδητοποιήσουμε αρχικά πόσο σημαντικά είναι τα chatbots και τον λόγο που αναπτύσσονται τόσο ραγδαία. Η βοήθεια που μπορούν να προσφέρουν είναι τεράστια ιδιαίτερα όταν αναλογιζόμαστε chatbots σε εξυπηρέτηση πελατών, με άμεση εξυπηρέτηση σε συνδυασμό με την συνεχή διαθεσιμότητα.

Όσο αφορά τη μελέτη των διαθέσιμων εργαλείων, μπορεί κανείς να πει ότι παρουσιάζει μεγάλο ενδιαφέρον, μιας και υπάρχουν ποικίλα συστήματα, εργαλεία ή Online πλατφόρμες. Υπάρχουν εργαλεία στα οποία μπορεί κανείς να δημιουργήσει το chatbot του χωρίς να απαιτούνται προγραμματιστικές γνώσεις, αλλά από την άλλη υπάρχουν πιο σύνθετα στα οποία ποικίλουν οι γλώσσες προγραμματισμού για την υλοποίηση.

Η αρχική μελέτη των διαθέσιμων εργαλείων μας βοήθησε να ανακαλύψουμε το συγκεκριμένο πεδίο και τους τρόπους υλοποίησης ενός chatbot. Τελικά η υλοποίηση της παρούσας εργασίας έγινε χρησιμοποιώντας το Rasa Framework. Μετά από την ολοκλήρωσή του θεωρούμε ότι πράγματι ήταν το κατάλληλο εργαλείο και επίσης οι δυνατότητες που παρέχει κάλυψαν τις ανάγκες υλοποίησης. Η εκμάθηση της Python, στην διάρκεια του project

πρόσθεσε άλλο ένα θετικό αποτέλεσμα στο σύνολο της εμπειρίας. Επιπλέον, προσφέρει πάρα πολλές δυνατότητες σαν framework από την δημιουργία ενός chatbot ως την διαδικασία του testing. Προσπαθήσαμε να αγγίξουμε έστω ή να πειραματιστούμε με τα βασικότερα προσφερόμενα κομμάτια και αποτελεί μια επιθυμία συνέχισης για το μέλλον.

6.2 Προτάσεις βελτίωσης και μελλοντική έρευνα

Αρχικά, πιστεύουμε ότι τα chatbot μπορούν πάντα και συνέχεια να βελτιώνονται. Με την έννοια ότι τα περιθώρια βελτίωσης δεν σταματάνε ποτέ ή τουλάχιστον μέχρι το chatbot να φτάσει στο ιδανικό σημείο να απαντά και να διαχειρίζεται τα θέματα ως άνθρωπος.

Μια πρόταση που θα θέλαμε μελλοντικά να υλοποιηθεί, είναι η αγγλική version του WiBot, δηλαδή το chatbot να μπορεί να απαντήσει και στην αγγλική γλώσσα, δίνοντας επιλογή στον χρήστη.

Επιπλέον, το WiBot δημιουργήθηκε με την βλέψη, πως θα αξιοποιηθεί στην πραγματικότητα από το Τμήμα. Οι προδιαγραφές αλλά και η υλοποίηση, σε όλο το φάσμα της εκπόνησης, είχαν σαν παράμετρο τις πραγματικές συνθήκες. Οπότε, θεωρούμε πως θα μπορούσε να ενσωματωθεί στην επίσημη σελίδα του Τμήματος ή και σε κάποια εφαρμογή όπως facebook, slack. Σε αυτή την περίπτωση, η παρούσα έκδοση θα μπορούσε να επεκταθεί, προσθέτοντας και άλλες λειτουργίες. Κυρίως όμως θα θέλαμε να συνεχίσει η ίδια λογική, με την δυναμική εξαγωγή πληροφορίας και όχι με την δημιουργία βάσης δεδομένων.

Ακόμη, θα θέλαμε με κάποιον τρόπο να μπορεί ο χρήστης να συνδεθεί και να ταυτοποιηθεί από άλλα συστήματα. Αυτό θα βοηθούσε πολύ σε σημεία που χρειάζεται να εξυπηρετηθεί από την Γραμματεία του Τμήματος. Για παράδειγμα, για την έκδοση κάποιου πιστοποιητικού. Μια ακόμη προσθήκη που θα θέλαμε να υλοποιήσουμε είναι η δημιουργία log files, για πιθανά προβλήματα αλλά και στατιστικά στοιχεία. Επιπλέον, επέκταση της αποστολής email, με την δυνατότητα επιλογής παραλήπτη (καθηγητές, γραμματεία κ.α.). Πρόσθετα σε αυτό θα μπορούσε να υπάρξει δυνατότητα είσοδος και ταυτοποίηση χρήστη για την αποστολή email, πάντα σε συμμόρφωση με την πολιτική του GDPR (Γενικός Κανονισμός Προστασίας Προσωπικών Δεδομένων).

Τέλος, αξίζει να αναφερθούμε στην ασφάλεια και ειδικότερα στην διαχείριση τεχνολογικών ευπαθειών. Μια διαδικασία που απαιτείται από κάθε εφαρμογή που βρίσκεται σε παραγωγικό περιβάλλον, είναι ο εντοπισμός και η αξιολόγηση τεχνολογικών ευπειθειών (vulnerability assessment). Θα θέλαμε τουλάχιστον μελλοντικά να εκτελέσουμε διάφορους ελέγχους για την διασφάλιση της εφαρμογής μας.

Βιβλιογραφία

- [1] Y. Francino, "user story", *Techtarget.com*. [Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/user-story>.
- [2] Pandorabots, Inc, "Pandorabots: Home," Pandorabots.com. [Online]. Available: <https://home.pandorabots.com/home.html>.
- [3] "SIML Bot," Syn.co.in. [Online]. Available: <https://developer.syn.co.in/tutorial/bot/siml/index.html>.
- [4] "Introduction to Rasa Open Source," Rasa.com. [Online]. Available: <https://rasa.com/docs/rasa/>.
- [5] S. Hussain, O. Ameri Sianaki, and N. Ababneh, "A Survey on Conversational Agents/Chatbots Classification and Design Techniques," in *Advances in Intelligent Systems and Computing*, Cham: Springer International Publishing, 2019, pp. 946–956.
- [6] IBM Cloud Education, "What is Natural Language Processing?," *Ibm.com*. [Online]. Available: <https://www.ibm.com/cloud/learn/natural-language-processing>
- [7] G. G. Chowdhury, "Natural language processing," *Annu. rev. inf. sci. technol.*, vol. 37, no. 1, pp. 51–89, 2005.
- [8] E. Kavlakoglu, "NLP vs. NLU vs. NLG: the differences between three natural language processing concepts," *Watson Blog*, 12-Nov-2020. [Online]. Available: <https://www.ibm.com/blogs/watson/2020/11/nlp-vs-nlu-vs-nlg-the-differences-between-three-natural-language-processing-concepts/>
- [9] IBM Cloud Education, "chatbots-explained," *Ibm.com*. [Online]. Available: <https://www.ibm.com/cloud/learn/chatbots-explained>.
- [10] "What is a chatbot?," *Oracle.com*. [Online]. Available: <https://www.oracle.com/chatbots/what-is-a-chatbot>
- [11] C. Farris, "Evolution of chatbots," *Capacity*, 07-Jan-2020. [Online]. Available: <https://capacity.com/chatbots/evolution-of-chatbots>
- [12] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, no. 100006, p. 100006, 2020.
- [13] IJERT Journal, "IJERT-Introduction to AI Chatbots," *International Journal of Engineering*

- [14] S. Andrieiev, "What are the Advantages and Disadvantages of Chatbots?," *Medium*, 21-Jan-2019. [Online]. Available: <https://jelvix.medium.com/what-are-the-advantages-and-disadvantages-of-chatbots-581fba4fea8f>.
- [15] T. Chong, T. Yu, D. I. Keeling, and K. de Ruyter, "AI-chatbots on the services frontline addressing the challenges and opportunities of agency," *J. Retail. Consum. Serv.*, vol. 63, no. 102735, p. 102735, 2021.
- [16] "Open source conversational AI," *Rasa.com*, 01-Dec-2020. [Online]. Available: <https://rasa.com/>.
- [17] "Introduction to," *TensorFlow*. [Online]. Available: <https://www.tensorflow.org/learn>
- [18] "Features - PyCharm," *Jetbrains.com*. [Online]. Available: <https://www.jetbrains.com/pycharm/features/>.
- [19] K. Fakhroutdinov, "Unified Modeling Language (UML) description, UML diagram examples, tutorials and reference for all types of UML diagrams - use case diagrams, class, package, component, composite structure diagrams, deployments, activities, interactions, profiles, etc," *Uml-diagrams.org*. [Online]. Available: <https://www.uml-diagrams.org/>.
- [20] "Apache Tika – Apache Tika," *Apache.org*. [Online]. Available: <https://tika.apache.org/>.
- [21] "Testing your assistant," *Rasa.com*. [Online]. Available: <https://rasa.com/docs/rasa/testing-your-assistant/>.
- [22] "Sklearn.Metrics.Precision_recall_fscore_support," *scikit-learn*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html.
- [23] Βλαχάβας Ι., Κεφαλάς Π., Βασιλειάδης Ν., Κόκκορας Φ. και Σακελλαρίου Η. (2006). Τεχνητή Νοημοσύνη (Β' έκδοση), Εκδόσεις Γκιούρδα
- [24] "Hands-on guide to natural language processing using spacy," *Analytics India Magazine*, 16-Dec-2020. [Online]. Available: <https://analyticsindiamag.com/nlp-deep-learning-nlp-framework-nlp-model/>.

ΠΑΡΑΡΤΗΜΑ Α : User Stories

Στο παράρτημα Α, καταγράφονται αναλυτικά τα user stories που υλοποιήθηκαν. Συγκεκριμένα, η πιθανή ερώτηση του χρήστη για κάθε user story και η αντίστοιχη απάντηση από το chatbot.

User story 1 (public) - Μαθήματα εξαμήνου

User: Θα ήθελα να μάθω ποια είναι τα μαθήματα ανά εξάμηνο.

ChatBot: Για το χειμερινό ή το εαρινό εξάμηνο ενδιαφέρεστε;

User: Και για τα δύο./ Για το χειμερινό./ Για το εαρινό.

ChatBot: Τα μαθήματα του (πχ) χειμερινού εξαμήνου είναι:

1. Μεθοδολογίες Έρευνας
2. Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές
3. Ασφάλεια Διαδικτυακών Πληροφοριακών Συστημάτων
4. Συστήματα Κινητού και Διάχυτου Υπολογισμού
5. Μηχανική Μάθηση

User story 2 (public) - Εξέταση Μάθημα Σεπτεμβρίου

User: Θα ήθελα να μάθω πως θα διεξαχθεί η εξέταση για το Μάθημα τον Σεπτέμβριο.

ChatBot: Παρακαλώ πείτε μου τον αριθμό του μητρώου σας.

User: 72020

ChatBot: Η εξέταση θα διεξαχθεί εξ αποστάσεως, μέσω της πλατφόρμας zoom.

User story 3 (private) - Στοιχεία επικοινωνίας Καθηγητή

User: Θα ήθελα να μάθω τους τρόπους επικοινωνίας με τον καθηγητή/καθηγήτρια «Καθηγητή».

ChatBot: Για να σας προωθήσω προσωπικά στοιχεία, θα πρέπει να γνωρίζω τον αριθμό του μητρώου σας.

User: AM 72020

ChatBot: Το τηλέφωνο επικοινωνίας είναι: 2234034498 και το mail: example@ihu.gr

User story 4 (public) - Τρόπους επικοινωνίας με το Τμήμα Φοιτητικής Μέριμνας

User: Θα ήθελα να μάθω τους διαθέσιμους τρόπους επικοινωνίας με το Τμήμα Φοιτητικής Μέριμνας

ChatBot: Με το Τμήμα Φοιτητικής Μέριμνας μπορείτε να επικοινωνήσετε μέσω email: infositisi@the.ihu.gr

τηλέφωνο επικοινωνίας: 2310013667

διεύθυνση: Διεθνές Πανεπιστήμιο της Ελλάδος, Αλεξάνδρεια Πανεπιστημιούπολη, ΤΘ 141
TK 57400 Σίνδος Θεσσαλονίκη

User story 5 (public) - Μαθήματα πρόγραμμα μερικής φοίτησης

User: Θα ήθελα να μάθω πόσα μαθήματα μπορώ να επιλέξω στο πρόγραμμα μερικής φοίτησης

ChatBot: Η διαδικασία μερικής φοίτησης έχει ελάχιστη διάρκεια 5 εξαμήνων. Κάθε εξάμηνο, οι φοιτητές θα πρέπει υποχρεωτικά να επιλέξουν και να δηλώσουν την παρακολούθηση τουλάχιστον 2 μαθημάτων μεταξύ αυτών τα οποία διδάσκονται το τρέχον εξάμηνο. Άνω όριο μαθημάτων ανά εξάμηνο αποτελεί το συνολικό πλήθος των μαθημάτων που διδάσκονται το τρέχον εξάμηνο, δηλ. 5 μαθήματα. Κάθε έτος, οι φοιτητές θα πρέπει υποχρεωτικά να έχουν επιλέξει και δηλώσει την παρακολούθηση τουλάχιστον 5 μαθημάτων συνολικά (50% των μαθημάτων). Το πρώτο εξάμηνο σπουδών είναι υποχρεωτική η παρακολούθηση τουλάχιστον των μαθημάτων:

M101 – Μεθοδολογίες Έρευνας

M102 – Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές

User story 6 (public) - Επικοινωνία γραμματεία

User: Θα ήθελα να μάθω τους τρόπους επικοινωνίας με την γραμματεία (διεύθυνση, mail)

ChatBot: Μπορείτε να επικοινωνήσετε μέσω email στα ακόλουθα:

info-msc@iee.ihu.gr, sec-msc@iee.ihu.gr

Το τηλέφωνο επικοινωνίας είναι: 2310013621

Και η διεύθυνση: Αλεξάνδρεια Πανεπιστημιούπολη, Σίνδος, Θεσσαλονίκη, TK: 57400

User story 7 (public) - Έναρξη χειμερινό/εαρινό εξάμηνο

User: Θα ήθελα να μάθω πότε ξεκινάει το χειμερινό/εαρινό εξάμηνο

ChatBot: Η απάντηση δίνεται από το US24 με το ακαδημαϊκό ημερολόγιο.

User story 8 (public) - Ανακοινώσεις Τμήματος

Η απάντηση είναι δυναμική και έρχεται μέσω του API

User story 9 (public) - Αίτηση εισαγωγής στο ΠΜΣ

User: Θα ήθελα να μάθω τα απαραίτητα έγγραφα για την υποβολή αίτησης στο πρόγραμμα

ChatBot: Οι υποψήφιοι μεταπτυχιακοί θα πρέπει να υποβάλουν:

Αίτηση, η οποία υποβάλλεται αποκλειστικά ηλεκτρονικά στην ιστοσελίδα του ΠΜΣ <https://msc.iee.ihu.gr/application/?lang=el> μαζί με τα υπόλοιπα δικαιολογητικά.

Πλήρες βιογραφικό σημείωμα, το οποίο περιλαμβάνει στοιχεία για τις σπουδές, την ερευνητική ή και επαγγελματική δραστηριότητα και τις πιθανές επιστημονικές εργασίες του υποψηφίου.

Αντίγραφο πτυχίου. Για όσους προέρχονται από ΑΕΙ του εξωτερικού, βεβαίωση ισοτιμίας πτυχίου από το ΔΟΑΤΑΠ.

Πιστοποιητικό αναλυτικής βαθμολογίας.

Τεκμηρίωση επαρκούς γνώσης της Αγγλικής γλώσσας επιπέδου γλωσσομάθειας τουλάχιστον B2 (Lower) ή νόμιμα αναγνωρισμένο από το ΔΙ.Κ.Α.Τ.Σ.Α. ή Δ.Ο.Α.Τ.Α.Π. πανεπιστημιακό τίτλο από αγγλόφωνο πρόγραμμα σπουδών.

Δύο συστατικές επιστολές, κατά προτίμηση η μία από μέλος ΔΕΠ. Οι συστατικές μπορούν να κατατίθενται ταχυδρομικά ή ηλεκτρονικά στην διεύθυνση sec-msc@iee.ihu.gr. Οι συστατικές επιστολές μπορούν να κατατίθενται σε μεταγενέστερο χρόνο από την αίτηση αλλά σε κάθε περίπτωση μέχρι την προθεσμία αποδοχής υποψηφιοτήτων. Προαιρετικά, οι υποψήφιοι μπορούν να καταθέσουν και τα παρακάτω δικαιολογητικά :

Επιστημονικές Δημοσιεύσεις, Διακρίσεις, λοιποί τίτλοι σπουδών πλην του πρώτου πτυχίου (εάν υπάρχουν).

Αποδεικτικά επαγγελματικής εμπειρίας (εάν υπάρχουν).

User story 10 (public) - Πλατφόρμα e-learning

User: θέλω να μάθω ποια είναι η πλατφόρμα e-learning

ChatBot: Η ηλεκτρονική πλατφόρμα του Τμήματος (Moodle) βρίσκεται στον ακόλουθο σύνδεσμο

<https://moodle.teithe.gr/course/index.php?categoryid=138>

Το Moodle αποτελεί ελεύθερο λογισμικό διαχείρισης εκπαιδευτικού περιεχομένου. Κάθε μάθημα διαθέτει σελίδα στο moodle, όπου ο κάθε διδάσκων παρουσιάζει το μάθημά του, το περίγραμμα, το περιεχόμενο, τη συνιστώμενη βιβλιογραφία, την ύλη του μαθήματος κ.ό.κ

User story 11 (public) - Πιστοποιητικό / Αποδεικτικό Εγγραφής

User: θέλω να εκδώσω Πιστοποιητικό / Αποδεικτικό Εγγραφής

ChatBot: Παρακαλώ αποστείλετε το αίτημα σας στο info-msc@it.teithe.gr

User story 12 (public) - Έκδοση της φοιτητικής ταυτότητας (Link)

User: θέλω να μάθω το site για την έκδοση της φοιτητικής ταυτότητας

ChatBot: Η ηλεκτρονική αίτηση για την έκδοση ακαδημαϊκής ταυτότητας μπορεί να πραγματοποιηθεί στη δικτυακή πύλη <https://submit-academicid.minedu.gov.gr>

* Για την έκδοση της ακαδημαϊκής ταυτότητας χρειάζεται να ενεργοποιηθεί πρώτα ο λογαριασμός στη διεύθυνση <https://uregister.the.ihu.gr>

User story 13 (public) - Διαθέσιμες ηλεκτρονικές υπηρεσίες

User: θέλω να μάθω ποιες είναι οι ηλεκτρονικές υπηρεσίες (webmail, pithia ..)

ChatBot: Οι διαθέσιμες υπηρεσίες καθώς και οι ιστότοποι πρόσβασης

Ηλεκτρονική γραμματεία : είναι <http://pithia.teithe.gr/unistudent/>

Ηλεκτρονικό ταχυδρομείο (email): <http://webmail.teithe.gr>

Πλατφόρμα ασύγχρονης εκπαίδευσης (Moodle): <https://moodle.teithe.gr>

Υπηρεσία εικονικού ιδιωτικού δικτύου (VPN): Οδηγίες για τη χρήση της υπηρεσίας δίνονται στη διεύθυνση <http://www.noc.teithe.gr/open-vpn>

Ηλεκτρονική σελίδα απόκτησης ακαδημαϊκής ταυτότητας (πασο): <https://submit-academicid.minedu.gov.gr>

Ηλεκτρονική σελίδα δηλώσεων συγγραμμάτων: www.eudoxus.gr όπου

στο menu φοιτητές βρίσκεται ο σχετικός σύνδεσμος για τη δήλωση συγγραμμάτων.

Υπηρεσία ασύρματης πρόσβασης (Eduroam): Οδηγίες για τη χρήση της υπηρεσίας δίνονται στη διεύθυνση <http://www.noc.teithe.gr/eduroam>

User story 14 (public) - Τρόποι πρόσβασης στην σχολή

User: θέλω να μάθω ποιοι είναι οι τρόποι πρόσβασης στην σχολή

ChatBot: 14ο χλμ. Θεσ/νίκης - Ν. Μουδανίων, 57001 Θέρμη Τ.Θ. 141, 574 00 Σίνδος

Με τις γραμμές αστικής συγκοινωνίας Νο 67 Β και 68 Α (ΟΑΣΘ)

User story 15 (public) - Πληροφορίες Erasmus

User: θέλω να μάθω πληροφορίες για το Erasmus

ChatBot: Οι φοιτητές του ΠΜΣ έχουν τη δυνατότητα μέσω του προγράμματος ERASMUS να πραγματοποιήσουν μέρος των σπουδών τους σε συνεργαζόμενα Ευρωπαϊκά Πανεπιστήμια,

καθώς επίσης και να αξιοποιήσουν προαιρετικά τη δυνατότητα επιδοτούμενης πρακτικής άσκησης διάρκειας 2–6 μηνών σε επαγγελματικούς και ερευνητικούς Ευρωπαϊκούς φορείς.

Περισσότερες πληροφορίες στον σύνδεσμο <http://erasmus.uop.gr/index.php>

User story 16 (public) - Πληροφορίες ορκωμοσία

User: θέλω να μάθω πληροφορίες για την διαδικασία της ορκωμοσίας

ChatBot: Τα δικαιολογητικά που πρέπει να υποβληθούν είναι:

Αίτηση σε μορφή docx

Ακαδημαϊκή ταυτότητα

Βεβαίωση από τη Βιβλιοθήκη της πανεπιστημιούπολης

Για να μπορέσει ένας φοιτητής να υποβάλλει αίτηση θα πρέπει να έχει ολοκληρώσει επιτυχώς τα δέκα μαθήματα του ΠΜΣ και τη Διπλωματική Εργασία.

User story 17 (public) - Λίστα Διδάσκοντες

User: θέλω να μάθω τους τρόπους επικοινωνίας με τους διδάσκοντες

ChatBot: Περισσότερες πληροφορίες στον σύνδεσμο <https://msc.iee.ihu.gr/faculty/?lang=el>

User story 18 (public) – Υποβολή αιτήσεων

User: θέλω να μάθω πότε ολοκληρώνονται οι αιτήσεις για τους νέους υποψήφιους

ChatBot: Για την εισαγωγή φοιτητών στο ΠΜΣ Ευφυείς Τεχνολογίες Διαδικτύου οι αιτήσεις υποβάλλονται ηλεκτρονικά περίπου από τέλος Ιουνίου έως τις πρώτες ημέρες του Σεπτεμβρίου. Ενημερωθείτε για το τρέχον Ακαδημαϊκό έτος, από την επίσημη ιστοσελίδα του Τμήματος.

User story 19 (public) - Ανακοινώσεις του Τμήματος

User: θέλω να μάθω τις τελευταίες ανακοινώσεις του Τμήματος

ChatBot: Απάντηση μέσω API

User story 20 (public) - Τρόποι επικοινωνίας καθηγητή

User: θέλω να μάθω τον τον τρόπο επικοινωνίας με τον/την καθηγητή/καθηγήτρια

ChatBot: Απάντηση μέσω API

User story 21 (public) - Πληροφορίες που προσφέρει το chatbot

User: θέλω να μάθω τι πληροφορίες μπορεί να προσφέρει το chatbot

ChatBot: Μπορώ να σου δώσω πληροφορίες όπως:

- Ανακοινώσεις
- Στοιχεία επικοινωνίας καθηγητών
- Γενικές πληροφορίες
- Τρόποι πρόσβασης στην σχολή
- Πληροφορίες για την υποβολή αίτησης στο πρόγραμμα
- Έναρξη εξαμήνου
- Μαθήματα ανά εξάμηνο
- Πληροφορίες για το πρόγραμμα μερικής φοίτησης
- Πιστοποιητικό / Αποδεικτικό Εγγραφής
- Πληροφορίες για την πλατφόρμα e-learning
- Τρόποι επικοινωνίας με την γραμματεία/ διεύθυνση
- Τρόποι επικοινωνίας με το Τμήμα Φοιτητικής Μέριμνας
- Διαθέσιμες πλατφόρμες- ηλεκτρονικές υπηρεσίες
- Πληροφορίες για το Erasmus
- Πληροφορίες για την έκδοση της φοιτητικής ταυτότητας
- Πληροφορίες για την διαδικασία ορκωμοσίας
- Το Ωρολόγιο πρόγραμμα
- Τον οδηγό σπουδών (δεν βρέθηκε αντίστοιχο αρχείο)
- Το ακαδημαϊκό ημερολόγιο
- Υποτροφίες
- Ώρες γραφείου
- Αποστολή email
- Πληροφορίες για το Τμήμα και την σχολή
- Πληροφορίες για το chatbot

User story 22 (public) - Ωρολόγιο πρόγραμμα

User: θέλω να μάθω το ωρολόγιο πρόγραμμα

ChatBot: Απάντηση από αρχείο PDF

User story 23 (public) - Οδηγό σπουδών

User: θέλω να δω τον οδηγό σπουδών

ChatBot: Απάντηση από αρχείο PDF

User story 24 (public) - Ακαδημαϊκό ημερολόγιο

User: θέλω να μάθω το ακαδημαϊκό ημερολόγιο

ChatBot: Απάντηση από αρχείο PDF

User story 25 (public) - Γενικές πληροφορίες για το Τμήμα και την σχολή

User: θέλω να μάθω πληροφορίες για το Τμήμα και την σχολή

ChatBot: Το Διεθνές Πανεπιστήμιο της Ελλάδος (ΔΙ.ΠΑ.Ε.) ιδρύθηκε με το νόμο 3391 του 2005 και άρχισε να λειτουργεί το 2008 όταν έλαβε χρηματοδότηση από το ΕΣΠΑ 2007-2013 μέσω του επιχειρησιακού προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και εδρεύει στη Θέρμη του νομού Θεσσαλονίκης

Στο Διεθνές Πανεπιστήμιο της Ελλάδας λειτουργούν 34 Τμήματα τα οποία υπάγονται σε επτά (7) Σχολές. Πιο συγκεκριμένα:

- Σχολή Οικονομίας και Διοίκησης, με έδρα τη Θεσσαλονίκη
- Σχολή Κοινωνικών Επιστημών, με έδρα τη Θεσσαλονίκη
- Σχολή Επιστημών Υγείας, με έδρα τη Θεσσαλονίκη
- Σχολή Μηχανικών, με έδρα τις Σέρρες
- Σχολή Επιστημών Σχεδιασμού, με έδρα τις Σέρρες
- Σχολή Θετικών Επιστημών, με έδρα την Καβάλα
- Σχολή Γεωτεχνικών Επιστημών, με έδρα τη Δράμα

Περισσότερες πληροφορίες θα βρείτε, στην επίσημη ιστοσελίδα: <https://www.ihu.gr/>

Όσο αφορά το Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ) Ευφυείς Τεχνολογίες Διαδικτύου, ανήκει στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων (του Διεθνούς Πανεπιστημίου της Ελλάδος) ιδρύθηκε και λειτουργεί από το Ακαδημαϊκό εξάμηνο 2012-2013 και εγκρίθηκε πρόσφατα η συνέχιση λειτουργίας του από το Υπουργείο Παιδείας (ΦΕΚ 2121/21-5-2021). Οδηγεί σε Μεταπτυχιακό Δίπλωμα Ειδίκευσης (Μ.Δ.Ε.) επιπέδου εκπαίδευσης 7, στο Εθνικό Πλαίσιο Προσόντων.

Περισσότερες πληροφορίες θα βρείτε, στην επίσημη ιστοσελίδα: <https://msc.iee.ihu.gr/>

User story 26 (public) – Αποστολή Email

User: θέλω να αποστείλω ένα mail

ChatBot: Συμπλήρωση στοιχείων φόρμας και εκτέλεση αποστολής

User story 27 (public) - Γενικές πληροφορίες για τις υποτροφίες

User: θέλω να μάθω πληροφορίες για τις υποτροφίες

ChatBot: Πληροφορίες για τις υποτροφίες της σχολής (για το 2020-2021) μπορείς να βρεις στο παρακάτω σύνδεσμο, εναλλακτικά μπορείς να ανατρέξεις στις ανακοινώσεις της σχολής.
<https://rc.ihu.gr/?p=4004>

User story 28 (public) - Πληροφορίες για τις ώρες γραφείου

User: Θα ήθελα να μάθω τις ώρες γραφείου των καθηγητών

ChatBot: Μπορείς να αναζητήσεις πληροφορίες για τις ώρες γραφείου και συγκεκριμένα για τα μέλη Δ.Ε.Π. στην σελίδα https://www.iee.ihu.gr/staff_category/faculty-member/. Στην ενότητα 'Προσωπικό' θα βρεις πληροφορίες και για το υπόλοιπο προσωπικό

ΠΑΡΑΡΤΗΜΑ Β : Evaluation Results (1)

Το παράρτημα Β, περιέχει τα αποτελέσματα των ερωτήσεων στο στάδιο της (πρώτης) αξιολόγησης. Ειδικότερα, παρουσιάζονται τα διαγράμματα των απαντήσεων καθώς και τα σχόλια και οι προτάσεις βελτίωσης.

Wi-Bot eval

Αξιολόγηση chatbot Wi-Bot

Πόσο εξοικειωμένοι είστε με τα chatbots; (1 δεν έχω χρησιμοποιήσει ποτέ, 5 χρησιμοποιώ πολύ συχνά) *

1

2

3

4

5

Πως αξιολογείτε την εμπειρία σας συγκριμένα με το Wi-Bot; (1 καθόλου καλή, 5 άριστη) *

1

2

3

4

5

Το Wi-Bot κάλυψε τις ανάγκες σας; (1 καθόλου, 5 άριστα) *

1

2

3

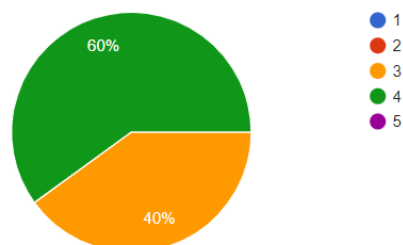
4

5

Εικόνα Β 1 Φόρμα αξιολόγησης Α'

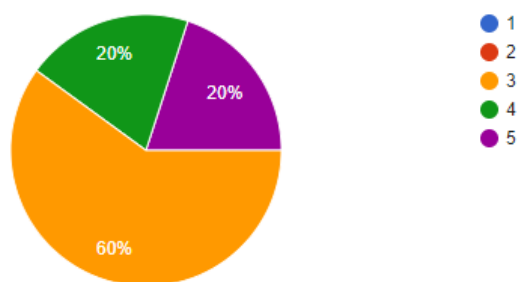
Πόσο εξοικειωμένοι είστε με τα chatbots; (1 δεν έχω χρησιμοποιήσει ποτέ, 5 χρησιμοποιώ πολύ συχνά)

5 απαντήσεις



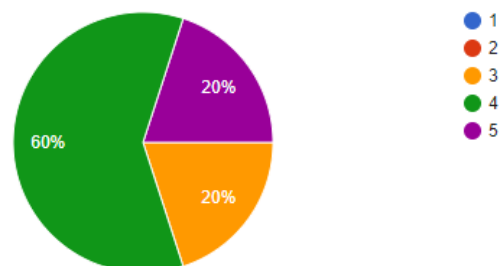
Εικόνα Β 2 Πόσο εξοικειωμένοι είστε με τα chatbots;

Πως αξιολογείτε την εμπειρία σας συγκεκριμένα με το Wi-Bot; (1 καθόλου καλή, 5 άριστη)
5 απαντήσεις



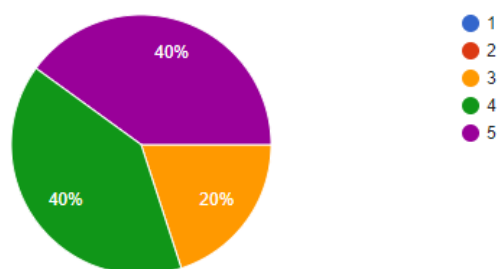
Εικόνα Β 3 Πως αξιολογείτε την εμπειρία σας συγκεκριμένα με το WiBot;

Το Wi-Bot κάλυψε τις ανάγκες σας; (1 καθόλου, 5 άριστα)
5 απαντήσεις



Εικόνα Β 4 Το WiBot κάλυψε τις ανάγκες σας;

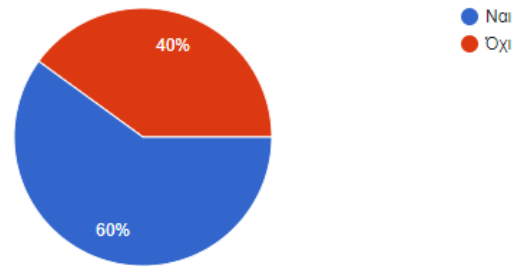
Το Wi-Bot ήταν εύκολο στην χρήση; (1 καθόλου, 5 άριστα)
5 απαντήσεις



Εικόνα Β 5 Το WiBot ήταν εύκολο στην χρήση;

Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος;

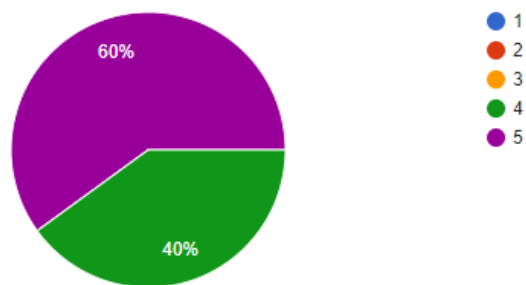
5 απαντήσεις



Εικόνα Β 6 Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος;

Πως αξιολογείτε την χρησιμότητα του Wi-Bot; (1 καθόλου χρήσιμο, 5 άριστα)

5 απαντήσεις



Εικόνα Β 7 Πως αξιολογείτε την χρησιμότητα του WiBot;

Ποιες άλλες πληροφορίες θα μπορούσε να δώσει;

3 απαντήσεις

τις ωρες γραφειου των καθηγητων (αν υπαρχουν κιολας)

Τηλέφωνα επικοινωνίας

Περισσότερες πληροφορίες όσον αφορά λεπτομέρειες της φοίτησης, όπως κόστος διδάκτρων, δυνατότητα λήψης υποτροφίας κλπ.

Εικόνα Β 8 Ποιες άλλες πληροφορίες θα μπορούσε να δώσει;

Το chatbot λειτουργεί αρκετά καλά και ήταν ευχάριστη η "επικοινωνία" μαζί του καθώς μπορέσα να εκμαιεύσω τις πληροφορίες που έψαχνα. Παρόλαυτα παρακάτω έχω προσθέσει κάποια σημεία που θα μπορούσαν να βελτιωθούν για να είναι περισσότερο user friendly.

- Όταν κάνω λάθος σε κάποια απάντηση θα ήθελα να με επαναφέρει στο θέμα που επέλεξα νωρίτερα στο μενού ή να μου ξαναεμφανίζει το μενού.
- Επίσης, γράφοντας κατηγορίες εμφανίζει τις Ανακοινώσεις του τμήματος ενώ θα ήθελα τις κατηγορίες του μενού.
- Θα βόλευε να γράφει κάποιες λέξεις κλειδιά που μπορώ να γράψω για να μου εμφανίσει το μενού και τι ανακοινώσεις ή ότι άλλο μπορεί και εμφανίζει το chatbot (π.χ. γράψτε την λέξη μενου για να εμφανιστεί το μενου)
- Εγραψα την λέξη μαθήματα, με ρώτησε το εξάμηνο, έβαλα 1ο, δεν το δεχτηκε. Μετά, εγραψα πρώτο και πάλι δεν το δεχτηκε οποτε δεν μπορέσα να συνεχισω απο εκει και περα καθώς επρεπε να γράψω μενου και να επιλέξω "Μαθηματα ανα εξαμηνο".
- Θα ήθελα το μενού να είναι μικρότερο για να μην φαχνει ο χρηστης παρα πολλες επιλογες. Για παραδειγμα θα μπορούσαν να υπάρχουν υποκατηγορίες ετσι ώστε να είναι πιο δομημένο το μενού.
- Η επιλογή την αποστολής email κρατάει το email και αν ας έχω πατήσει στην ερωτηση αποστολη "Όχι".
- Επίσης η γραμματοσειρά να είναι ίδια σε όλο το chatbox.

Ευχαριστώ για την εμπειρία!

Εικόνα Β 9 Αφήστε κάποια σχόλια (1)

μετα την πρωτη απαντηση θα μπορουσε να ρωττει για να ξαναεμφανισει το αρχικο μενου με τις προτασεις αναζητησης, ωστε να μην χρειαζεται να πληκτρολογησει ο χρηστης την ερωτηση του

Στο πρόγραμμα και στα στοιχεία επικοινωνίας θα μπορούσε να φέρνει link για πρόγραμμα. Στην ερώτηση εξάμηνο μπορεί να προστεθεί το "Θερινό ή εαρινό" ώστε να οδηγήει τον χρήστη να επιλέξει ένα από τα 2 (ή "1ο ή 2ο" αντίστοιχα). Εντολή "stop" για τις περιπτώσεις που η πληροφορία που πρέπει να παρέχεται από το bot είναι πολύ μεγάλη. Στο "εκπρόσωπος" απαντάει με την διεύθυνση της σχολής. Προσθήκη live chat εάν ο χρήστης θέλει να ρωτήσει κάτι πιο συγκεκριμένο (ή καθοδήγηση για επικοινωνία με άλλο τρόπο). Στο "Ανακοινώσεις" δίνει σαν απάντηση τις ανακοινώσεις της σχολής και μετά μπορώ είτε να επιλέξω είτε να γράψω "Επιλογή κατηγορίας" που μου άρεσε πάρα πολύ αλλά δεν μπορώ να συνεχίσω να το κάνω στο επόμενο βήμα γράφοντας πχ "Διδακτορικά". Εξαιρετική η αποστολή email μέσω bot.

Θα ήθελα να μπορεί να απαντήσει σε περισσότερες ερωτήσεις, αλλά εξεπλάγην από κάποιες απαντήσεις του, καθώς ήταν αρκετά κατατοπιστικές.

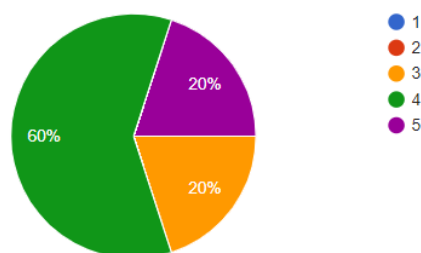
Εικόνα Β 10 Αφήστε κάποια σχόλια (2)

ΠΑΡΑΡΤΗΜΑ Γ: *Evaluation Results* (2)

Το παράρτημα Γ, περιέχει τα αποτελέσματα των ερωτήσεων στο στάδιο της δεύτερης αξιολόγησης. Ειδικότερα, παρουσιάζονται τα διαγράμματα των απαντήσεων καθώς και τα σχόλια και οι προτάσεις βελτίωσης.

Πόσο εξοικειωμένοι είστε με τα chatbots; (1 δεν έχω χρησιμοποιήσει ποτέ, 5 χρησιμοποιώ πολύ συχνά)

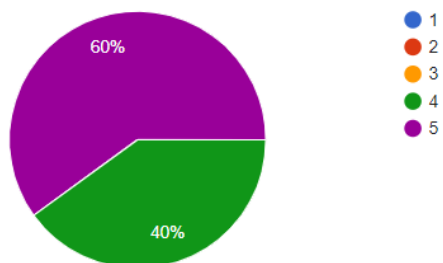
5 απαντήσεις



Εικόνα Γ 1 Πόσο εξοικειωμένοι είστε με τα chatbots;

Πως αξιολογείτε την εμπειρία σας συγκεκριμένα με το Wi-Bot; (1 καθόλου καλή, 5 άριστη)

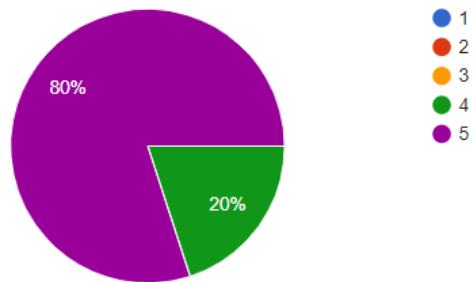
5 απαντήσεις



Εικόνα Γ 2 Πως αξιολογείτε την εμπειρία σας συγκεκριμένα με το WiBot;

Το Wi-Bot κάλυψε τις ανάγκες σας; (1 καθόλου, 5 άριστα)

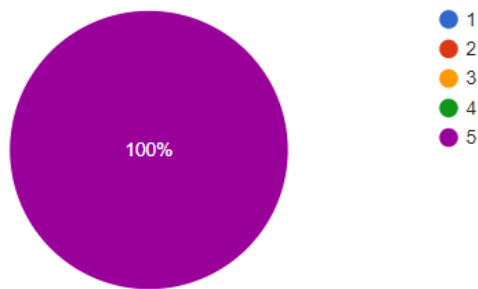
5 απαντήσεις



Εικόνα Γ 3 Το WiBot κάλυψε τις ανάγκες σας;

Το Wi-Bot ήταν εύκολο στην χρήση; (1 καθόλου, 5 άριστα)

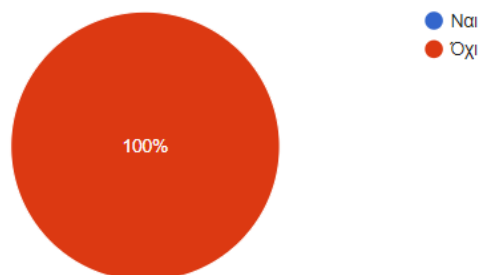
5 απαντήσεις



Εικόνα Γ 4 Το WiBot ήταν εύκολο στην χρήση;

Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος;

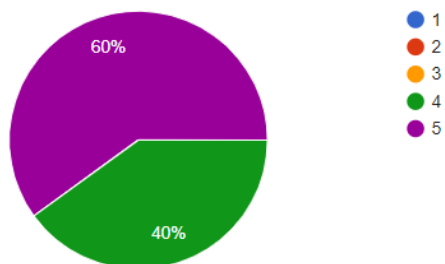
5 απαντήσεις



Εικόνα Γ 5 Θεωρείται ότι απάντησε σε κάποιο σημείο λάθος;

Πως αξιολογείτε την χρησιμότητα του Wi-Bot; (1 καθόλου χρήσιμο, 5 άριστα)

5 απαντήσεις



Εικόνα Γ 6 Πως αξιολογείτε την χρησιμότητα του WiBot;

Ποιες άλλες πληροφορίες θα μπορούσε να δώσει;

1 απάντηση

Δεν παρατήρησα κάτι που να μην μπορούσε να απαντήσει

Αφήστε κάποια σχόλια

1 απάντηση

Πολύ καλή δουλειά, φάνηκε πολύ ενημερωμένο και χρήσιμο

Εικόνα Γ 7 Άλλες πληροφορίες και σχόλια