

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
«Ғ.ДАУКЕЕВ АТЫНДАҒЫ АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС  
УНИВЕРСИТЕТІ»

коммерциялық емес акционерлік қоғамы  
Телекоммуникациялық жүйелер және желілер кафедрасы  
«ҚОРҒАУҒА ЖІБЕРІЛДІ»  
Кафедра меңгерушісі PhD, доцент Темырканова Э.К.  
(ғылыми дәрежесі, атағы, Т.А.Ж.)

«\_\_\_\_\_» «\_\_\_\_\_» \_\_\_\_\_ 2020ж.  
(қолы)

**ДИПЛОМДЫҚ ЖОБА**

Тақырыбы: Python бағдарламалық қамтамасыздандыруының көмегімен  
нейрожеліні құру

Мамандығы 5B071900 Радиотехника, электроника және телекоммуникациялар  
Орындаған Қанатбаева Сындара Нұралықызы Тобы РЭТ(МТС)-16-2  
(Т.А.Ж.)

Ғылыми жетекшісі PhD, доцент Темырканова Эльвира Кадылбековна  
(ғылыми дәрежесі, атағы, Т.А.Ж.)

Кеңесшілер:

экономикалық бөлім бойынша:

доцент Тузелбаев Бакберген Ибадиллаевич  
(ғылыми дәрежесі, атағы, Т.А.Ж.)

«\_\_\_\_\_» «\_\_\_\_\_» \_\_\_\_\_ 2020ж.  
(қолы)

өміртіршілігі қауіпсіздігі бойынша:

доцент Жандаулетова Фарида Рустембековна  
(ғылыми дәрежесі, атағы, Т.А.Ж.)

«\_\_\_\_\_» «\_\_\_\_\_» \_\_\_\_\_ 2020ж.  
(қолы)

есептеу техникасын қолдану бойынша:

доцент PhD Темырканова Эльвира Кадылбековна  
(ғылыми дәрежесі, атағы, Т.А.Ж.)

«\_\_\_\_\_» «\_\_\_\_\_» \_\_\_\_\_ 2020ж.  
(қолы)

Нормобақылаушы: аға оқытушы Накисбекова Балауса Рысқожаевна  
(ғылыми дәрежесі, атағы, Т.А.Ж.)

«\_\_\_\_\_» «\_\_\_\_\_» \_\_\_\_\_ 2020ж.  
(қолы)

Пікір беруші:

«\_\_\_\_\_» «\_\_\_\_\_» \_\_\_\_\_ 2020ж.  
(ғылыми дәрежесі, атағы, Т.А.Ж.)  
(қолы)

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
«Ғ.ДАУКЕЕВ АТЫНДАҒЫ АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС  
УНИВЕРСИТЕТІ»

коммерциялық емес акционерлік қоғамы

Ғарыш инженериясы және телекоммуникациялар институты  
телекоммуникациялық жүйелер және желілер кафедрасы  
Мамандығы 5B071900 – Радиотехника, электроника және  
телекоммуникациялар

Дипломдық жобаны орындауға берілген

**ТАПСЫРМА**

Студент Қанатбаева Сындара Нұралықызы

(Т.А.Ж.)

Жобаның тақырыбы Python бағдарламалық қамтамасыздандыруының  
көмегімен нейрожеліні құру

2019 ж. «11» 11 №147 университет бұйрығымен бекітілді.

Аяқталған жобаны тапсыру мерзімі «\_\_» 06. 2020 ж.

Жобаға алғашқы деректер (талап етілетін зерттеу (жоба) нәтижелерінің  
параметрлері және зерттеу нысанының алғашқы деректері):  
Мекеменің сапа менеджмент жүйесіне басшылық ету: ИСО-9001 халықаралық  
стандарты ,дипломалды практикасының мәліметтері.

Диплом жобасындағы әзірленуі тиіс мәселелер тізімі немесе диплом  
жобасының қысқаша мазмұны:

Кіріспе

1 Негізгі бөлім

2 Бағдарламалық өнімді жобалау

3 Бағдарламалық өнімді әзірлеу

4 Өміртіршілік қауіпсіздігі бөлімі

5 Экономикалық бөлім

Қорытынды

Әдебиеттер тізімі

А қосымшасы. Техникалық тапсырма

Б қосымшасы. Бағдарлама листингі

Графикалық материалдардың (міндетті түрде дайындалатын сызбаларды көрсету) тізімі:

Жасанды нейрондық желі құрылымы

Нейрожелі моделі

Нейрондық желілердің оқытылу процесі

Нейрожелілердің жұмыс жасау деңгейі

Нейрожелілердің жіктелуі

Жинақталған нейрожелі архитектурасы

Қайталанатын нейрожелі архитектурасы

БӨ шығын жиынтығы

Реккуренттік нейрожелілер

Кеңсенің жұмыс орындарының және құрылғылардың орналасу жоспары

Қолданбалы көмірқышқыл өрт сөндіргіші: ОУ-2

Негізгі ұсынылатын әдебиеттер:

1 Кузнецов В. В. Перспективы развития чат-ботов//Успехи современной науки . — 2016. — № 12, 16–19 стр. «HTML и CSS. Разработка и дизайн веб-сайтов» /Джон Дакетт; [пер. с англ. М. А. Райтмана]. – Москва: Эксмо, 2019. - 480 с.: ил. - (Мировой компьютерный бестселлер).

2 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489>.

3 Бегимбетова А.С., Жандаулетова Ф.Р. Безопасность жизнедеятельности. Защита от производственного шума. Методические указания к выполнению дипломного проекта, Алматинский институт энергетики и связи. – Алматы, 2010.

4 Б.И.Түзелбаев, А.А.Жакупов, Бітірушілер жұмысының экономикалық бөлімін орындауға арналған әдістемелік нұсқаулар, Алматы 2010, -5-19 б.Жандаулетова, Ф. Р. Охрана труда : учебник для вузов / Ф.Р. Жандаулетова, Т.Е. Хакимжанов, Т.С. Санатова; МОН РК, НАО АУЭС. - Алматы : АУЭС, 2019. - 399 с.

Жоба бойынша жобаның бөлімдеріне қатысты белгіленген кеңесшілер

Бөлімдері	Кеңесшілері	Мерзімі	Қолы
Негізгі бөлім	Темырканова Э.К.	10.05.2020	
Есептеу техникасын қолдану	Темырканова Э.К.	10.05.2020	
Экономика	Түзелбаев Б.И.	28.04.2020	
Өміртіршілік қауіпсіздік бөлімі	Жандаулетова Ф.Р.	6.05.2020	
Нормабақылау	Накисбекова Б.Р.	14.06.2020	

Диплом жобасын дайындау

КЕСТЕСІ

Бөлімдердің атауы, әзірленетін мәселелердің тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Кіріспе	23.02.2020	орындалды
Бағдарламалық өнімнің қолданылу аймағы	02.02.2020	орындалды
Чат боттардың қарапайым қосымшалардан артықшылығы	13.03.2020	орындалды
Чат боттардың негізгі қызметтерін нақтылау	21.03.2020	орындалды
Жобаны іске асыруға арналған бағдарламалық жасақтаманы таңдау	28.03.2020	орындалды
Бағдарламалық өнімді құру деңгейлері	6.04.2020	орындалды
Мақсаттар мен міндеттерді нақтылау	13.04.2020	орындалды
Нейрондық желі тақырыбының жаңашылдығы	21.04.2020	орындалды
Жасанды нейрондық желілер құрылымын зерттеу	21.04.2020	орындалды
Нейрондық желі артықшылықтары	24.04.2020	орындалды
Қолайлы нейрожелі архитектурасын таңдау	2.05.2020	орындалды
Бағдарламалық өнімді әзірлеу	2.05.2020	орындалды
Өміртіршілік қауіпсіздік бөлімі	3.05.2020	орындалды
Есептеу бөлімі	3.05.2020	орындалды
Бағдарламалық жасақтаманы (БЖ) әзірлеумен байланысты дипломдық жұмыстың техникалық-экономикалық негіздемесі	23.04.2020	орындалды
Бағдарламалық өнімнің бағасын анықтау	23.04.2020	орындалды
Қорытынды	6.05.2020	орындалды

Тапсырманың берілген уақыты «20» ақпан 2020 ж.

Кафедра меңгерушісі \_\_\_\_\_ (Темырканова Эльвира Кадылбековна)  
(қолы) (Т.А.Ж.)

Жобаның ғылыми жетекшісі \_\_\_\_\_ (Темырканова Эльвира Кадылбековна)  
(қолы) (Т.А.Ж.)

Орындалатын тапсырманы қабылдаған студент \_\_\_\_\_ (Қанатбаева Сындара Нұралықызы)  
(қолы) (Т.А.Ж.)

## **Андатпа**

Дипломдық жобаның тақырыбы «Python бағдарламалық қамтамасыздандыру көмегімен нейрожеліні құру». Бұл жұмыстың мақсаты: интернет дүкендері мен анықтама қызметінде жұмыскерлердің уақытын үнемдеп, жұмыс сапасын көтеру мақсатында чат-бот құру. Бот пайдаланушылардан келіп түскен кез-келген сұраныстарға жауап беріп олармен байланыс орнатады. Алға қойған мақсатқа жету үшін келесідей технологиялар қолданылды:

- Бот алгоритмін құру үшін: NLTK кітапханасы;
  - Бот программасын жазу үшін: Python бағдарламалық қамтамасыздандыруы;
  - Ботты іске қосу үшін: WhatsApp немесе Telegram қосымшалары .
- Дипломдық жоба кіріспеден, 5 бөлімнен және қорытындыдан тұрады.

## **Аннотация**

Тема дипломного проекта: «Построение нейросети с помощью программного обеспечения Python». Целью данной работы является: создание чат-бота для экономия времени работников и для повышения качества работы в call-центрах и онлайн маркетов. Бот будет отвечать на все вопросы клиентов и устанавливать связь с ними. Для достижения поставленной цели использовались технологии:

- для создания алгоритма бота: библиотека NLTK;
- для программирования бота: программное обеспечение Python;
- для работы бота: приложения WhatsApp или Telegram .

В дипломный проект входит введение, 5 глав и итоговое заключение.

## **Annotation**

Theme of the graduation project: «The building a neural network by using Python software».

The purpose of this work is: creating a chatbot to save employees time and improve the quality of work in call centers and online markets. . The bot will answer all customer questions and establish communication with them. Technologies used to achieve the goal:

- to create an algorithm of the bot: the NLTK library;
- for bot programming: Python software;
- for the bot to work: WhatsApp or Telegram apps .

The graduation project includes an introduction, 5 chapters and a conclusion.

## Мазмұны

Кіріспе	7
1 Негізгі бөлім	8
1.1 Бағдарламалық өнімнің қолданылу аймағы	8
1.2 Чат боттардың қарапайым қосымшалардан артықшылығы	9
1.3 Чат боттардың негізгі қызметтерін нақтылау	10
1.4 Жобаны іске асыруға арналған бағдарламалық жасақтаманы таңдау	12
1.5 Бағдарламалық өнімді құру деңгейлері	16
1.6 Мақсаттар мен міндеттерді нақтылау	18
2 Бағдарламалық өнімді жобалау	18
2.1 Нейрондық желі тақырыбының жаңашылдығы	18
2.2 Жасанды нейрондық желілер құрылымын зерттеу	19
2.3 Нейрондық желі артықшылықтары	23
2.4 Нейрондық желілердің маңызды мәселелерді шешудегі орны	26
2.5 Қолайлы нейрожелі архитектурасын таңдау	27
3 Бағдарламалық өнімді әзірлеу	31
4 Өміртіршілік қауіпсіздік бөлімі	42
4.1 Еңбек шартын талдау	42
4.2 Есептеу бөлімі	48
5 Бағдарламалық жасақтаманы (БЖ) әзірлеумен байланысты дипломдық жұмыстың техникалық-экономикалық негіздемесі	54
5.1 Бағдарламалық жасақтаманы әзірлеудің еңбек салымы	54
5.2 Бағдарламалық жасақтаманы зерттеуге кеткен шығындарды есептеу	55
5.3 Бағдарламалық өнімнің бағасын анықтау	63
Қорытынды	66
Әдебиеттер тізімі	67
А қосымшасы	69
Б қосымшасы	70

## **Кіріспе**

Нейрондық желілер - жасанды интеллект жүйелерін дамыту бағыттарының бірі. Басты мақсаты адамның жүйке жүйесінің жұмысын мүмкіндігінше мұқият модельдеу, дәлірек айтқанда, оның қателіктерді бақылап және түзету қабілеті. Бұл кез-келген нейрондық желінің басты ерекшелігі - ол әрдайым аз қателіктер жіберіп, алдыңғы тәжірибе негізінде өз бетінше білім алып, әрекет ете алады.

Нейрондық желілер әр түрлі бағыттарда қолданылады және оларды пайдалану дамыған қолдау жүйесінде бәсекеге қол жеткізуге мүмкіндік береді. Осылайша, нейрондық желілер қазіргі әлемде әртүрлі қызмет салаларында жоғары сұранысқа ие.

Дипломдық жұмыстың мақсаты: өндірісте үлкен көлемдегі ақпаратты өңдеуге қолданылатын тез әрі сапалы жасайтын жасанды желі негізінде чат бот құрып, оның жұмысын анықтау. Бұл көптеген салалардағы мамандар үшін шешім қабылдау процесін жеңілдетуге, жұмыс уақытын тиімді пайдалануға, қызмет көрсету сапасын жоғарлатуға және объектінің даму кезеңдерін болжауға мүмкіндік береді.

Дипломдық жұмыста негізгі міндеттер анықталды:

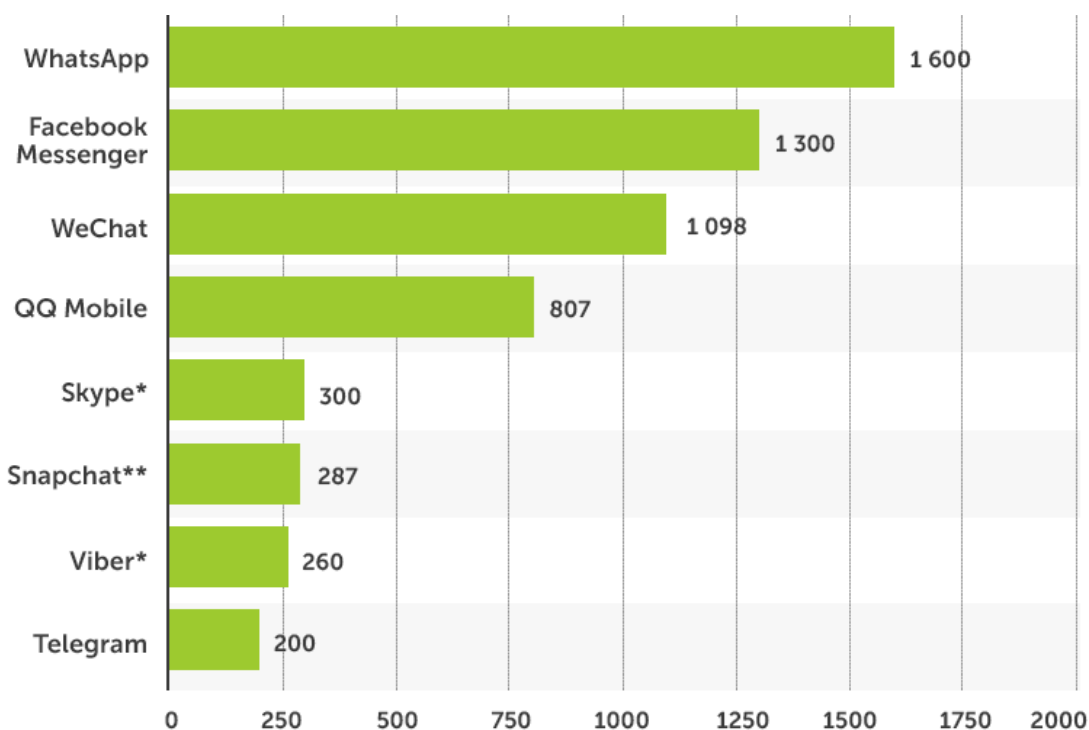
1. Нейрондық желілердің артықшылықтары мен кемшіліктерін анықтау.
2. Телекоммуникация саласындағы нейрондық желілердің үлгілерін қарастыру.
3. Python бағдарламалық жасақтамасының артықшылықтарын зерттеу
4. Нейрондық желілердің алгоритмдерінің қолданылуын тәжірибеде көрсету.
5. Чат бот құру алгоритмін жасап, оның жұмысының тиімділігін дәлелдеу.

Дипломдық жұмыстың нәтижесінде Python бағдарламасы арқылы ыңғайлы әрі сапалы түрде қызмет ететін нейрондық желі негізінде жұмыс жасайтын чат бот құрылды. Жұмыста іске асырылған жұмыстың негізгі принципі қарапайымдылық пен қол жетімділік қағидасы болды. Зерттеу әдісі - бұл қолданыстағы нейрожелілерді бақылау және талдау. Зерттеудің ғылыми жаңалығы мен теориялық маңыздылығы – қазіргі таңда жоғары сұранысқа ие чат бот құрылды. Зерттеудің практикалық маңыздылығы алынған нәтижелер негізінде нейрожеліні құру өлшемдері мен әдістері зерттелетіндігінде.

## 1.Негізгі бөлім

### 1.1 Бағдарламалық өнімнің қолданылу аймағы

Соңғы он жылда мобильді құрылғылар ұйымдардың тұтынушылармен және қызметкерлермен өзара іс-қимылының қуатты арнасына айналды. Facebook Messenger, WhatsApp және SMS сияқты арналар арқылы хабар алмасу тұтынушылар байланысының негізгі түрі болып табылады. Бүкіл әлемде 4,1 миллиардтан астам адам жедел хабар алмасу жүйесін пайдаланады (1.1 сурет). Олардың пайдаланушы базасы әлеуметтік желілерге қарағанда әлдеқайда жылдам өсті. Бұл арналардың маңызды артықшылығы-шынайы уақыт режимінде экран арқылы немесе push хабарламалары арқылы қол жеткізілетін сұхбаттасушының жедел жауап беруі. Адамдар өмірде жедел хабар алмасуға үйреніп, компьютерден де осындай тез жауап күтеді және сол үйреншікті арналар арқылы олармен байланысуды қалайды. Клиент-серверлік қосымшаларды ауыстыруға келген браузерлер сияқты, бұл арналар қосымшалардың функцияларын, атап айтқанда сол браузерлерді алмастыра отырып, өзіне алады. Бұл ұйымдарға қажетті ауқымда осындай арналар арқылы қарым-қатынасты автоматтандыруға көмектесетін жасанды интеллекті бар инновациялық чат-боттарды әзірлеуді қажет етеді.



1.1 сурет – Тұтынушылардың ай сайынғы белсенділік көлемі, млн



Чат-бот (виртуалды әңгімелесуші немесе жай бот) — бұл компьютерлік бағдарлама, ол тірі әңгімелесушіні имитациялап, пайдаланушы адаммен сөйлеседі (көбінесе Интернет арқылы). Зияткерлік боттардың тез таралуы бірнеше факторлардың үйлесуімен байланысты. Біріншіден, пайдаланушыларға қосымшаларды жүктеу және оларды орнату жайсыздық алып келеді. Қосымшаларды күнделікті пайдалану адамдарды шаршатады. Екіншіден, адамдар жаппай хабар қызметіне өтті және үнемі чаттарда отырады. Осылайша, қолданбаның орнына қолданушы құрылғысына орнатылған бағдарламада жұмыс жасайтын жүйені құруға болады. Чат-боттар жасанды интеллект саласындағы жетістіктердің арқасында көп сұранысқа ие байланыс құралы болып табылады. Компьютерлік оқытудың қуатты алгоритмдерінің негізінде құрылған компьютер (чат-бот) оператордың көмегенсіз қолданушымен диалог құруға қабілетті.

## **1.2 Чат боттардың қарапайым қосымшалардан артықшылығы**

Қазіргі заман талаптарына сай үнемі техникалық қолдауды, жүктеуді және дизайн әзірлеуді қажет ететін қосымшалармен салыстырғанда чат-бот үнемі ұтады. Олар "жинақы" түрде жасалғанымен қарапайым қолданбадан гөрі функциялардың кең жиынтығын қамтиды. Клиенттер өздері чат-ботпен байланысқа баратыны өте маңызды, оларды осы жаңа технологияға жүгінуге итермелейтін күш бұл тірі қарым-қатынас иллюзиясының болу факторы. Өйткені кез келген адам оның сұраныстарын естігенді ғана емес, тыңдап білгенді де ұнатады. Бот клиенттің жұмыстың көп бөлігін жасайтын жеке көмекшісі ретінде болады. Бұл клиенттермен байланысудың жаңа тәсілі. Осы боттар арқылы қолданушы сұраныстарына жедел жауап беру, қолдау шығындарын қысқартады және компанияны неғұрлым жоғары деңгейге шығаратын шешім болып табылады.

Чат-боттар қосымшалармен салыстырғанда бірнеше жоғары мүмкіндіктерге ие:

- Басты артықшылықтардың бірі клиенттерге қосымша бағдарламаларды жүктеудің қажеті жоқ, олар жай ғана оларға ыңғайлы мессенджерді ашады және чат-ботына жазады. Олар барлық платформаларда (telegram, what ' s app, Facebook, Viber, Вконтакте) жұмыс істейді.

- Жұмыс жасауға кедергі келтіретін адами факторларды шегеріп, 24/7 сағат бойы жұмыс істеуге қабілетті. Чат-ботпен жұмыс жағдайында қарым-қатынас психологиясы маңызды рөл атқарады, клиенттер оны шын мәнінде тыңдап тұрғанын және шын мәнінде көмектесуді қалайтынын түсінеді, бұл сондай-ақ үлкен артықшылығы болып табылады.

- Пайдаланушылар мәзірдің қажетті функциясын іздеуге уақыт жұмсамайды, ал жай ғана оларды қызықтыратын ақпаратпен "код" немесе

сөзді жазады және негізгі сөздерді өңдегеннен кейін бот ағымдағы сұранымға дұрыс нәтиже береді.

- Клиенттердің кез келген санаты үшін қол жетімді .
- Қосымшада бар функциялардан басқа, чат-бот сұрақтарға жауап бере отырып онымен қалай жұмыс істейтінін өзі көрсетеді.
- Арзан әзірлеу, кез келген табыс деңгейі бар компанияларға чат-ботты құруға мүмкіндік береді.
- Кәсіпорынға әрдайым соңғы үрдісте болуға және қатардан қалмауға мүмкіндік береді.
- Ыңғайлылық. Google мәліметтері бойынша, пайдаланушылардың 65% - ы кәсіпорындарға қоңырау шалмай, олармен хат жазу арқылы байланысқанды қалайды.
- Навигацияны жеңілдету. Чат-боттар сұраныс бойынша сайттан қажетті материалдарды таба алады. Бұл ретте олар дәстүрлі іздеуден гөрі, сөз қорын тануға қабілетті, сондықтан ыңғайлы.
- Ақпаратты бейімдеу. Ботпен хат алмасу келушіге қажет мәліметтердің дәл көлемін береді. Сайттың беттерінде мәтінді оқу міндетті емес – ақпараттық артық жүктеме жағдайында бұл үлкен артықшылық болып саналады.
- 24/7 қолдау. Виртуалды көмекші FAQ жауап режимінде пайдаланушыларды қолдауды жүзеге асыра алады. "Тірі" мамандарға қарағанда, ол тәуліктің кез келген уақытында хабарласу үшін қол жетімді.

### **1.3 Чат боттардың атқаратын қызметтеріне қойылатын талаптар**

Чат боттар төмендегідей негізгі қызметтерді атқырау мақсатында құрылады:

1. Колл-орталықтарының немесе техникалық қолдау операторларының барлығы клиенттермен телефон, пошта немесе мессенджерлер арқылы сөйлесіп үлгермей жатырған жағдайда және жүктеме өте үлкен болған жағдайда адам орнын алмастырады;

2. Компания жұмысында автоматтандыруды талап ететін менеджерлер жұмысының тиімділігін арттыратын қарапайым, жиі қайталанатын бизнес-процестер болған кезде;

3. Егер компания қызметтерін тәулік бойы, демалыссыз және мерекесіз пайдаланса, сауатты бапталған чат-бот колл-орталықтың үлкен штатында шығындарды қысқартуға мүмкіндік береді, стандартты мәселелерді өз бетінше шешеді, ал күрделі жағдайлар тірі операторларға қайта бағыттайды;

4. Компанияның ассортименті техникалық күрделі тауарлардан құралса, оларды таңдау үшін кеңес қажет жағдайда;

5.Операторларға тұрақты қайталанатын қарапайым тапсырыстар (бағасын білу, түсін және т. б. тапсырыс) түскенде;

6.Тауарлар каталогының құрылымы түсінікті болған жағдайда, тауарларды санаттарға және кіші санаттарға оңай бөлуге, ілеспе тауарлар мен аксессуарларды ұсынуға қолданылады;

7. Қызметкерлерге қосымша көмекші ретінде олардың уақытын үнемдеп, жұмыс сапасын жоғарлатады.

Чат боттар өнеркәсіпте төмендегідей мәселелер кедергі келтірген жағдайда көмекке келеді :

- Сайт қызметтерін тәулік бойы, 24\7 және демалыссыз пайдаланса;
- Мақсатты аудитория-жас (40 жасқа дейін);
- Ассортиментте кеңес алуды қажет ететін тауарлар көп болса;
- Операторларда сұраныстардың көп үлесі тақырып бойынша қайталанса;

- Операторларда үлкен жүктеме түсіп және осы мәселеге байланысты штатты кеңейту қажет болған жағдайда.

### **1.3.1 Сенімділік пен тиімділікке қойылатын талаптар**

Чат-ботта қарапайым әрі көрнекі пайдаланушы интерфейсі болуы тиіс. Ол ақпараттық технологиялармен қолдану саласында арнайы білімі жоқ пайдаланушыға жұмыс істеуге мүмкіндік беруі тиіс.

Чат боттың тұрақты жұмыс істеуін қамтамасыз ету үшін :

- оның әр түрлі браузерлермен үйлесімділігін тексеруді жүзеге асыру;
- операциялық жүйелердің әртүрлі кодтарымен оның үйлесімділігін тексеруді жүзеге асыру;
- оның экран рұқсаттарымен үйлесімділігін тексеруді жүзеге асыру;
- сессияда, деректер базасында сақталған ақпараттан тұратын арнайы файлдарда деректерді сенімді сақтау үшін;
- сервер жұмысындағы іркіліс нәтижесінде жоғалған ақпаратты қалпына келтіруді қамтамасыз ету.

Чат-боттың тиімді жұмыс істеуі үшін келесі шарттар қажет:

- чат- бот сақталатын қосымша жұмысының сапасы;
- қосымша іркіліс нәтижесінде жоғалған ақпаратты қалпына келтіруді қамтамасыз ету;
- енгізілген сұраныстарды қайтару немесе жою мүмкіндігі, сондай-ақ бастапқы кіру деректеріне көшу мүмкіндігі.

Чат-бот жұмыс қабілеттілігіне оңай тексерілуі тиіс, оның дұрыс жұмысының алдын-ала белгілі нәтижелері бар арнайы тесттер дайындалуы тиіс. Тексеруші құрал чат боттың қолданушылармен жұмысын тексеруді – пайдаланушының кіруін, сұранысын, мысал бойынша іздеуді және жаңа хабарламаларды қосуды, қиыншылық туған жағдайда операторға жүгіну

шапшандығын, сауалнаманың жұмыс қабілеттілігін тексеруді – оның дұрыс емес деректерді енгізуге реакциясын, деректерді өзгерту және енгізуді, келуді есептегішті тексеруді көздеуі тиіс.

### **1.3.2 Кіріс және шығыс деректерін ұйымдастыруға қойылатын талаптар**

Чат ботты пайдаланушы қосымшасы терезесінде көрсету үшін пайдаланушы мен сервер арасында кіріс және шығыс деректерімен алмасу қажет – бұл процесс жеткізу деп аталады.

Чат ботты пайдаланушыға жеткізу оны құру сияқты маңызды. Оның танымалдығы оның келушінің іс-әрекетіне жауап беру уақытына байланысты, бұл өз кезегінде соңғы пайдаланушының оған көрінісін білдіреді. Көптеген дизайнерлер жылдамдық қажеттілігін сезінеді. Тіпті бұл жағдайда, чат бот жүктеу жылдамдығының төмендеуін түсіндіру үшін, дизайнерлер әдетте бірнеше аспектілерге назар аударады: чат боттың графикалық файлдарының мөлшері немесе соңғы пайдаланушының қосылу жылдамдығы. Кешіктірудің нақты себептері айқын емес болуы мүмкін. Жүктеу жылдамдығын айқындануы мүмкін көптеген факторларға байланысты, мысалы желі: трафик түрі, қосымша сапасы және оның мазмұны. Дизайнерлер чат боттың жеткізудің барлық аспектілерін анықтауы тиіс, өйткені соңғы пайдаланушы ақпаратты берудің жеке компоненттерін бөле алмайды, ботты бірыңғай жүйе ретінде қарастырады.

Барлығы қалыпты жағдайда да, чат ботпен жұмыс істеу ұзаққа созылып, көп уақыт кетуі мүмкін. Бот-мастер әрқашан пайдаланушы сұрақтарына жауап табады: қызмет көрсету, деректерде жоқ сауалдарды өңдеу және қалпына келтіру, сондай-ақ жауаптың қол жетімділігін тексеру. Сауалға жауап алу процесі шамамен былайша өтеді:

- пайдаланушы өзіне керекті сауалды енгізеді;
- жүйеде сауалды түсіну қалыптасады;
- оған жауап ізделеді;
- егер жауап деректерде табылса, құралдарымен беріледі;
- сервер сұрау алады және біраз уақыттан кейін жауап береді;
- бот жауапты талдайды, деректерді көрсетеді немесе сақтайды;
- егер жауапта басқа нысандар болса, процесс қайталанады.

### **1.4 Жобаны іске асыруға арналған бағдарламалық жасақтаманы таңдау**

Бұл бөлім дипломдық жобаны әзірлеу кезінде қандай технологиялардың қолданылғаны, олардың негізгі артықшылықтары мен кемшіліктері туралы баяндайды. Сондай-ақ, осы технологиялардың неге таңдалғаны және қосымшадағы қандай бөлігі үшін олар жауапты екені түсіндіріледі.

Қазіргі уақытта байқалып отырған дербес есептеу техникасының қарқынды дамуына байланысты бағдарламалау тілдеріне қойылатын талаптардың біртіндеп өзгеруі орын алуда. Жеке компьютерлердің өсіп келе жатқан қуаты түсіндірілетін бағдарламаларды орындаудың жеткілікті жылдамдығын қамтамасыз ете бастайтындықтан, түсіндірілетін тілдер үлкен рөл атқара бастайды. Ал компиляцияланатын бағдарламалау тілдерінің бірден-бір маңызды артықшылығы олар жасайтын жоғары жылдамдықты код болып табылады. Бағдарламаны орындау жылдамдығы сыни шамада болмаса, ең дұрыс таңдау қарапайым және икемді бағдарламалау құралы ретінде түсіндірілетін тіл болады.

Осыған байланысты, белгілі бір қызығушылық 90-жылдардың басында Гвидо ван Россумның жасаған Python бағдарламалаудың салыстырмалы жаңа тілін қарастыру болып табылады.

Бұл тілдің ерекшеліктері :

- кіру шегі өте төмен, бір күнн оқығаннан кейін қарапайым бағдарламалар жазуға болады;
- аз мөлшерде құрылымды минималды тіл жинағы;
- қысқаша код;
- орама-бағдарламаларды жасау үшін өте қолайлы, Си-кітапханаларды импорттау қолданылады;
- көптеген құрылғыларды қамтиды : CPython (негізгі ); Jython (JVM үшін); IronPython( CLR); PyPy;
- математикалық есептеулерді өте жақсы қолдау (NumPy, SciPy кітапханалары);
- табиғи тілдерді өңдеу үшін қолданылады (NLTK);
- дамыған web-фреймворктардың үлкен саны (Django, TurboGear, CherryPy, Flask).

Басқа тілдердің әсері:

Салыстырмалы түрде кеш пайда болған Python көптеген бағдарламалау тілдерінің әсерінен құрылды:

- ABC-операторларды топтастыру үшін шегіністер, жоғары деңгейлі деректер құрылымы (map) (Python нақты ABC жобалау кезінде жіберілген қателерді түзету әрекеті ретінде құрылды);
- Modula-3-пакеттер, Модульдер, try және except бірге else пайдалану, функция аргументтері (сондай-ақ Common Lisp әсер етті);
- C, C++ - кейбір синтаксистік конструкциялары (Гвидо ван Россум өзі жазады-ол Python с-программистерге бей тудырмау үшін C ең қарама-қайшы емес конструкциясын пайдаланды);
- Smalltalk-объектілі-бағытталған бағдарламалау;
- Lisp-функционалдық бағдарламалаудың жеке ерекшеліктері (lambda, map, reduce, filter және басқалар));

- Fortran-массивтердің кесінділері, кешенді арифметика;
  - Miranda-тізімдік өрнектер;
  - Java-модульдері logging, unittest, threading (түпнұсқа модульдің мүмкіндіктерінің бөлігі іске асырылмаған), xml.sax стандартты кітапхана, ерекшеліктерді өңдеу кезінде finally және except ортақ пайдалану, декораторлар үшін пайдалану;
  - Icon-генераторлар.
- Python басқа мүмкіндіктерінің көп бөлігі (мысалы, бастапқы кодты байт-компиляция) бұрын басқа тілдерде іске асырылған.

#### **1.4.1 Python бағдарламасының тандалу себептері**

##### **1.Бағдарламалық қамтамасыздандыру сапасы:**

Көптеген адамдар үшін Python тілінің негізгі артықшылығы бағдарламалау тілдері әлеміндегі басқа құралдардан ерекшеленетін ыңғайлы, анық және жоғары сапада болып табылады. Python тіліндегі бағдарламалық код оңай оқылады, демек, оны бірнеше рет пайдалану және қызмет көрсету басқа сценарий тілдерінде бағдарламалық кодты пайдаланудан әлдеқайда оңай орындалады. Python тілінде бағдарламалық кодты ресімдеудің біркелкілігі оны құруға қатыспаған адамдар үшін де оның түсінігін жеңілдетеді. Сонымен қатар, Python объектілі-бағытталған бағдарламалау (ООП) болып табылатын бағдарламалық кодты көп рет пайдаланудың ең заманауи механизмдерін қолдайды.

##### **2.Жоғары даму жылдамдығы:**

C, C++ және Java, Python сияқты компиляцияланған немесе қатаң типтелген тілдермен салыстырғанда әзірлеушінің еңбек өнімділігін бірнеше есеге арттырады. Python тілінде бағдарламалық код көлемі әдетте C++ немесе Java тілінде эквивалентті бағдарламалық кодтың үштен бір бөлігін немесе тіпті бесінші бөлігін құрайды. Бұл пернетақтадан аз енгізу көлемін білдіреді, аз уақыт жөндеу және аз еңбек шығыны сүйемелдеу. Сонымен қатар, Python тіліндегі бағдарламалар кейбір басқа бағдарламалау тілдерінде қажетті компиляция мен байланыстырудың ұзақ кезеңдерінен өтіп, бірден іске қосылады, бұл программист еңбегінің өнімділігін арттырады.

##### **3.Бағдарламалардың төзімділігі:**

Python тіліндегі бағдарламалардың көп бөлігі барлық негізгі платформаларда өзгеріссіз орындалады. Linux операциялық жүйесінен Windows-ке бағдарламалық кодты тасымалдау әдетте бір машинадан екіншісіне бағдарлама файлдарын қарапайым көшіру болып табылады. Сонымен қатар, Python тасымалданатын графикалық интерфейстерді, деректер қорына қол жеткізу бағдарламаларын, веб-қосымшаларды және басқа да көптеген бағдарламаларды құру бойынша көптеген мүмкіндіктерді ұсынады. Python тілінде бағдарламаларды іске қосу және каталогтарды өңдеу

әдісін қоса алғанда, операциялық жүйелердің интерфейстері де тасымалданатын тәсілмен іске асырылған.

#### 4.Қолдау кітапханалары:

Python құрамында стандартты кітапхана ретінде белгілі жинақталған және тасымалданатын функционалдық мүмкіндіктер көп келеді. Бұл кітапхана үлгі бойынша мәтінді іздеуден бастап желілік функцияларға дейін қолданбалы бағдарламаларда талап етілетін көптеген мүмкіндіктерді ұсынады. Сонымен қатар, Python сіздің өз кітапханаларыңыздың есебінен де, бөгде әзірлеушілер құрған кітапханалардың есебінен де кеңейтуге жол береді. Үшінші тарап әзірлемелерінің ішінен веб-сайттарды құру құралдарын, математикалық есептеулерді бағдарламалау, дәйекті портқа кіру, ойын бағдарламаларын әзірлеу және т.б. атауға болады. Мысалы, NumPy кеңейтімі Matlab математикалық есептеулерді бағдарламалау жүйесінің еркін және қуатты баламасы ретінде орналасады.

#### 5.Компоненттерді біріктіру:

Python сценарийлері әр түрлі интеграция механизмдерінің арқасында қосымшаның басқа бөліктерімен өзара әрекеттесуі мүмкін. Бұл интеграция бағдарламалық өнімдердің функционалдық мүмкіндіктерін теңшеу және кеңейту үшін Python пайдалануға мүмкіндік береді. Қазіргі таңда Python тіліндегі бағдарламалық код C/C++ тілінде кітапханалардан функцияларды шақыруға, C/C++ тілінде жазылған бағдарламалардан өзі шақыртуға, Java тілінде бағдарламалық компоненттермен интегралдауға, COM және .NET сияқты платформалармен өзара әрекеттесуге және SOAP, XML-RPC және CORBA сияқты хаттамалардың көмегімен тізбектелген порт немесе желі арқылы деректер алмасуды жүргізуге мүмкіндік береді.

Python функционалдық мүмкіндіктері тұрғысынан гибрид деп атауға болады. Оның құралдық құралдары сценарийлердің дәстүрлі тілдері (Tcl, Scheme және Perl сияқты) мен бағдарламалық жүйелерді әзірлеу тілдері (C, C++ және Java сияқты) арасында ауқымға қойылады. Python сценарий тілінің қарапайымдылығы мен еркіндігін және әдетте компиляцияланатын тілдерде табуға болатын қуатты қамтамасыз етеді. Басқа сценарий тілдерінің мүмкіндігін асыра отырып, мұндай комбинация Python ірі масштабты жобаларды әзірлеудің ыңғайлы құралы болып табылады. Төменде Python арсеналында бар негізгі мүмкіндіктер тізімі:

##### 1.Динамикалық типтеу:

Python бағдарлама кодында ұзақ және күрделі хабарландыруларды жазу қажет емес, бағдарламада пайдаланылатын нысандардың түрлерін өзі қадағалайды. Шын мәнінде, Python тілінде әдетте түрі мен айнымалыларды жариялау қажеттілігі жоқ. Python тіліндегі бағдарламалық код деректер түрлерінің рамкаларымен қысылмаған болғандықтан, ол автоматты түрде нысандардың толық ауқымын өңдей алады.

## 2. Жадты автоматты басқару:

Python автоматты түрде объектілерге жадты бөледі және оны ("қоқыс жинау") босатады. Көптеген нысандар қажет болған жағдайда жад көлемін ұлғайтуы және азайтуы мүмкін.

## 3. Модульдік бағдарламалау:

Python ірі жүйелерін құру үшін модульдер, сыныптар және ерекшеліктер сияқты мүмкіндіктерді ұсынады. Олар жүйені құрауыштарға бөлуге, көп рет пайдаланылатын бағдарламалық кодты құру үшін қолдануға және пайда болған оқиғалар мен қателерді талғамды өңдеуге мүмкіндік береді.

## 4. Салынған нысандар түрлері:

Python бағдарламалау тіліне тән ерекшеліктер түрінде тізімдер, сөздіктер және жолдар сияқты ең типтік деректер құрылымын ұсынады. Бұл түрлер жоғары икемділік пен ыңғайлылықпен ерекшеленеді. Мысалы, кірістірілген нысандар қажет болған жағдайда кеңейтілуі және қысылуы мүмкін, күрделі құрылымды деректерді ұсыну үшін бір-бірімен біріктірілуі мүмкін.

## 5. Кірістірілген құралдар:

Python құрамында осы нысандардың барлық типтерімен жұмыс істеу үшін, конкатенация (коллекцияларды біріктіру), кесінділерді алу (коллекциялардың бөліктерін алу), сұрыптау, бейнелеу және т.б. сияқты операцияларды қоса алғанда, қуатты және стандартты құралдар бар.

## 6. Кітапхана жинақтамасы:

Python құрамына неғұрлым тар міндеттерді орындау үшін, сондай - ақ қажет болуы мүмкін барлық іс жүзінде қолдау көрсететін кітапханалық құралдардың үлкен топтамасы кіреді-тұрақты өрнектерді пайдалана отырып іздеуден желіде жұмыс істеуге дейін. Python тілінің кітапханалық құралдары - бұл операциялардың көп бөлігі орындалатын орын.

## 7. Бөгде әзірлеушілердің жинақтамалары:

Python-бұл ашық бағдарламалық өнім, сондықтан әзірлеушілер өздерінің алдын ала компиляцияланған тапсырмаларды қолдау құралдарын жасай алады, оларды ішкі құралдармен шешу мүмкін емес.

# 1.5 Бағдарламалық өнімді құру деңгейлері

## 1. Боттың негізгі мақсаттары мен міндеттерін қою:

Біздің кәсіпорынның әрбір клиенті чат бот түрлеріндегі айырмашылықтарды көре алмайды, оның мақсаттары мен міндеттерін өз бетінше анықтай алады. Боттың клиентке қандай мақсатта керектігін, ол қандай міндеттерді шешуі тиіс екендігін: компанияның Интернеттегі өкілдігі функциясын орындау немесе тауар мен қызметтерді сату, компанияның сапалы жарнамалық жарнамасы болу немесе жас фирма үшін стартап болу анықтаймыз. Осының бәрі тапсырыс берушімен келісіледі. Ол өзіне чат



боттың не екенін, оның компанияларына немен көмектесетінін түсінуі тиіс. Міндеттер қойылғаннан кейін мақсатты аудитория анықталады;

### 2. Техникалық тапсырманы құру, пысықтау (ТТ):

Техникалық тапсырманы әзірлеу оны құратын әрбір маман үшін сүйенетін негіз болып саналады. Сондықтан ТТ құруға тапсырыс берушінің қатысуы қажет. Ол клиент келісім беріп, құжатқа қол қойғанға дейін ескертіледі және өңделеді. Осыдан кейін ғана жоба құру кезеңіне өтеді. ТТ қалыптастыру және талқылау кезеңінде боттың құрылымы, оның навигациясы, қолдану аймағы және т. б. әзірленеді.

### 3. Чат боттың дизайн макетін жасау:

Чат ботты құру үшін барлық кезеңдері маңызды болып саналады, бірақ әрлендіру негізгілерінің бірі. Өйткені дизайн-бұл келушінің бірінші кезекте көріп, оны бағалайды және бетте қалуға немесе оны жабуға шешім қабылдайды.

Дизайнер техникалық тапсырмаға негізделеді, кнопкаларды, баннерлерді және басқа да графикалық элементтерді салады. Басқаша айтқанда, ботты әзірлеудің бірінші кезеңінде жасалған прототип эстетикалық сыртқы түрін алады, яғни тапсырыс беруші таңдаған түстерде жасалады. Егер компанияның корпоративтік стилі болса, онда дизайн оған сәйкес әзірленеді. Немесе алдымен фирмалық стиль, содан кейін оның негізінде дизайн жасалуы мүмкін. Дизайнердің әр бетті емес, веб-дизайн үрдістерін пайдалана отырып, бірнеше негізгі үлгілердің дизайнын бейнелейді. Дайын макет клиентке беріледі және өзінің мақұлдауын күтеді. Егер пысықтау қажет болса, дизайнер оларды орындайды және қайтадан тапсырыс берушіге макетін көрсетеді. Дизайн ол бекітілгенге дейін пысықталады.

### 4. Бағдарламалау:

HTML тілінің көмегімен дайын дизайнды жұмыс жобасына аударады. Ресурс серпінді болады, барлық түймелер жұмыс істейді. Ресурс барлық байланыс жүйелерінде дұрыс көрсетіледі. Осы кезеңде CSS стильдері жасалады.

### 5. Мазмұнды толтыру:

Беттерді графикалық және ақпараттық мазмұнмен толтыру бейне, фото, мәтін және келуші көре алатын немесе оқи алатын басқа да ақпарат орналастырылады. Беттер ішкі оңтайландырудан өтеді.

## **1.6 Негізгі мақсаттар мен міндеттерді талдау**

Мақсатқа қол жеткізу бір сәттік іс-шара болып табылмайды. Мақсатқа қол жеткізу мәні бойынша өзара байланысты және бірыңғай мақсатқа қол жеткізу үшін бағытталған міндеттердің жиынтығын шешуді білдіреді.

Әрбір міндет жобада қандай да бір заттарды зерттеуге немесе әзірлеуге бағытталған. Осылайша, міндеттер-белгілі бір мақсатқа қол жеткізілетін өзіндік "кірпіштер".

Төменде қойылған мақсатқа жету үшін іске асыру қажет дипломдық жобаның мақсаты мен міндеттері келтірілген.

Дипломдық жобаның мақсаты-кәсіпқойлар жұмысының өнімділігін арттыруға мүмкіндік беретін бағдарламалық өнімді әзірлеу.

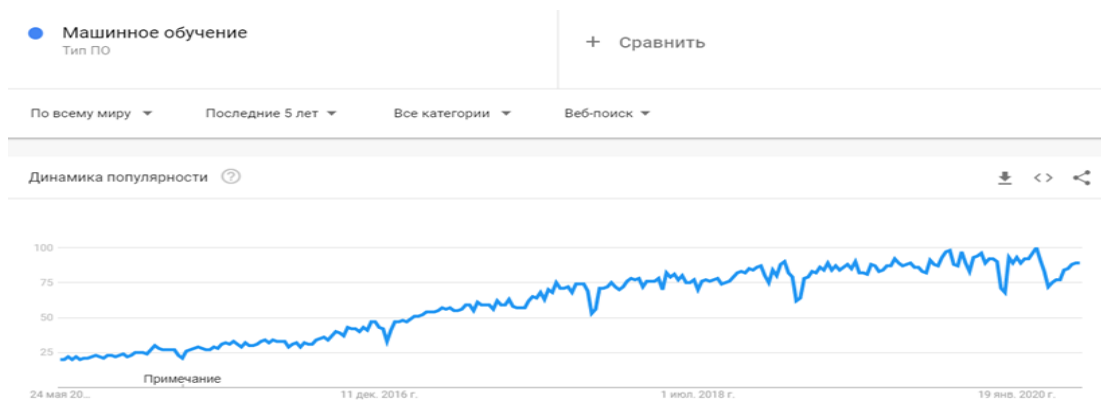
Дипломдық жобаның негізгі міндеттері:

- пәндік саланы зерттеу;
- бағдарламалық өнімге қойылатын негізгі талаптарды анықтау;
- бағдарламалық өнімді әзірлеу үшін технологияларды таңдау және негіздеу;
- пайдаланушы веб-интерфейсті әзірлеу;
- пайдаланушылардың деректерін сақтау үшін деректер базасын жобалау;
- сервер жағында функционалдық қосу: тіркеу, аутентификация, чат, жоспарлаушы;
- әзірленетін өнімнің экономикалық орындылығын негіздеу;
- іске асырылатын жоба шеңберінде еңбек жағдайларын жақсарту бойынша іс-шараларды ұсыну.

## **2 Бағдарламалық өнімді жобалау**

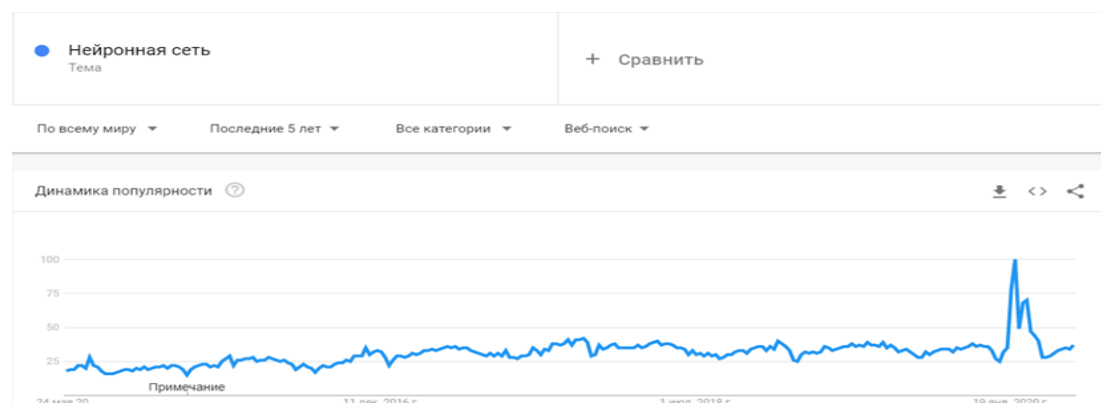
### **2.1 Нейрондық желі тақырыбының жаңашылдығы**

Машиналық оқыту, ақылды алгоритмдер және жасанды интеллект тақырыптары қазіргі кезде өте танымал және жоғары сұранысқа ие. Мұны әр түрлі IT-порталдардағы статистика растайды. Төмендегі графиктер Google Trends жүйесіндегі 2005 жылдан бастап осы уақытқа дейінгі деректерді көрсетеді (2.1 сурет).



2.1 сурет - Машиналық оқыту тақырыбының бүкіл әлем бойынша сұраныс графигі

“Нейрондық желілер” тақырыбының бүкіл әлем бойынша сұраныс графигі:



2.2 сурет - Нейрондық желілер тақырыбының бүкіл әлем бойынша сұраныс графигі

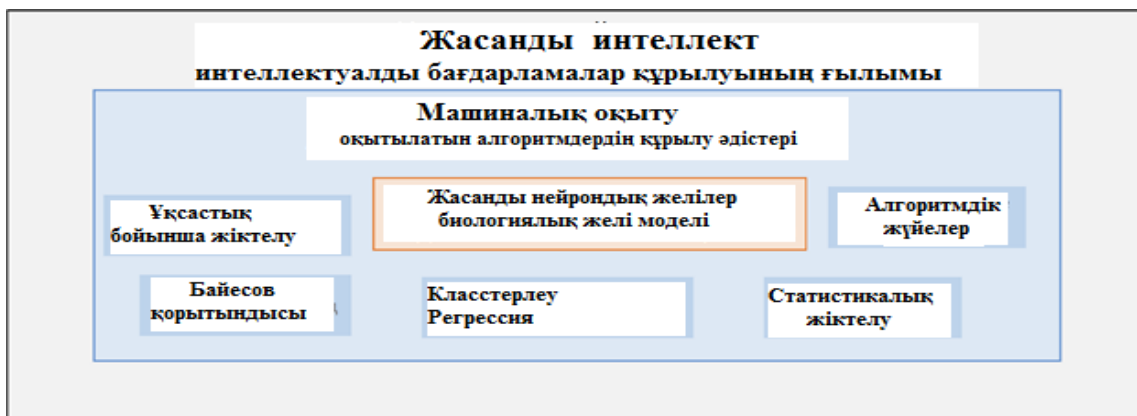
Осы статистикалар арқылы қазіргі таңда нейрондық желілер және оларды машиналық оқыту тақырыбы жоғары сұранысқа ие болып, дамып келе жатырғанын көре аламыз.

## 2.2 Жасанды нейрондық желі (ЖНЖ) құрылымын зерттеу

Жасанды нейрондық желілер— бұл биологиялық нейрондық желілерді ұйымдастыру және жұмыс істеу қағидаттарына негізделген бағдарламалық немесе аппараттық қамтамасыз етілген математикалық модельдер.

Нейрондық желілер сөздің әдеттегі мағынасында бағдарламаланбайды, олар оқытылады. Оқу қабілеті - нейрондық желілердің дәстүрлі алгоритмдерге қарағанда басты артықшылықтарының бірі. Техникалық тұрғыдан оқыту

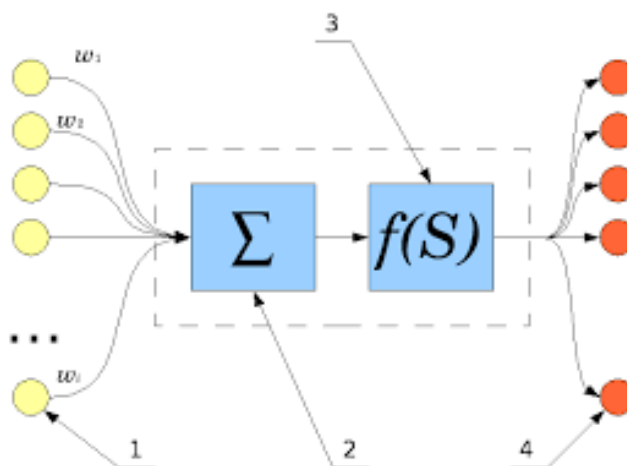
нейрондар арасындағы байланыс коэффициенттерін табудан тұрады. Оқу процесінде нейрондық желі кіріс пен шығудың күрделі байланысын анықтай алады, сонымен қатар жалпылауды жүзеге асырады (2.3 сурет).



2.3 сурет - Жасанды интеллект құрылымы

Машиналық оқыту (Machine learning) — оқытылатын алгоритмдер құрылымының әртүрлі әдістерін зерттейтін жасанды интеллект бөлімшесі. Машиналық оқыту — өте кең білім саласы. "Оқыту" сөзін әртүрлі анықтауға және әр жолы қызықты нәтижелер алуға болады. Алайда машиналық оқытудағы көптеген парадигмалар мен тәсілдердің арасындағы өте қызықты облыс — жасанды нейрондық желілер болып табылады. Машиналық оқыту тұрғысынан, нейрондық желі дегеніміз - үлгіні танудың, дискриминациялық талдаудың, кластерлеу әдістерінің және т.б. әдістерінің ерекше жағдайы.

Әрбір жасанды нейрондық жүйенің негізі салыстырмалы түрде қарапайым, көп жағдайда бір типті, ми нейрондарының жұмысына еліктейтін элементтерден (жасушалардан) тұрады (бұдан әрі - нейрон деп біз жасанды нейрон, жасанды нейрондық тордың жасушасын айтамыз).

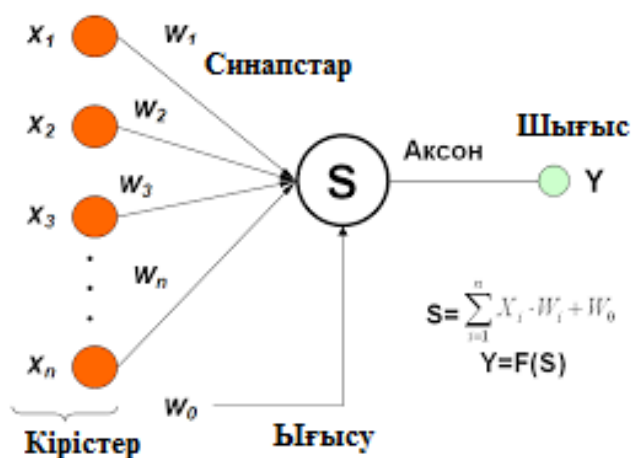


2.4 сурет - Жасанды нейрондық желі құрылымы

2.4 суретте көрсетілген нейрондық желі құрылымы осылардан тұрады:

1. Шығысына ақпараттың кіріс сигналдары енгізілетін нейрондар
2. Кіріс сигналдарының сумматоры
3. Беріліс функциясының есептегіші
4. Кірісіне ақпараттың шығыс сигналы берілетін нейрондар
5.  $w_i$ - кіріс сигналдарының салмағы

Сигналды түрлендіретін әрбір нейрон саны көп (миллиардтар) болғандықтан, кірісіндегі сигнал шығысында дұрыс сигналға түрленеді. Барлық нейрондар бір алгоритм бойынша жұмыс істейді. Басқаша айтқанда, кез келген нейронның ішінде сигнал әрдайым бірдей өңделеді. Бірақ олай болса ешқандай жеке түрлендіру бола алмайды. Соған сәйкес желінің де жауабы әрқашан бірдей болады. Шынында бұл сұрақтың жауабы нейрожелінің синапс деп аталатын бөлігінде орналасқан (2.5 сурет).



2.5 сурет - Нейрожелі моделі

Синапстар – сигнал күшейтілуі мен сөнуі жүзеге асырылатын бір нейронның шығысы мен екінші нейронның кірісінің түйісетін аймағы. Әрбір синапс синоптикалық қосылыстың мәні немесе салмағы  $w_i$ -мен сипатталады. Әрбір синапстың өзіндік ерекшелігі бар. Ол өзінен өтетін сигналды күшейтеді немесе әлсіретеді. Сонымен қатар, уақыт өте келе синапстар өзгеруі мүмкін, соған орай сигналдың өзгеру сипаты да түрленеді. Егер синапстардың параметрлеріне дұрыс таңдау жасалса, онда нейрондық желіден өтетін кіріс сигналы дұрыс шығыс сигналына айналады. Дәл осылай көптеген кіріс сигналдарының дұрыс шығыс шешіміне айналуы орындалады.

Әр нейронның ағымдағы күйі бар, оны әдетте оның кірістерінің өлшенген қосындысы ретінде анықтайды:

$$s = \sum_{i=1}^n x_i \cdot w_i, \quad (2.1)$$

Аксон – берілген сигналдың келесі нейронның синапстарына өтуін қамтамасыз ететін шығыс байланыс жолы.

Бұл нейрондық желі моделінен жасанды нейронның жұмысы туралы жалпы түсінік ала аламыз. Кіріс белгілі бір сигналдар санын алады, оны шешілетін мәселенің түріне байланысты таңдау критерийлері бойынша нейрондық желінің сәулетшісі алдын-ала анықтайды. Бұл сигналдар  $W$  салмағы бойынша көбейтіліп, жинақталады. Осыдан кейін активтендіру функциясы алынған мәліметтерді жауап түріне түрлендіреді. Әдетте бұл жауап  $[0; 1]$  немесе  $[-1; 1]$  ауқымында болады. Активация функциялары әртүрлі болады:

- сызықтық функция

$$f(x) = x, \quad (2.2)$$

- сигмоидты функция

$$f(x) = \frac{1}{1 + e^x}, \quad (2.3)$$

- гиперболалық тангенс

$$f(x) = \frac{e^{2x} - 1}{1 + e^{2x}}, \quad (2.4)$$

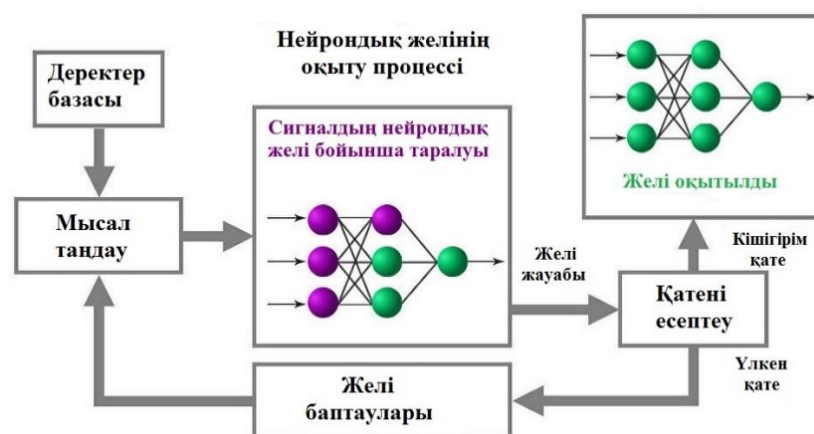
Жасанды нейрондық желілерде мұндай активтендіру функциясының алуан жиынтығы бар. Бірнеше нейрондарды біріктіру арқылы жасанды нейрондық желі жасалады.

## 2.3 Нейрондық желі артықшылықтары

Нейрондық желілер өздерінің күшін, біріншіден, параллельді ақпаратты өңдеуге, екіншіден, үйрену қабілетіне, яғни жалпылау құратынына сарқылтатыны анық. Жалпылау термині оқу процесінде кездеспеген мәліметтер негізінде нәтиже алуды қамтамасыз етеді. Бұл қасиеттер нейрондық желілерге қазіргі кезде мүмкін емес болып саналатын күрделі мәселелерді шешуге мүмкіндік береді. Алайда, іс жүзінде, автономды жұмыс кезінде нейрондық желілер дайын шешімдерді бере алмайды. Оларды күрделі жүйелерге біріктіру қажет.

Нейрондық желілердің дәстүрлі есептеу жүйелеріне қарағанда кейбір артықшылықтары бар.

- Үйрену қабілеті. Нейрондық желілердің негізгі артықшылықтарының бірі-олардың оқуы. Дәстүрлі алгоритмдерде жаңа деректердің пайда болуы оларды қайта есептеу қажеттілігін тудырады, бұл нейрондық желілерге тән емес – оқыту жүйесінің арқасында олар қойылған міндеттерді шешу үшін жаңа деректерді әр жолы қолдана отырып, әр түрлі өзгерістерге бейімделеді. Нейрондық желіні оқыту итерациялық сипатқа ие. Оқытудың әрбір қадамы цикл деп аталады. Осындай бір цикл кезінде оқу үлгісінен алынған мысалдарды негізге алатын синаптикалық коэффициенттерді есептеу болжанады. Осы іріктеменің құрамына өзіндік мәні мен шығыс параметрлеріне ие әртүрлі деректер кіреді. Нейрондық желі маман тағайындаған мысалдардарға сәйкес оқиды. Мұндай маманның жұмысы келесіден тұрады: ол кіріс параметрлерінің белгілі бір жиынтығын таңдап алады, содан кейін желіде алынған деректерді ерекше құрылымға түрлендіретін оқыту алгоритмін іске қосады (2.6 сурет).



2.6сурет – Нейрондық желілердің оқытылу процессі

- Жан-жақтылығы. Олар, өз кезегінде, деректерді талдау және өңдеу міндеттерінің кең шеңберін: бейнелерді тану және жіктеу, қаржы нарықтарын болжау, процестерді басқаруды автоматтандыру және т. б. шешуге қабілетті әрі қуатты құрал болып табылады, Әрбір міндет үшін классикалық шешу әдістері де бар, бірақ нейрожелінің артықшылығы дәстүрлі әдістердің көмегімен шешілетін тапсырманы қайта құрудың қажеттілігінің жоқтығында.

- Әрекет ету шапшандығы. Дәстүрлі алгоритмдерді пайдалану кезінде оның жұмысын бағалаудың негізгі өлшемдерінің бірі деректердің кіріс көлеміне байланысты орындау жылдамдығы болып табылады. Нейрондық желілерде кіріс параллельді шешімдердің бөліп таратылған желісіне түрленеді. Деректерді параллель өңдеу жүйесінің арқасында нейрондық желінің әрекет ету жылдамдығы жоғарлайды (2.7 сурет).

#### Биологиялық аналогия



#### Жасанды желілерде

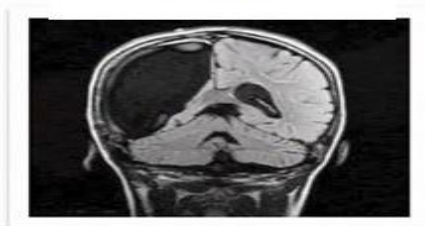


2.7 сурет – Әрекет ету шапшандылығы

- Істен шығуға тұрақтылығы. Аппараттық деңгейде іске асырылған нейрондық желінің өнімділігі қолайсыз жағдайларда төмендеуі мүмкін. Бірақ нейрондық желіде ақпаратты сақтаудың бөлінген сипатының арқасында жүйенің жұмыс қабілетін оның құрылымын күрделі зақымдау арқылы ғана айтарлықтай төмендетуге болады (2.8 сурет).

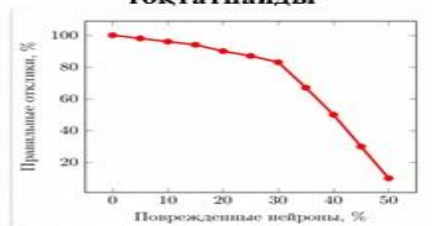
#### Биологиялық аналогия

Гемисферэктомия  
мидың жарты бөлігінің  
зақымдануы



#### Жасанды желілерде

Нейрондық желілер олардың  
элементтерінің көп бөлігі  
зақымданса да жұмысын  
тоқтатпайды



2.8 сурет – Жасанды желілердің жұмыс сапасы

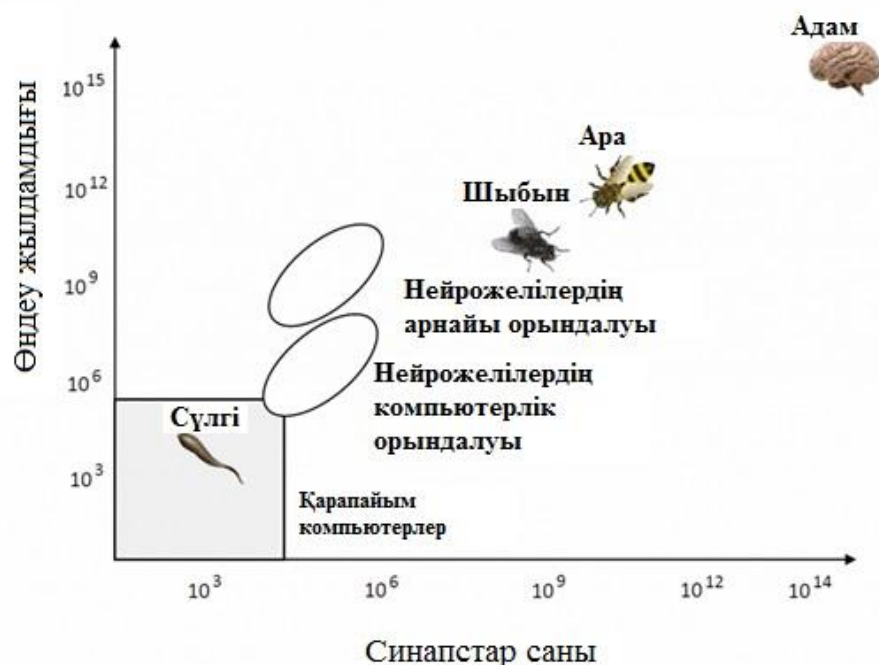


Сонымен қатар, нейрондық желілер кіріс деректерінде "шуылдарға" жоғары төзімділікке ие. Егер қандай да бір себеппен талдау үшін қойылған міндетті шешуге қатысы жоқ артық параметрлер алынса, онда нейрондық желі мұндай деректерді сүзеді. Жаттығудан кейін олар шулы деректер берілетін кіреберістерге назар аудармауға қабілетті (2.9 сурет).



2.9 сурет – Нейрондық желілердің сыртқы шуларды қабылдамау қабілеттілігі

Нейрожелілердің күрделілік деңгейін анықтау. Біз жасанды нейрондық желілерді құру үшін нейрондар санын азайтамыз. Бұл соманы қаншалықты азайта аламыз? Төменде суретке назар аударыңыз (2.10 сурет).



2.10 сурет – Нейрожелілердің жұмыс жасау деңгейі

Көлденең ось бойынша бізде синапстер саны бар (оны желінің күрделілігі ретінде қабылдаймыз). Тік ось желі бойынша жүйелердің жұмыс жылдамдығы белгіленген. Сол жақтағы төменгі сұр түсті шаршы-қарапайым компьютерлердің мүмкіндіктерін көрсетеді (бірақ нейрон желілер негізінде жасалған компьютерлер емес). Содан кейін екі эллипс орналасқан. Олар жасанды нейрондық желілерді қазіргі заманғы бағдарламалық түрінің жұмысын көрсетеді.

Нейрондық желілердің мүмкіндіктерінің адам миына жетуіне әлі де ұзақ уақытты қажет етеді. Сонымен қатар, мұнда логарифмдік шкала қолданылады. Бұл графикте диагональ бойынша өте аз жылжудың өзі синапстардың саны мен жұмыс жылдамдығының миллиондаған есе ұлғаюын білдіреді.

## 2.4 Нейрондық желілердің маңызды мәселелерді шешудегі орны

Жалпы айтқанда, бізге түрлі мәселелерді шешу керек. Компьютерлік бағдарламалар-мәселені шешу тәсілдерінің бірі. Ал олар шексіз көп. Әрбір мәселені талқылаудың орнына олардың жіктелуін енгізуге болады (2.11 сурет). Қазіргі заманда маңызды мәселелерді шартты күрделілік бойынша (қарапайым, орташа және жоғары күрделілік) және оның шешу жолдарын анықтау бойынша (дәл білеміз, шамамен білеміз, мүлдем білмейміз) ажырататын боламыз.



2.11 сурет – Нейрондық желілердің қолданылу аймағы

Шешу жолдары белгілі шағын/орташа күрделілік мәселелері:

- қарапайым теңдеуді шешу;

- экранда бағдарлама терезесін шығару;
- құжатты принтерде басып шығару.

Мұндай есептер қарапайым компьютерлік бағдарламалардың көмегімен шешіледі.

Шағын/орташа күрделілік мәселелері, олардың шешімі ішінара белгілі:

- қарапайым болжау;
- қателіктерді есептеу;
- теңдеулерді жуықтап шешу.

Мұндай міндеттерді арнайы статистикалық әдістердің көмегімен шешуге болады.

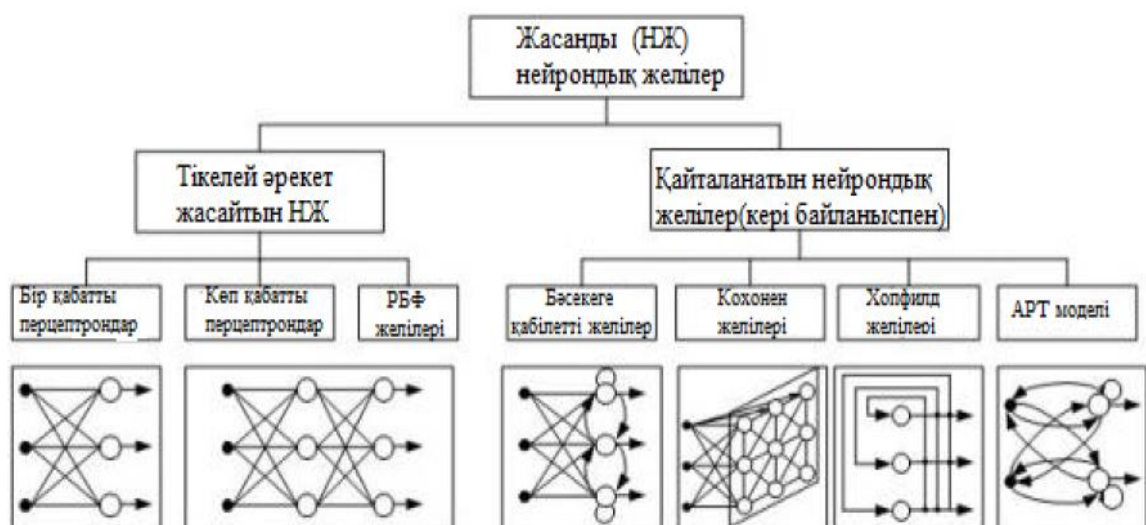
Шешімдері түсініксіз жоғары күрделілік мәселелері:

- бейнелерді тану;
- сөйлеуді тану;
- күрделі болжамдар.

Мысалы, белгілі бір алгоритм жіне дайындалған мысалдар бойынша қызметкерге келіп түсетін әртүрлі сауалдарға тез әрі дұрыс жауап беретін чат бот құру керек. Бұл мәселенің шешімі ретінде нейрондық желілер көмекке келеді. Биологиялық нейротейлердің үлгісі бола отырып, олар осындай міндеттерді шеше алады. Нейрондық желілер шешу алгоритмі белгісіз мәселелерді шешу үшін қолданылады. Бұл қасиете нейрожелілерді қабілеті және оларды танымал етті. Нейрожелілерді ойындар ойнауға, хаттарға жауап беруге, дауысты тануға, бағаның өсуін/құлдырауын болжауға үйретуге болады.

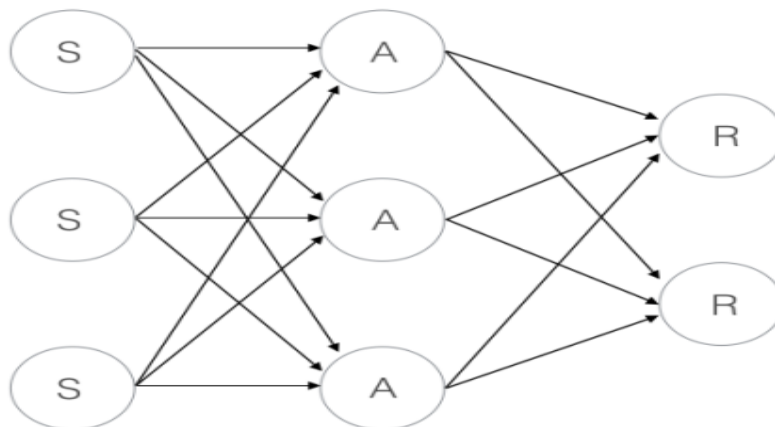
## 2.5 Қолайлы нейрожелі архитектурасын таңдау

Нейрондық желілердің классификациясын келесі суретте көрсетілген:



2.12 сурет - Нейрожелілердің жіктелуі

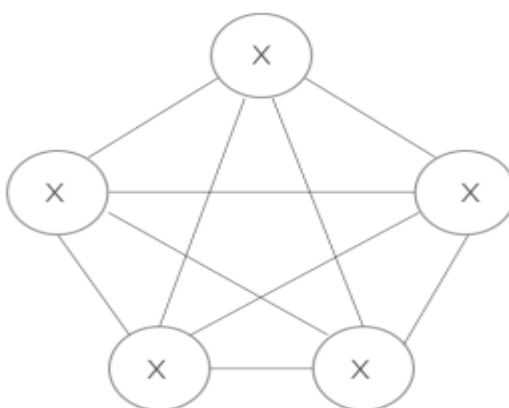
1.Перцептрон - әртүрлі типтегі сенсорлар жүйесі: сенсорлар (S), ассоциативті (A) және реактивті (R) . Перцептронның жалпы көрінісі 2.12 суретте көрсетілген.



2.13 сурет - Перцептрон архитектурасы

Әр нейронның 2.13-суретте көрсетілген формасы бар екенін ескере отырып, S элемент қабаты қандай да бір түрде мәліметтерді өзгертпейді, тек оларды келесі қабатқа өткізеді, сондықтан оны алып тастауға болады деп айта аламыз. Тіпті мұндай модель жіктеу мәселесін шеше алады және негізгі логикалық элементті қалыптастыра алады, мысалы немесе элементі.

2.Хопфилд желісі – желі 2.14-суретте көрсетілген толық жалғанған график түрінде болады.

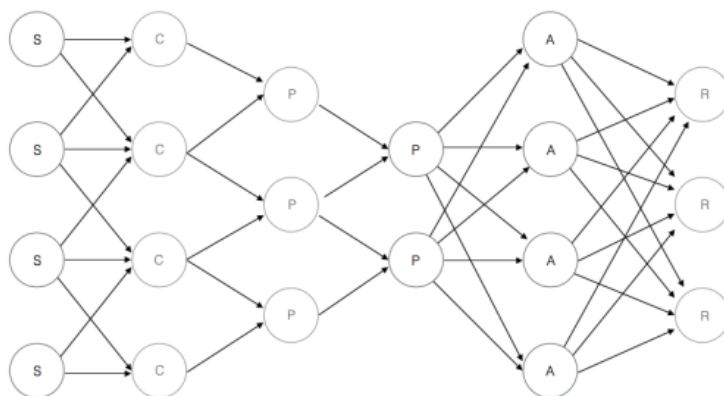


2.14 сурет - Хопфилд желісі архитектурасы

Бұл желіде әр нейрон үш формада болады. Жаттығу алдында әр нейрон сенсорлы болады, жаттығу жүріп жатқанда бәрі ассоциативті, жаттығудан кейін олар жауап береді. Бұл желі тұрақты күй қабылданғаннан кейін жауап

береді. Жүйе соңына дейін тұрақтанбайды, өйткені желінің жалпы энергиясы оқудың әр итерациясымен төмендейді. Нейрондар, әдетте, қатар үйренеді.

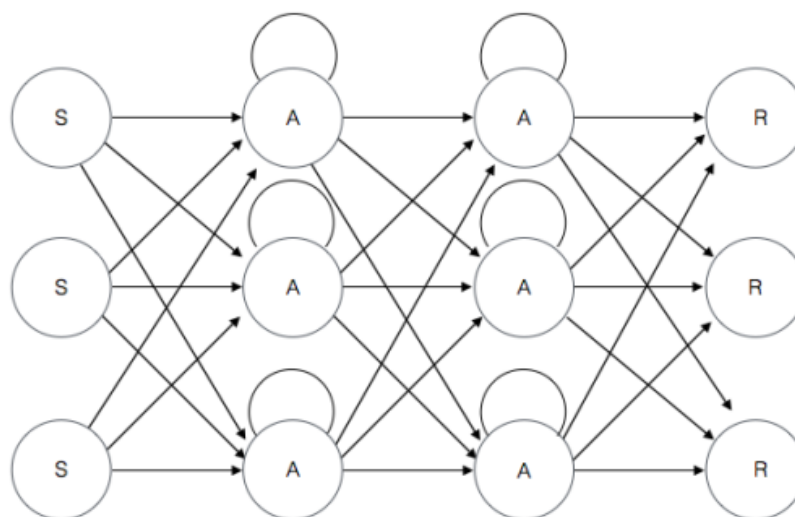
3.Жинақталған нейрондық желі басқалардан өте ерекшеленеді, көбінесе графикалық және аудио ақпаратпен жұмыс жасау үшін қолданылады. Мұндай желінің моделі 2.15-суретте көрсетілген. Бұл архитектура деректерді толығымен емес, фрагменттерде өңдейді, бірақ сонымен бірге мәліметтер фрагменттелмейді, бірақ бірізді іске қосылады. Содан кейін деректер одан әрі қабаттар арқылы жіберіледі. Шоғырлану қабаттарынан басқа (C) одақтық қабаттар қолданылады. Мысалы, дыбыстық тректен тек негізгі сәттер таңдалады: амплитудалық шырлар мен кідірістер.



2.15 сурет - Жинақталған нейрожелі архитектурасы

Бұл сәулет кескінді тану және жіктеу үшін ең қолайлысы. Оның көмегімен есептеулерді параллельдеу және нәтижесінде графикалық процессорларды қолдану мүмкін болады. Алайда, бұл архитектураның өзгермелі параметрлері көп болуы керек, мысалы қабаттар саны, әр қабаттағы ядролар саны, әр нейронның активтену функциялары және басқалары.

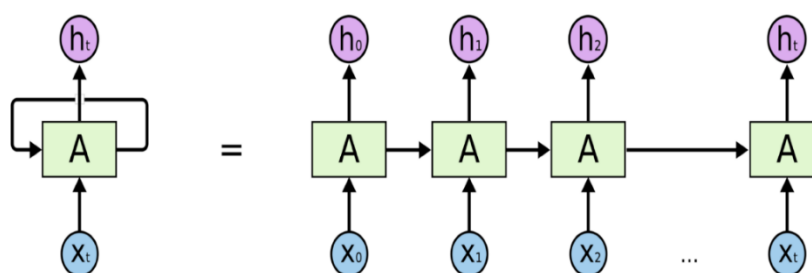
4.Қайталанатын нейрондық желі архитектурасы тікелей тарату желісіне ұқсас, бірақ уақыттың ауысуымен. Нейрондарға арналған ақпарат тек үстемдік етуші қабаттардан ғана емес, сонымен бірге өздері де басым итерациямен келеді. Бұл желі 2.16-суретте көрсетілген.



2.16 сурет - Қайталанатын нейрожелі

Өкінішке орай, бұл желілердегі ақпарат уақыт өте келе активтену функцияларына байланысты жылдамдықпен жоғалады. Бұл архитектураны әртүрлі жағдайларда қолдануға болады, өйткені кез-келген мәліметтерді дәйектілік түрінде беруге болады.

5. Рекуррентті нейрондық желілер - табиғи тілді өңдеуде қолданылатын танымал модельдер (NLP). Біріншіден, олар мәтіндерде қаншалықты жиі кездесетіні негізінде сөйлемдерді бағалайды. Бұл бізге грамматикалық және семантикалық түзету өлшемін береді. Мұндай модельдер машиналық аудармада қолданылады. Екіншіден, тіл үлгілері жаңа мәтінді жасайды (2.17 сурет). Адамның сұраққа жауап беруінің модельдерді оқыту сол адамның жауабына ұқсас жаңа мәтінді жасауға мүмкіндік береді. Рекурренттік нейрондық желі тікелей нейрондық желіге қарағанда рекурсивті ЖНЖ нұсқасы болып табылады, онда нейрондар арасындағы байланыс — бағытталған циклдар. Соңғы шығыс ақпараты ағымдағы кіріске ғана емес, сондай-ақ алдыңғы қадамдағы нейронның жағдайына байланысты. Мұндай жады пайдаланушыларға NLP міндеттерін шешуге мүмкіндік береді: қолжазба мәтінін немесе сөйлеуді тану.



2.17 сурет - Реккурентті нейрондық желілер



Әдетте мұндай желі яғни тікелей тарату желісінде ешқандай жады болмайды. Яғни, біз осы желіге белгілі бір суретті жібердік, ол одан бір бейне тапты делік, енді келесі суретті бергеннен кейін оның алдыңғы сурет туралы есінде ештеңе болмайды. Ол суреттер тізбегін өзара байланыстырмайды. Ал біз, мысалы, кейбір мәтінді алайық, табиғи тілдегі мәтін, кітапшаның қандай да бір тарауы, онда бұл мәтін сөзден тұрады және бұл сөздер кейбір бірізділікті құрайды. Бұл сөзді желіге жібердік, содан кейін келесі сөзге жібердік, бірақ ол алдыңғы сөз туралы ұмытпау, ол туралы есте сақтау керек және оның болғанын есіне түсіріп, алдыңғы тарихты ескере отырып, әрбір келесі сөзді талдауы қажет. Дәл осындай жүйелерде рекурренттік нейрожелілер қолданылады. Суретте көрініп тұрғандай, бізде кіріс белгілерінен кейбір нейрондық желінің шығуына жол бар, біз сондай-ақ алдыңғы уақыт қадамдарының ішкі қабаттардағы осы сигналдарды ескереміз, яғни нейрожелі ақпаратты есте сақтай алады.

#### Қорытынды

Қорытындылай келе, нейрондық желіні таңдау негізінен шешілетін мәселенің түріне (сақтау, болжау, кластерлеу) және енгізілген мәліметтерге (өлшемдер, негізгі факторлар) негізделген. Сақтау қабілетіне ие және деректерді бірізділікті өңдей алатындықтан жобаның мақсаты ретінде алынған чатботты құру мақсатында нейрожелінің рекуррентті түрі таңдап алынды. Бұл желі түрі бізге жұмыс барысында керекті барлық қасиеттермен қамтылған.

### 3 Бағдарламалық өнімді әзірлеу

Клиент-серверлік қосымшаның екі бөлігі бар: клиент және сервер. Бұл қолданба негізінен ұялы телефонға орнатылған қосымша арқылы пайда болады және пайдаланушымен өзара әрекеттеседі.

Клиент, бұл пайдаланушыға көрсетілетін қосымшаның бөлігі және ол пайдаланушымен визуалды түрде әрекеттеседі. Бұл жақта HTML, CSS және JavaScript, Python сияқты таңбалар, стильдер және бағдарламалау тілдері жұмыс істейді.

Қосымшаның серверлік бөлігі жеке визуалды көрініске ие емес және веб-браузер арқылы пайдаланушымен өзара әрекеттеседі. Бұл барлық әрекеттер браузерден мыңдаған шақырым қашықтықта, сондай-ақ тікелей жақын жерде, тіпті бір машинаға дейін орналасуы мүмкін арнайы компьютердегі серверде орындалады. Серверде әдетте деректер базасы орналасқан және Java, PHP, C#, Python және т.б. сияқты тілдерді қолданады.

Бағдарламалық өнімді әзірлеудің басында осы жобаны әзірлеу үшін барлық қажетті құралдарды қамтитын жобаны құру қажет.

Ол үшін Python бағдарламасын ашып және жобаны құру үшін барлық қажетті параметрлерді таңдау қажет. Жоба құрастырушысы Python

бағдарламасы, Java бағдарламалау тілі, Spring Boot 2.1.4 нұсқасы таңдалды. Бұдан әрі жоба үшін керекті метадеректер көрсетілген.

Бот санатының инициализациясын жазамыз:

"Wabot.py" файлын құрамыз ,онда біздің бот санатын сипаттаймыз. Бізге осы кітапханаларды импорттау қажет:

```
import json
import requests
import datetime
```

### 3.1 сурет - Кітапханалар тізімін шығару командасы

Json кітапханасы Json пішімін өңдеуге жауап береді. Requests-API сайтымен байланысу үшін қажет.

XML-мен салыстырғанда оның қысқа болуы есебінен JSON пішімі күрделі құрылымдарды сериализациялау үшін неғұрлым қолайлы. Веб-қосымшаларда браузер мен сервер арасында (AJAX) және серверлер арасында (бағдарламалық HTTP жұптастыру) деректер алмасу үшін қолданылады.

JSON пішімі JavaScript тілінің синтаксисінің ішкі жиыны болғандықтан, ол кірістірілген функциямен тез қайта іске асырылады.

Кейін WABot санатын жасаймыз:

```
class WABot():
def __init__(self, json):
    self.json = json
    self.dict_messages = json['messages']
    self.APIUrl = 'https://eu41.chat-api.com/instance12345/'
    self.token = 'abcdefg'
```

### 3.2 сурет - WABot санатын жасау командасы

Онда кіріс хабарламалары туралы ақпаратты қамтитын json қабылдайтын санат құрылымын сипаттаймыз (оны Webhook қабылдайды және санатқа жібереді). Қабылданатын json қалай көрінетінін білу үшін жеке кабинетте ұсынылатын тестілеудің ыңғайлы бөліміне кіреміз. Онда сұрауларды және Webhook тестілеуге болады.

Мұнда біз мән класының атрибуттарына ие боламыз. Dict\_messages-біз қабылдаған json файлдағы хабарламадан ақпаратты қамтитын сөздік.

```
self.json = json
self.dict_messages = json['messages']
```

### 3.3 сурет - Ақпаратты қамтитын сөздік командасы



Json құрылымын "Webhook тексеру" элементінде тестілеу бөлімінен көруге болады. Тексеруді іске қосып, чатқа хабарламалар жібереміз. Экранда webhook жіберілетін json командасы шығады.

```
self.apiUrl = 'https://eu41.chat-api.com/instance12345/'  
self.token = 'abcdefg'
```

### 3.4 сурет - Json құрылымын тексеру командасы

APIUrl және token мәліметтерін орнату қажет, оларды API-ге сұранысты қалыптастыру үшін пайдаланамыз.

Бот функционалын жазамыз:

Жұмыс істеу үшін API сұрауларын жіберу қажет. Біздің параметрлерге сүйене отырып, сұрауларды қалыптастыратын және оларды жіберетін функцияны жазамыз. API бағдарлама ұсынатын функционалдылықты анықтайды (модуль, кітапхана), бұл ретте API дәл осы функционалдылықтың іске асырылуына мүмкіндік береді.

Егер бағдарламаны (модуль, кітапхана) қара жәшік ретінде қарастырса, онда API — бұл осы жәшікті пайдаланушыға қол жетімді және оны пайдалануға мүмкіндік беретін "тұтқалар".

Бағдарламалық компоненттер API арқылы бір-бірімен өзара әрекеттеседі. Бұл ретте, әдетте компоненттер иерархияны құрайды — жоғары деңгейлі компоненттер төмен деңгейлі компоненттердің API-ін пайдаланады, ал өз кезегінде, неғұрлым төмен деңгейлі компоненттердің API-ін пайдаланады.

Мұндай қағидат бойынша интернет арқылы деректерді беру хаттамалары жасалған. Стандартты хаттамалар стегі (OSI желілік моделі) 7 деңгейден тұрады (физикалық бит беру деңгейінен HTTP және IMAP хаттамаларына ұқсас қосымшалар хаттамаларының деңгейіне дейін). Әрбір деңгей алдыңғы ("төменде") деректер беру деңгейінің функционалдығын пайдаланады және өз кезегінде келесі ("жоғары") деңгейдің қажетті функционалдығын ұсынады.

Функциялар мен сыныптар кітапханалары API сигналдар мен функциялар семантикасы сипаттамасынан тұрады.

```
def send_requests(self, method, data):  
    url = f"{self.apiUrl}{method}?token={self.token}"  
    headers = {'Content-type': 'application/json'}  
    answer = requests.post(url, data=json.dumps(data), headers=headers)  
    return answer.json()
```

### 3.5 сурет - API сұрауларын жіберу командасы

Send\_requests-екі параметрді қабылдайды: method және data.

Method - ChatAPI әдісі қандай болуы керек екенін анықтайды.

Data - жіберу үшін қажетті деректерді қамтиды.

Data - json қалыптастыратын, post әдісімен серверге жіберілетін деректер сөздігі. (Өйткені біздің бот функционалы үшін тек Post әдістері қажет.)

Осы кезеңде біз API-ге сұраныс жолын қалыптастырамыз:

```
url = f"{self.APIUrl}{method}?token={self.token}"
```

### 3.6 сурет - API сұраныс жолы

Бұл әдіс Бут жаңартуларды жіберетін вебхук URL мекенжайын орнату үшін қажет. Бұл мекен-жайға жаңартуды алған сайын JSON-да сериалданған update нысаны бар HTTPS POST жіберіледі. Серверге сұраныс сәтсіз болса, әрекет қалыпты рет қайталанады.

Біз әрдайым деректерді json пішімімен жіберетін болғандықтан, міндетті түрде header Content-Type нұсқап, оның Application/Json белгілеу керек:

```
headers = {'Content-type': 'application/json'}
```

### 3.7 сурет - Json форматын енгізу командасы

Енді requests арқылы толыққанды сұрау қалыптастырамыз. Post және api серверіне біздің деректерді жібереміз. Json.dump(data) – сөздікті data форматынан json форматына өзгертеді.

```
answer = requests.post(url, data=json.dumps(data), headers=headers)
return answer.json()
```

### 3.8 сурет - Серверлерге деректерді жіберу командасы

Json пішіміндегі сервер жауап функциясын қайтарады.

Хабарламаларды жіберу:

Send\_message-екі параметрді қабылдайды: chatId және text.

Chatid-хабарлама жіберу қажет чат ID

Text- хабарлама мәтіні

Енді біздің чатқа хабарлама жіберуге мүмкіндік беретін әдісті жазамыз.

```
def send_message(self, chatID, text):
    data = {"chatID" : chatID,
           "body" : text}
    answer = self.send_requests('sendMessage', data)
    return answer
```

### 3.9 сурет - Ботқа хабарлама жіберу командасы

"ChatId" - Id денесін қамтитын data сөздігін қалыптастырамыз, онда бізге қажетті мәтінмен хабарлама мен 'body' денесін жіберуді талап етеді.

```
data = {"chatID" : chatID,
        "body" : text}
```

### 3.10 сурет - Сөздік қалыптастыру командасы

Бұдан әрі біздің деректерді алдыңғы кезеңде жазған әдіске береміз:

```
answer = self.send_requests('sendMessage', data)
return answer
```

### 3.11 сурет - Деректерді жіберу командасы

Chat Api-ге хабар жіберу үшін "sendMessage" әдісі қолданылады, сондықтан оны біз data сөздігімен бірге параметр ретінде функцияларға жібереміз. Содан кейін сервердің жауабын қайтарамыз.

Қарсы алу

Welcome әдісін ботты "hi" командасы бойынша және жоқ команда енгізгенде шақыратын боламыз.

Chatid-хабарлама жіберуге қажетті чат Id

noWelcome- чатқа қандай мәтін жіберілетінін анықтайтын Булева айнымалысы: сәлемдесу немесе командалар тізімі.

```
def welcome(self, chatID, noWelcome = False):
    welcome_string = ''
    if (noWelcome == False):
        welcome_string = "WhatsApp Demo Bot Python\n"
    else:
        welcome_string = ""Incorrect command

Commands:
1. chatid - show ID of the current chat
2. time - show server time
3. me - show your nickname
4. file [format] - get a file. Available formats: doc/gif/jpg/png/pdf/mp3/mp4
5. ptt - get a voice message
6. geo - get a location
7. group - create a group with the bot""
    return self.send_message(chatID, welcome_string)
```

### 3.12 сурет - Сәлемдесу командасы

NoWelcome айнымалысына сүйене отырып, жолданатын мәтін ретінде send\_message функциясына жібереміз.

ChatID шығысы

```
def show_chat_id(self, chatID):  
    return self.send_message(chatID, f"Chat ID : {chatID}")
```

### 3.13 сурет - Жолданатын мәтін командасы

Уақытты шығару

```
def time(self, chatID):  
    t = datetime.datetime.now()  
    time = t.strftime('%d:%m:%Y')  
    return self.send_message(chatID, time)
```

### 3.14 сурет - Уақытты көрсету командасы

Me командасы бойынша сұхбаттасушының аты туралы ақпаратты шығарады.

```
def me(self, chatID, name):  
    return self.send_message(chatID, name)
```

### 3.15 сурет - Сұхбаттасушы туралы ақпарат беру командасы

File функциясы

Көрсетілген пішімді файлды диалогқа жібереді.

Chatid-хабарлама жіберу қажет чат Id

format-жіберу қажет файл пішімі. Барлық жіберілген файлдар серверде сақталады.

```
def file(self, chatID, format):  
    availableFiles = {'doc' : 'document.doc',  
                      'gif' : 'gifka.gif',  
                      'jpg' : 'jpgfile.jpg',  
                      'png' : 'pngfile.png',  
                      'pdf' : 'presentation.pdf',  
                      'mp4' : 'video.mp4',  
                      'mp3' : 'mp3file.mp3'}  
    if format in availableFiles.keys():  
        data = {  
            'chatId' : chatID,  
            'body': f'https://domain.com/Python/{availableFiles[format]}',  
            'filename' : availableFiles[format],  
            'caption' : f'Get your file {availableFiles[format]}'  
        }  
    return self.send_requests('sendFile', data)
```

### 3.16 сурет - Барлық деректерді сақтау командасы

Мұнда біз кілттер ретінде бізге қажетті форматтар, ал серверде жатқан және жөнелтуді күтетін файлдар атауларының мәні ретінде қамтылған сөздік қалыптастырдық:

```
availableFiles = {'doc' : 'document.doc',  
'gif' : 'gifka.gif',  
'jpg' : 'jpgfile.jpg',  
'png' : 'pngfile.png',  
'pdf' : 'presentation.pdf',  
'mp4' : 'video.mp4',  
'mp3' : 'mp3file.mp3'}
```

### 3.17 сурет - Қажетті форматтар мен сөздіктерді шығару командасы

Бұдан әрі пайдаланушы берген формат біздің сөздікте бар ма екенін тексереміз.

Егер бар болса, файлды жіберу сұрауын қалыптастырамыз, онда:

```
if format in availableFiles.keys():  
    data = {  
        'chatId' : chatID,  
        'body': f'https://domain.com/Python/{availableFiles[format]}',  
        'filename' : availableFiles[format],  
        'caption' : f'Get your file {availableFiles[format]}'  
    }  
    return self.send_requests('sendFile', data)
```

### 3.18 сурет - Мәтінді жіберу командасы

Chatid-хабарлама жіберу қажет чат Id

Body-жіберу қажет файлға тікелей сілтеме

Filename-файл аты

Caption-файлмен бірге жіберілетін мәтін

Send\_requests сұрауын "sendFile" параметрімен қалыптастырамыз және оған біздің деректерді жібереміз.

РТТ функциясы- дауыстық хабарды диалогқа жібереді

РТТ - екі жақты радиоинтерфейспен және бір уақытта тек бір бағытта сигнал беру мүмкіндігімен жартылай спектрлі дауыстық байланыс стандарты. Дауысты қабылдау және беру режимдерінің арасында ауысу үшін пайдаланушы радиоқұрылғыларда тиісті батырманы басу/жіберу қажет.

РТТ жылжымалы телефония үшін қолданылған ең ерте байланыс стандарттарының бірі болып табылады және шын мәнінде сымды телефон қызметінің операторы арқылы жанама түрде мобильді аналогы болды. Операторды шақыру үшін ұялы телефон пайдаланушысы құрылғыдағы РТТ (тангент) түймесін бірнеше секунд ішінде басып ұстап тұрды. Абонент түймені жібергеннен кейін оператор жауап беріп, абонент қоңырау шалған телефон нөмірін сұрайды.

```
def ptt(self, chatID):
    data = {
        "audio" : 'https://domain.com/Python/ptt.ogg',
        "chatId" : chatID }
    return self.send_requests('sendAudio', data)
```

### 3.19 сурет - Дыбыстық хабарды жіберу командасы

Деректер сөздігін қалыптастырамыз, мұнда:  
 Chatid-хабарлама жіберу қажет чат Id  
 audio-ogg форматындағы файлына тікелей сілтеме  
 Api-ге "sendAudio" әдісімен сұрау жібереміз.  
 Geo функциясы- координаттарды жібереді.

```
def geo(self, chatID):
    data = {
        "lat" : '51.51916',
        "lng" : '-0.139214',
        "address" : 'Баш адрес',
        "chatId" : chatID
    }
    answer = self.send_requests('sendLocation', data)
    return answer
```

### 3.20 сурет - Орналасу координаттарын жіберу командасы

Chatid-хабарлама жіберу қажет чат Id  
 lat-алдын ала берілген координаттар  
 lng-алдын ала берілген координаттар  
 address- сіздің мекен-жайыңыз немесе сізге қажетті жолы.  
 Сөздікті қалыптастырғаннан кейін API-ге " sendLocation" сұраныс жібереміз.

Group функциясы- бот орналасатын топты құрады

```
def group(self, author):
    phone = author.replace('@c.us', '')
    data = {
        "groupName" : 'Group with the bot Python',
        "phones" : phone,
        "messageText" : 'It is your group. Enjoy'
    }
    answer = self.send_requests('group', data)
    return answer
```

### 3.21 сурет - Боттың орналасу тобын құратын команда

author-webhook жіберген json денесі хабарламаны кім жібергені туралы ақпаратты қамтиды.

Бұл дене пайдаланушының нөмірі туралы ақпаратты қамтиды

```
data = {  
    "groupName" : 'Group with the bot Python',  
    "phones" : phone,  
    'messageText' : 'It is your group. Enjoy'}
```

### 3.22 сурет - Пайдаланушы нөмерін көрсету командасы

groupName-конференция құрылғаннан кейінгі аты

phones-конференция қатысушыларының телефондары

messageText-конференцияда бірінші хабар

Сұрауды'group' әдісімен жібереміз

Пайдаланушылардың сұрауларын өңдейміз:

Боттың бүкіл функционалын сипаттадық, енді ол командаға жауап беру және пайдаланушымен өзара іс-қимыл жасау үшін бот жұмысының логикасын ұйымдастыру талап етіледі. Бұл үшін тағы бір функцияны сипаттаймыз.

```
def processing(self):  
    if self.dict_messages != []:  
        for message in self.dict_messages:  
            text = message['body'].split()  
            if not message['fromMe']:  
                id = message['chatId']  
                if text[0].lower() == 'hi':  
                    return self.welcome(id)  
                elif text[0].lower() == 'time':  
                    return self.time(id)  
                elif text[0].lower() == 'chatid':  
                    return self.show_chat_id(id)  
                elif text[0].lower() == 'me':  
                    return self.me(id, message['senderName'])  
                elif text[0].lower() == 'file':  
                    return self.file(id, text[1])  
                elif text[0].lower() == 'ptt':  
                    return self.ptt(id)  
                elif text[0].lower() == 'geo':  
                    return self.geo(id)  
                elif text[0].lower() == 'group':  
                    return self.group(message['author'])  
            else:  
                return self.welcome(id, True)  
        else: return 'NoCommand'
```

### 3.23 сурет - Пайдаланушымен ақпарат алмасу командасы

Бұл функцияны біздің webhook бағдарламасында деректерді алатын сайын шақыратын боламыз.

Біз басында жасаған dict\_messages командасы барлық қабылдаған хабарлар сөздігін қамтиды. Бұл тексеру хабарламалары жоқ деректерді ажыратады. Өйткені webhook-қа хабарламасыз сұрау келуі мүмкін.

```
if self.dict_messages != []:
```

### 3.24 сурет - Хабарламаларды тексеру командасы

Шын мәнінде, бізге бір сұраныста бірнеше хабарлама келуі мүмкін, бот олардың барлығын өңдеуге тиіс. Ол үшін біз dict\_messages парағы қамтитын барлық сөздіктерді қарастырамыз.

```
for message in self.dict_messages:  
    text = message['body'].split()
```

### 3.25 сурет - Барлық сөздіктен іздеу командасы

Циклге кіргеннен кейін біз text айнымалысын жариялаймыз-ол біздің хабарламамыздағы сөздер парағы болады. Ол үшін біз кіріс хабарының мәтінін алу үшін ['body'] кілт арқылы message сөздігіне жүгінеміз және мәтінді сөздерге бөлуге мүмкіндік беретін split функциясын шақырамыз.

Одан әрі біз " fromMe " кілтіне жүгініп, онда орналасқан True немесе False командалары арқылы хабарламаның кімнен екенін тексереді.

Сұрауларды өңдеу үшін Flask серверін қолданамыз. Файлды құрып app.py, барлық қажетті кітапханаларды импорттаймыз.

```
from flask import Flask, request, jsonify  
from wabot import WABot  
import json  
  
app = Flask(__name__)  
  
@app.route('/', methods=['POST'])  
def home():  
    if request.method == 'POST':  
        bot = WABot(request.json)  
        return bot.processing()  
  
if __name__ == '__main__':  
    app.run()
```

### 3.26 сурет - Сұрауларды өңдеу командасы

Flask сыныбы болатын app айнымалысын инициализациялаймыз:



```
app = Flask(__name__)

@app.route('/', methods=['POST'])
def home():
    if request.method == 'POST':
        bot = WABot(request.json)
        return bot.processing()
```

### 3.27 сурет - Инициализациялау командасы

Және ол үшін app жолын жазамыз, route('/', methods = ['POST']). Бұл декоратор біздің home функциясы flask серверіне басты жолмен post сұрау арқылы жүгінген сайын шақырылатын болады дегенді білдіреді.

Серверге POST әдісінің көмегімен келгенін тексереміз. Біздің бот данасын жасап, оған json мәліметтерін жібереміз.

requests.json-біздің серверге жіберілген сұрау денесінен json файлдарды алуға мүмкіндік береді.

```
def home():
    if request.method == 'POST':
        bot = WABot(request.json)
        return bot.processing()
```

### 3.28 сурет - Керекті жауаптарды алу командасы

Енді біздің нысанасұрауларды өңдеуге жауап беретін bot processing әдісін шақырамыз .

Содан соң біздің серверді хостинг ботпен бірге жүктеп және webhook ретінде доменімізді көрсету керек. Әрбір кіріс хабары серверге келіп, деректер өңделетін болады.

#### Қорытынды

Жұмыс барысында орындалған басты міндет-нақты еңбек жағдайларында толық пайдаланылуы және оған тиесілі барлық функцияларды орындауы мүмкін қосымшаны жасау болып табылады.

Әзірлеудің қорытындысы бойынша пайдаланушылардың сауалдарына жылдам жауап беріп, жұмысшылардың уақытын тиімді пайдалануға мүмкіндік беретін функционалдық қосымша алынды. Пайдаланушылардың барлық деректері тек өздері қол жетімді және қосымшаны пайдалану процесі әрбір қызметкер тіркелгеннен кейін ғана іске қосылады.

## **4 Өміртіршілік қауіпсіздік бөлімі**

### **4.1 Еңбек шартын талдау**

#### **4.1.1 Жұмыс орынындағы еңбек жағдайының сипаттамасы**

Бұл бөлімде дипломдық жұмыстың тақырыбы “Python бағдарламалық қамтамасыздандыруы арқылы нейрожелінің құрылуы” .Дербес компьютерге жазылған Python бағдарламасы арқылы құрылған нейрожелінің телекоммуникация саласындағы қолданылуын қарастыру. Бұл бағдарлама арқылы құрылған нейрожелі телекоммуникациялық қызметтермен қамтамасыз етумен айналысатын ТОО “Мунайтелеком” компаниясының Ақтау қаласының орталығындағы кеңсеге орнатылады. Телекоммуникациядағы маңызды мәселенің бірі,тораптар арасында трафик жіберудің тиімді жолын табу нейрондық желілердің көмегімен шешіледі. Нейрондық желілер жаңа телекоммуникациялық желілерді жобалауда пайдаланылады.

Осы тапсырмаларды жүзеге асыру үшін бірнеше ақпараттық-бағдарламалық жасақтамалар мен құрылғылар пайдаланылды. Жұмыс екі бөліктен тұратын, оның бірінші бөлігі техникалық тапсырманы әзірлеу кезеңі, ал екінші кезеңге бағдарламалық жасақтама жатады. Нейрожеліні құру кезінде жұмыс қауіпсіздігінің барлық ережелері сақталды. Желіні құру компьютермен орындалды. Осыған байланысты өмір тіршілік қауіпсіздік бөлімінде,инженерге әсер ететін осы жағдайлар қарастырылды:

- жарықтандыру;
- жұмыс орнының эргономикасы;
- электрқауіпсіздік;
- өрт қауіпсіздігі.

Бұл нейрожелі телекоммуникациялық қызметтерді ұсынатын компаниялар үшін жобаланды . Еңбек процесі материалдық және өндірістік ортаның элементтері мен факторларының жиынтығымен сипатталатын өндірістік ортаның белгілі бір жағдайларында жүзеге асырылады. Жұмысқа 3 Asus F571GD i7 9750H компьютері және енгізу-шығару құралы (принтер,сканер) қолданылды. Жұмыс жабдығы,яғни компьютер осындай жағдайда оңтайлы жұмыс жасайды:

- жұмыс орны температурасы 0-ден 30-қа дейін;
- ылғалдылығы 5-тен 95% - ға дейін;
- қорек көзі: айнымалы ток-кернеуі 100-ден 220 В дейін, жиілігі 60/70Гц,ток 3 -6 А; тұрақты ток-кернеуі 45-ден 65 В дейін.

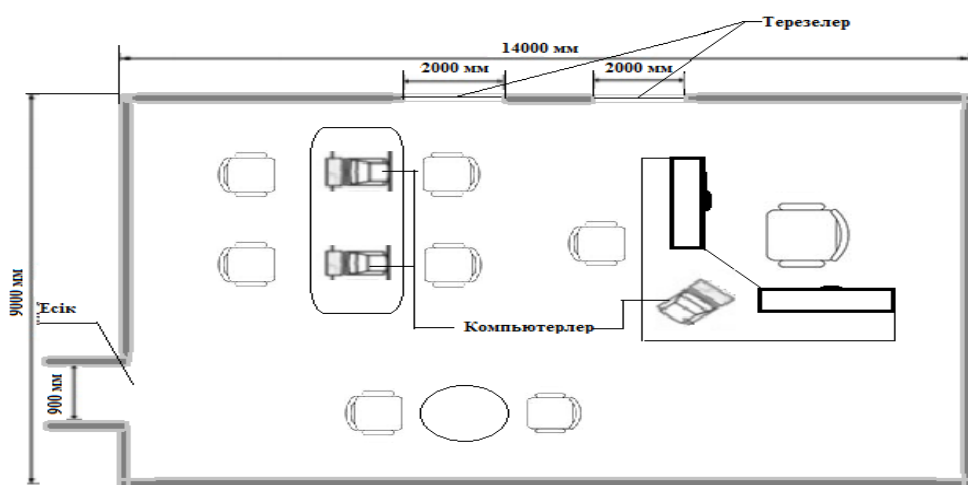
Компьютер орналасатын жұмыс орнының параметрлері:

- ұзындығы 14 м;
- ені – 7 м;

-биіктігі – 5 м.

Жұмыс орнының шу деңгейі 45 дБ, жарықтандыруы 400 лк. Жұмыс аймағының ауасының температурасы- 20 ° С, жылдамдығы - 0,6 м / с, ылғалдылығы - 50%.

Жұмыс орнындағы жабдықтың орналасу жоспары 4.1-суретте көрсетілген: кеңседе 3 қызметкер жұмыс істейді. ”Адамдарға әсер ететін физикалық факторлардың (компьютерлер мен бейне терминалдар) көздерімен жұмыс істеуге қойылатын санитарлық-эпидемиологиялық талаптар” бойынша бір адамға кем дегенде 6,8 м<sup>2</sup> ауданы берілген. Біздің жұмыс орнымыз 126 м<sup>2</sup>, оның ішінде 3 м<sup>2</sup> ауданын жабдықтар алып жатыр, яғни бір қызметкерге 41 м<sup>2</sup> бөлме ауданы келеді. Еңбекті қорғауды басқару жүйелері 2008 жылғы 21 мамырдағы № 249-НҚ [3] талаптарды қанағаттандырады.



4.1 сурет - Жұмыс орнындағы жабдықтың орналасу жоспары

#### 4.1.2 Жарықтандыру жүйесін талдау

Адамдардың қызмет етуіне жеткілікті табиғи және жасанды жарықтандыру атқарылатын жұмыстардың жоғары сапалылығын, қауіпсіздікті қамтамасыз етеді, еңбек жағдайларын жақсартып өнімділігін арттырады, салдарынан, жұмыс жасаушылардың психологиялық күйіне әсер етеді. Кеңсе ғимараттың жоғарғы қабаттарының бірінде орналасқандықтан аралас табиғи жарықтандыру пайдаланылған. Аралас табиғи жарықтандыруда жоғары табиғи жарықтандыру мен бүйірлі табиғи жарықтандыру бірігіп қолданылады. Табиғи жарықтандырудың бүйірлі жарықтандыру түрі қолданылады, яғни кеңседе орналасқан екі терезе арқылы жарық сәулесі түседі. Терезе қабырғаның тек бір жағында ғана орналасқан, сондықтан көзді шағылыстырмайды. Бұл ҚР ҚН 2.04-01-2011 [4] талаптарына сәйкес келеді.

#### 4.1 кесте - Компьютер орналасқан кеңселердегі жарықтандыру нормалары

Ақпарат тасымалдаушысындағы мәтін сипаттамасы	Жарықтандыру жүйесіндегі нормативтер лк
Орташа түс қанықтығы бар мәтін шрифті 1,5 мм-ден төмен болған жағдайында	500
Жоғары түс қанықтығы бар мәтін шрифті 1,5 мм-ден төмен болған жағдайында	400
Мәтін шрифті 1,5 мм-ден жоғары болған жағдайында	300

Кеңсе жарық ағыны 3450 Лм болатын 3 жарықдиодты шамдармен жарықтандырылған. Олар үнемді, сондықтан иесіне энергия шығынын азайтуға мүмкіндік береді, бірақ оған аса баптау қажет емес. Жарықдиодты шамдар 50-100 мың сағат қызмет етеді, жақсы түс беруімен және жарықтың ең жоғары шығуымен ерекшелінеді. Жарық шығуы 90 лм / Вт-тан басталады, ал қыздыру шамы үшін бұл параметр небары 10-20 лм / Вт құрайды. Жалпы алғанда 90-100 лм / Вт аралығындағы көрсеткіш өте жоғары болып саналады және жарықтандыру құрылғысының сапасын жоғары екенін көрсетеді.

Дәл бұл шамның қолдану себебі ,компьютерлер бар кеңселерде жұмыс жасауға оңтайлы болу үшін жарықтандыру құрылғыларының өлшемі ҚР ҚНЖЕ 2.04-01-2011 [4] талаптарына сәйкес 300 лк –тен төмен болмайтын шамдар қолданылды. Ал егер жарықтандыру өлшемі сол шамадан төмен болса,ол төмендегідей зиянды факторларға әкеліп соғады:

- жұмыс жасау қабілетінің төмендеуі;
- ұйқының бұзылуы;
- денсаулыққа қатысты проблемалар (бас ауруы, көру қабілетінің төмендеуі, жүрек-тамыр аурулары және т.б.);
- шаршау;
- депрессия және стресс;
- қозғалтқыш белсенділігінің төмендеуі;
- еңбек өнімділігінің төмендеуі.

Осы факторлардың алдын-алу үшін нормативтерде көрсетілген талаптарға сәйкес жұмыс орны жоспарланып,жарықтандырудың дұрыс жолы қолданылды.

### **4.1.3 Компьютермен жұмыс жасау кезіндегі жұмыс орнының эргономикасы**

Компьютермен тығыз байланысты жұмыс болғандықтан одан келетін зиянды факторлардан қорғану мақсатында жұмыс аймағымызды барлық нормативтерге сай ұйымдастыру өте маңызды болып табылады. Компьютері бар жұмыс орнын оңтайлы жоспарлауда негізгі болып саналатын- үстел, орындық және компьютер. Мұндай жұмыста арқа, мойын және көру мүшелері зардап шегеді. Жұмыс орнын дұрыс ұйымдастыру зиянды азайтуға ғана емес, сонымен қатар жұмыс процесінің өзін ыңғайлы болуына да көмектеседі. Қарастырылып отырған жұмыс орнында 3 дербес компьютер және олар орналасатын екі жиһаз түрі бар.

Компьютер орналасқан үстел параметрлері:

- биіктігі – 64 см;
- тереңдігі – 85 см;
- ені – 120 см .

Бірінші, кеңсеміздегі үстел биіктігі қолайсыз болып саналады. Себебі жалпы адам бойының орташа мәнін алатын болсақ салыстырмалы түрде бұл биіктік өте аз. Жұмысшы компьютер алдында отырған кезде монитор деңгейімен бірдей орналасқан шарт. Олай болмаған жағдайда адам тізесін шектен тыс бүгіп және арқасын бүкірейтіп отырады. Бұл остеохондроз секілді арқа және мойын ауруларына әкеліп соғады. Мұндай жағдайларды алдын алу мақсатында ҚР ҚНЖЕ 3.02-04-2002 [5] талаптарына сәйкес биіктігі - 68-80 см ,тереңдігі - 80-100 см ,ені - 80-140 см аралығында болатын үстелдерге ауыстырылды.

Енді екінші мәселе: үстел тереңдігі кем дегенде 80 см болуы. Оның себебі қазіргі мониторлардың үлкеюінде. Жұмыс барысында қолданылатын компьютерде диагональ 20-24 дюймге ие. Медициналық стандарттарға сәйкес көріністі көру қабілетін бұзбау үшін мониторды 60-75 см қашықтықта орналастырдық. Үстелді таңдағанда, монитордың өлшемін ескерілді, ол неғұрлым үлкен болса, соғұрлым оны алшақ орналастырдық . Мониторды қандай қашықтықта орнату керектігін білу үшін, оның диагоналын 3-ке көбейтіп қауіпсіз арақашықтықты анықтай аламыз.

Және соңғы параметр: үстелдің ені. Оған жұмыс үшін барлық керек-жарақтар мен құрылғыларды «бәрі жақын» қағидаты бойынша орналастырылған. Біздің жұмыс үстеліміздің үстінде шығару және енгізу құрылғылары орналасқандықтан ені небәрі үлкен үстел таңдалды. Жалпы, жұмыс бетінің ауданы қосымша құралдардың санына байланысты. Мысалы, ені 80 см болатын үстелге монитор, пернетақта, тінтуір, құжаттар жинағын және бір шыны кофені орналастыруға болады.

Компьютердегі жұмыс монитор экранында көрсетілген ақпаратты визуалды қабылдаумен байланысты, сондықтан көру мүшесіне көп жүктеме түсіріледі. Көру мүшесінің жұмысының нашарлануының белгілерін екі топқа бөлуге болады:

- көздің белгілері (ауырсыну, тітіркену, қызу, қызару, қышу);
- визуалды белгілер (көз бұлдырлауы, екі жақты көру немесе жыпылықтау).

Жұмыс орнына адам жиі келетін болғандықтан 3 жұмысшы және 5 келушілерге арналған орындықтар орналастырылған. Компьютермен жұмыс жасау барасында ыңғайлы және барлық талаптарға сәйкес келетін орындықтар таңдалды, кеңсеге арнайы эргономикалық отырғыштар алынды. Отыру барысындағы адам таңдаған параметрлерді сақтап тұрады. Эргономикалық орындықтар күніне 6 сағаттан артық отыратын адамдарға арналған. Олар адамның физиологиялық ерекшеліктерін ескере отырып жасалған және әртүрлі жеке түзетудің механизмдерімен жабдықталған. Орынның анатомиялық формасы омыртқа бағанасын дұрыс қалыпта қолдайды, төменгі арқа мен мойынға жүктемені азайтады. Ұзақ отыру кезінде жайлылықты қамтамасыз етеді және бұлшықеттің босаңсуына ықпал етеді және ауырсынудың пайда болуына жол бермейді. Жұмыс орнының ұтымды ұйымдастыру нормативтеріне сәйкес болмаған жағдайда жұмыскер көптеген зиянды әсерлерге ұшырайды. Соның ішінде жалпы адам денсаулығына зияны. Осы айтылған зиянды факторлардың алдын алу мақсатында инженерлермен орындалған кешендер:

-Білектің орналасуын бақыланды. Қолды білекке қатты бүгуге немесе майыстыруға, үстелге сүйенбеу қарастырылды;

-Компьютер тінтуірінің қатты қысылуын болдырмадық, бұл қолдың және саусақтардың бұлшық еттеріне қан берілуінің бұзылуына, олардың шамадан тыс зақымдануына әкеледі;

-Тінтуір қолдың алға қарай толық созудың қажеті болмайтындай етіп орналастырылды;

-Компьютерде жұмыс істеуге кететін уақыт шектелді;

-Әр сағат сайын кем дегенде 15 минуттық үзіліс жасау ұсынылды.

#### **4.1.4 Электрқауіпсіздікті талдау**

Компьютерлермен және компьютерлік техникамен жабдықталған жұмыс бөлмесі жоғары қауіпті бөлмелерге жатады. Компьютер - 220 В 50 Гц бір фазалы айнымалы ток желісімен жұмыс жасайды. Кез-келген электр құрылғысымен жұмыс істеу кезінде ток соғу қаупі болады. 3 Гц-ден 300 МГц-ке дейінгі электромагниттік өрістер, электростатикалық өрістер, көзге көп күш түсіру, әртүрлі типтегі үлкен жүктемелер маңызды факторлар болып табылады. Электромагниттік өріс орталық жүйке жүйесіне теріс әсер етеді,

бас ауруы, айналуы, жүрек айнуы, депрессия, ұйқысыздық, тәбеттің кетуіне әкеліп соғады. Төмен жиіліктегі ЭМӨ тері ауруларын (безеу, экзема, қызғылт қынап және т.б.), жүрек-тамыр жүйесі мен асқазан-ішек жолдарының ауруларын тудырады; ол лейкоциттерге әсер етеді, бұл ісіктердің, соның ішінде қатерлі ісіктердің пайда болуына әкелді.

Электростатикалық өрістің негізгі көзі (ЭСӨ) - электронды сәулені жеделдету үшін экранның ішкі бетіне берілетін оң потенциал. ESP монитор экраны мен адамның арасындағы айырмашылыққа байланысты қалыптасады. Айналадағы заттардың және ауа ылғалдылығының мөлшері оның мәніне айтарлықтай әсер етеді (ылғалдылығы 50% -дан жоғары кезінде ЭСӨ болмайды). ЭСӨ-нің өсуіне пернетақта мен тінтуірдің қолмен үйкелісінен туындайтын электрлендірілген қосылыстары әсер етеді. Жоғары кернеудің электростатикалық өрісі жасушалардың дамуын өзгертеді және тоқтатуы мүмкін, сондай-ақ объективтің кейіннен булануымен катаракта тудырады. Сондықтан электр қондырғыларын пайдалану кезінде белгілі бір қорғаныс шараларын сақтадық.

Біріншіден, бөлмеде компьютермен жұмыс істеу үшін қажетті жағдайлар жасалынды: ауаның ылғалдылығы 75% , ал температурасы 30 ° С, еден төсенішінің өткізгіш шаңының жоқтығына және еден қаптауының электр қауіпсіздігіне көз жеткізілді. Компьютерді орналастыруда қол жететін аймақта металл заттардың: жылыту радиаторларың, құбырлардың жоқтығына аса мән бердік. Бұл ҚР ҚНЖЕ 4.04-06-2002 [11] талаптарына сәйкес келеді.

Бөлмеде компьютерлік жабдықты орнатпас және пайдаланбас бұрын, розеткалардың, ажыратқыштардың және сымдардың жақсы күйде екендігіне және еш қиындықсыз жұмыс жасайтынына көз жеткізілді. Бір розеткаға барлық жабдықты бірден қосуға болмайды. Жүктеме бірнеше розеткаға біркелкі таратылған. Құрылғымен жұмыс кезінде электр сымның оқшаулауының тұтастығын мұқият қадағаланды.

Электрқауіпсіздікті сақтау мақсатында жұмыс орындарында міндетті түрде төмендегі шарттар орындалды:

- қорғанысты жерге қосу;
- оқшаулау және тоқ жүретін аймақтарды қоршау;
- төменгі кернеуді және қос оқшаулауды қолдану.

Жұмыс орындары стационарлық автоматты өрт сөндіру жүйелерімен және өрт детекторларымен жабдықталды. Компьютерлер мен перифериялық құрылғылар жұмыс істейтін бөлмені электромагниттік сәулеленудің сыртқы көздерінен барынша оқшауланған.

Ағашты ішкі безендіруде пайдалану шектеулі немесе азайтылған болуы , ал отқа төзімді қосылыстармен сіндірілген кезде қолданылды. Өртті сөндіру үшін суды пайдалану тек өрттің өршуі жағдайында ғана қолданылады. Компьютерлік технология судың тиіп кетуінен мүмкіндігінше оқшауланған .

Электр жарақатының ықтимал қаупін болдырмау үшін «Тұтынушылардың электр қондырғыларын пайдалану ережелері» және «Тұтынушылардың электр қондырғыларын пайдалану қауіпсіздігі ережелерінде» (тұтынушылардың электр қондырғыларын), сондай-ақ «Электр қондырғыларын орнату ережелерінде» көрсетілген талаптары орындалды.

Компьютермен жұмыс істеу кезінде электр тогының соғуын болдырмау үшін токтың жетекші бөлшектері жанаспайтындай етіп қорғайтын қосымша құрылғылар орнатылды; қауіпті азайту үшін оқшаулау трансформаторын негізгі желіден оқшаулау үшін пайдаланылды және барлық жағдайда электр жабдықтарының қорғаныс жерге тұйықталуы жүзеге асырылды. Компьютерлердің жоғары сапалы жұмыс істеуі үшін жеке жерге тұйықтау құрылды.

Жұмыс кезінде техникалық ақаулардың негізгі себебі болып табылатын компьютерлерді сапасыз электрмен жабдықтаудан қорғау үшін үздіксіз қуат көзі қолданылды. Олардың магистральды желідегі апат кезінде жүктемені электрмен қамтамасыз етті. ҰҚК пайдалану кезінде қорғаныс тізбегі және бейтарап сым бөлек орнатылды. Сонымен қатар, нашар жерлендіру жабдықтағы (мониторда) пайда болатын электромагниттік кедергілерден қорғауды азайтты.

## **4.2 Есептеу бөлімі**

### **4.2.1 Алғашқы өрт сөндіру құралдарына қажеттілікті есептеу**

Есептеу «Өрт қауіпсіздігі негіздері» әдістемелік нұсқаулығына [9] жүгіне отырып жүргізілді. Кеңседе орнатылған күзет-өрт сигнализациясының қазіргі заманғы жүйесі (ОПС) арнайы құрылғылардың тұтас жиынтығынан тұратын техникалық кешен. Олардың негізгі мақсаты-қауіп туралы алдын ала ескерту үшін жағдайды қамтамасыз ету. Жобаны әзірлеуде тек қана кәсіпқойлар айналысып, олардың нақты орналасқан жерін ескере отырып, өрт сөндіру және күзет құрылғыларының үлкен тобының жұмыс режимдерінің күрделі есебі жүргізілді. Орнату барысында жобаға сәйкес құбырларда (немесе арнайы қорғау гофрасында) төселуі тиіс кабельдер мен сымдардың қосымша қорғанысы қолданылды.

Бір кеңседе датчиктер санын есептеу үшін, олар орнатылған жердің ұзындығын анықтаймыз.

Өрт сөндіргіш ретінде ДИП-3 түтін хабарлағышын қолдандық. Кеңсенің биіктігі 5 м болғанда, бір хабарлағышпен бақыланатын аудан 14 м<sup>2</sup>.

ДИП-3 түтін хабарлағышының санын мына формула бойынша анықтаймыз:

$$M = C \cdot \frac{S}{S_0} \quad , \quad (4.1)$$



мұндағы Ц - бүтін санға дейін дөңгелектеу;

S - кеңсенің ауданы;

S<sub>0</sub> - бір ДИП-3 бақылайтын аудан.

$$M = Ц \cdot \frac{98}{14} = 7$$

Ауданы 98 м<sup>2</sup> кеңсеге 7 ДИП-3 түтін хабарлағышын орналастырамыз. Хабарлама пульті ретінде "Топаз - 3 М" пультін орнатамыз. Пультке барлық бөлме қосылады. "Топаз – 3 М" пульті хабарлаудың 10 аймағын бақылауға арналған. Сондықтанда бізде бір пультті қолнадылды.

Кеңседе электр көзіне жалғанған техникалық жабдықтар болғандықтан өрт болған жағдайда ҚР ҚНЖЕ 2.02-05-2012 [7] талаптарына сәйкес суды қолдануға болмайды. Сондықтан кеңсеге ОПУ-8 типті ұнтақты өрт сөндіргіш орнатылды. Техникалық сипаттамалар 4.2.1 - кестеде келтірілген.

4.2.1 кесте - ОПУ-8 типті ұнтақты өрт сөндіргішінің техникалық сипаттамалары

Параметрлер атауы	Өрт сөндіргіштердің өлшемдеріне арналған нормалар
Өрт сөндіргіш заттың салмағы, кг	8
Ұнтақ ағысының ұзындығы, м	5
Өрт сөндіргішті іске қосу уақыты, с	4
Ұнтақтың шығу уақыты, с	11
Өрт сөндіру ұнтағының қалдығы, %	12
Пайдалану үшін қол жетімді орта температурасы, С	-30
	+50
Диаметрі, мм	161
Биіктігі, мм	545
Зарядталған өрт сөндіргіштің салмағы, кг	12.5
В класты өрт сөндіру ауданы, м <sup>2</sup>	3.4
Жұмыс қысымы, МПа	1.4
Корпустың сыйымдылығы, г	10

ОПУ түріндегі ұнтақты унифицирленген өрт сөндіргіштер А класты (қатты заттар), В класты (сұйық заттар), С класты (газ тәріздес заттар) және 1000 В дейінгі электр қондырғыларындағы өртті сөндіруде пайдаланылады. Барлық өрт сөндіргіштер мерзімді тексеруден өтеді және қайта зарядталады.

Ауданы үлкен өртті сөндіру үшін  $m_d$  ұнтақты құрамының есептік салмағы мына формула бойынша анықталады:

$$m_d = k \cdot g_n \cdot V, \quad (4.2)$$

мұндағы  $k = 1,2$  - құрамның ескерілмейтін шығындарын өтеу коэффициенті;

$g_n = 0,4$  - құрамның нормативтік массалық шоғырлануы;

$V$  - кеңсе көлемі.

$$V = A \cdot B \cdot H, \quad (4.3)$$

мұндағы  $A = 14$  м - кеңсенің ұзындығы;

$B = 7$  м - кеңсенің ені;

$H = 5$  м - кеңсенің биіктігі.

$$V = 14 \cdot 7 \cdot 5 = 490 \text{ м}^3$$

Демек:

$$m_d = 1,2 \cdot 0,4 \cdot 490 = 235,2 \text{ кг}$$

Үй-жайларды жабатын конструкциялар алаңының аумағы 1% -дан 10% -ға дейін болатын ашық тесіктер болған жағдайда, тесіктердің алаңының  $1 \text{ м}^2$  үшін 5 кг-ға тең қосымша ағын қажет. ( $232,2 + 5 = 237,2$  кг).

Цилиндрлердің болжамды саны 20 литрлік 12,5 кг ұнтақ композициясының сыйымдылығын есептеуден анықталады. Яғни біздің есептеулер бойынша 20 литрлік ұнтақты ОПУ-8 өрт сөндіргіші қанағаттандырмайды. Сондықтан біз кеңсеге барлығы 18 өрт сөндіргішін орнаттық.

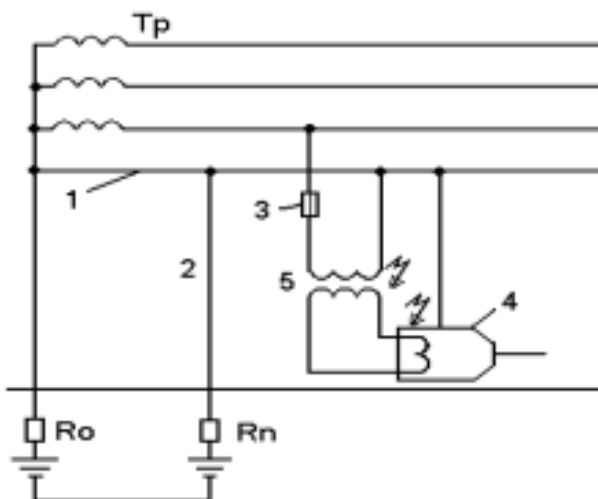
#### 4.2.2 Қорғаныстық нөлдеу есебі

Есептеу «Нөлдеу есебі» әдістемелік нұсқаулығы [8] арқылы жүргізілді. Бағдарлама орнатылатын кеңседе міндетті түрде электр құрылғылар жұмыс жасайды. Олардың бойынан ток өтеді, ал оның адам өміріне қаупі бар. Негізгі ток көзі жиілігі 50Гц және кернеуі 220В болатын үшфазалық жүйеден келеді.

Ток соғу қаупінен сақтанып, барлық құрылғылар ҚР ҚНЖЕ 4.04-06-2002 [6] талаптарына сәйкес жерлендірілген.

Шекті сатыдағы кернеумен токтың көрсеткіші  $U = 2$  В  $I = 0,3$  мА-дан, ал авариялық жағдайда кернеумен токтың көрсеткіші  $U = 20$  В  $I = 6$  мА-дан аспайды. Қорғандырылған жерлендіру ток өткізетін өткізгіштердің изоляциясының зақымданған жерленірінен ток соғу қаупін болдырмайды. Электр қондырғыларда оқшаулаудың ақаулығының жанында токтың соғуынан қорғанысы үшін қорғаныштық жерлендіру ескеріледі. Нөлдеу өткізгіші әр 250 м сайын қайталана жерлендіріп отырады және соңына қарай 200 м сайын. Осыған сәйкес өткізгіш мына талаптарға сәйкес тандалды, тұйықталу кезінде корпуста немесе нөлдік өткізгіште қысқаша тұйықтау пайда болды, ол автоматтың немесе жақындағы предохранительдің өшуін қамтамасыз етеді.

Тұйықталу кезіндегі зиянды потенциалдарды төмендету үшін, кедергісі 10 Ом-нан аспайтын қайта нөлдеу қолданылды. Кеңседегі электр құрылғысын нөлдеу схемасы (4.2 суретте) көрсетілген.



4.2 сурет - Кеңседегі электр құрылғысын нөлдеу схемасы

$R_o$  – бейтарап жерлендіру кедергісі;  $R_n$  – адамның есептелген кедергісі; 1 – нөлдеу магистралы; 2 – магистралды қайта жерлендіру; 3 – сөндіру аппараты; 4 – электрлі қондырғы; 5 - трансформатор.

Есептеудің мақсаты нөлдеудің тізбекті өшіру қабілеттілігін тексеруге негізделеді:

$$J_{KT} > 3J_{бал.сіл} > 1,25J_{авт}, \quad (4.4)$$

Бастапқы мәліметтер:  $U = 380$  В; типі–4А160М2,  $N = 18,5$  кВт.

$J_{кз}$  келесі формуламен есептеледі:

$$J_{\kappa 3} = \frac{U_{\phi}}{Z_t / 3 + Z_n}, \quad (4.5)$$

мұндағы  $U_{\phi}$  – фазалық кернеу, В;

$Z_t$  – трансформатор кедергісі, Ом;

$Z_n$  – «фаза-нөл» циклінің кедергісі, ол келесідей формуламен анықталады:

$$Z_n = \sqrt{(R_{\phi} + R_{nn})^2 + (X_{\phi} + X_o + X_n)^2}, \quad (4.6)$$

$$Z_n = 0,487 \text{ Ом.}$$

1. Электр қозғалтқышының Р қуатын біле отырып, оның  $J_n$  эл.қоз. номиналды тогын анықтаймыз:

$$P = \sqrt{3} \cdot U_n \cdot J_n^{\text{эл.қоз}} \cos \alpha / 1000 \text{ кВт}, \quad (4.7)$$

мұндағы Р – қозғалтқыштың номиналды қуаты, кВт;

$U_n$  – номиналды кернеу, В;

$\cos \alpha = 0,92$  – токтың қай бөлігінен активті қуат алынатынын және магниттелгенін көрсететін қуат коэффициенті.

$$J_n^{\text{эл.қоз}} = 1200 \cdot 18,5 / \sqrt{3} \cdot 380 \cdot 0,92 = 36,6 \text{ А}$$

2.  $R_n$  және  $R_{\phi}$  активті кедергілерің анықтау үшін, нөлдік және фазалық өткіштердің ауданы, ұзындығын сонымен қатар материалы таңдалды.

Түсті металдан жасалған өткізгіш кедергісі келесі формуламен анықталады:

$$R = \rho \cdot \ell / S \text{ Ом}, \quad (4.6)$$

Фазалық сым ретінде алюминийден жасалған ауданы  $30 \text{ мм}^2$  және ұзындығы  $\ell = 130 \text{ м}$  өткізгішті алдық. Нөлдік өткізгіштің ауданы мен материалы фазалық өткізгішке қатысты таңдалды. Яғни, нөлдік және фазалық өткізгіштердің аудандары өзара тең болады. Алюминийден жасалған фазалық және нөлдік өткізгіштің активті кедергісі оның параметрлері  $\ell = 130 \text{ м}$ ,  $S = 30 \text{ мм}^2$  болғанда келесідей:

$$R_{\phi} = 0,028 \cdot 130 / 30 = 0,12 \text{ Ом},$$

$$R_n = 0,028 \cdot 130 / 30 = 0,12 \text{ Ом}.$$

$X_{\phi}$  және  $X_o$  мәні келесідей :

$$X_{\phi}=0,0156 \cdot 0,15 = 0,0023 \text{ Ом};$$

$$X_o = 0,0156 \cdot 0,15 = 0,0023 \text{ Ом}.$$

$$X_n = 0,6 \text{ Ом/км, іс жүзінде } 0,6 \text{ Ом/км деп алынады.}$$

3. А106М2 электрқозғалтқышының негізгі техникалық сипаттамаларын анықтаймыз:  $N = 16,5$ ;  $\cos \alpha = 0,92$ .

$$J^{\text{пуск}} / J^{\text{ном}} = 7,5 , \quad (4.7)$$

4.  $J^{\text{н}}_{\text{эл.қоз}}$  мәнін біле отырып электрқозғалтқышының бастапқы тогын анықтаймыз:

$$J^{\text{пуск}}_{\text{эл.қоз}} = 7,5 \cdot J^{\text{н}}_{\text{эл.қоз}} = 7,5 \cdot 36,6 = 274,5 \text{ А} , \quad (4.8)$$

Балқыма сілтеменің номиналды тогын анықтаймыз:

$$J^{\text{н}}_{\text{бал.сіл}} = J^{\text{пуск}}_{\text{эл.қоз}} / \alpha = 274,5 / 2,5 = 109,8 \text{ А} , \quad (4.9)$$

мұндағы:  $\alpha$  – жұмыс режимінің коэффициенті, сирек қосылатын механизмдерді іске келтіретін қозғалтқыштар үшін (транспортер, вентиляторлар). Біздің жағдайымызда  $\alpha=2,5$ .

5. Қысқы тұйықталу тогының мәнін анықтаймыз:

$$J_{\text{КТ}} > 3J^{\text{н}}_{\text{бал.сіл}} = 3 \cdot 109,8 = 329,4 \text{ А} , \quad (4.10)$$

Нөлдік және фазалық өткізгіш тогының  $\delta$  тығыздығын есептейміз. Алюминийден жасалған өткізгіштердің рұқсат етілген ток тығыздығы 4-8 А/мм<sup>2</sup>- ден аспайды.

$$\delta = J^{\text{н}}_{\text{эл.қоз}} / S = 36,6 / 30 = 1,2 \text{ А/мм}^2 , \quad (4.11)$$

6.  $X_n = 0,6 \text{ Ом/км}$  екенін біле отырып, «фаза-нөл» циклының сыртқы индуктивті кедергісін анықтаймыз:

$$X_n = 0,6 \cdot 0,15 = 0,09 \text{ Ом} , \quad (4.12)$$

7. «Фаза-нөл» циклының кедергісін және қысқа тұйықталу тогын есептейміз:

$$Z_{\Pi} = \sqrt{(R_{\phi} + R_{HH})^2 + (X_{\phi} + X_o + X_{и})^2} = \\ = \sqrt{(0.17 + 0.17)^2 + (0.0023 + 0.0023 + 0.09)^2} = 0.35 \text{ Ом}, \quad (4.13)$$

$$J_{\kappa 3} = \frac{U_{\phi}}{Z_t / 3 + Z_n} = \frac{220}{0.487 / 3 + 0.35} = 429 \text{ А} \quad (4.14)$$

Сенімді қорғау жағдайы қамтамасыз етілуін тексерейік:

$$J_{\kappa T} > 3J_{\text{бал.сіл}}^H; \\ 429 > 3 \cdot 91,8 \text{ А}; \\ 429 > 275,4 \text{ А} \quad (4.15)$$

Анықтауымыз бойынша,  $J_{\kappa T}$  балқыма сілтеменің сақтандырғышының номиналды тогынан үш есеге артық, сондықтан қысқа тұйықталу кезінде бұзылған фазаны 5...7с ішінде өшіреді.

#### Қорытынды

Қорытындай келе, бұл бөлімде бағдарламалық өнім орнатылатын кеңсенің жұмыс эргономикасы, өрт қауіпсіздігі және электр қауіпсіздігі қарастырылды. Еңбек жағдайларын зерттеу барысында жұмысшы денсаулығына әсер ететін зиянды факторлар анықталып, олардың алдын-алу жоспары құрылды. Сонымен қатар сол жоспар бойынша есептеулер жүргізілді. Атап айтқанда өрт өшіру құрылғыларына қажеттелік және электр қауіпсіздікті қамтамасыз ету мақсатында электрқұрылғысын нөлдеу есептелді.

#### **5. Бағдарламалық жасақтаманы (БЖ) әзірлеумен байланысты дипломдық жұмыстың техникалық-экономикалық негіздемесі**

Бұл дипломдық жұмыс “Python” бағдарламасы арқылы құрылатын нейрожелінің телекоммуникация саласындағы маңыздылығын сипаттайды.

Бұл жұмыстың мақсаты: Нейрожелінің құрылу жолының ең оңтайлы тәсілін зерттеп, онын телекоммуникация саласындағы күрделі мәселелерді шешу мақсатында маңыздылығын және пайдасын анықтау. Бұл нейрожелі телекоммуникациялық қолдаумен айналысатын белгілі-бір мекемеге компьютерлеріне орнатылады.

Дипломдық жұмыстың осы бөлімінде уақыт, еңбек және қаржылық шығындарды айқындайтын экономикалық компоненттердің есебі жүргізіледі.

## 5.1 Бағдарламалық жасақтаманы әзірлеудің еңбек салымы

Бағдарламалық жасақтаманы әзірлеудің күрделілігін анықтау үшін, алдымен, әзірлеу кезінде орындалуға тиісті барлық негізгі кезеңдер мен жұмыс түрлерінің тізімі жасалады.

Бағдарламалық өнімнің күрделілігі 5.1 кестеде көрсетілген.

5.1 кесте- Жұмысты кезеңдері мен түрлері бойынша бөлу және олардың күрделілігін бағалау

БЖ-ны іске асыру кезеңі	Осы кезеңдегі жұмыс түрі	Іске асырудың күрделілігі , адам× сағ.
БЖ құру мақсаты туралы мәлімдеме	Мақсат таңдау	1 сағ
	Бағдарламалық жасақтаманы құру таңдауын негіздеу	1,5 сағ
Нарықты талдау	Нарықты зерттеу	5 сағ
Алгоритмдеу	Алгоритмдер мен әртүрлі процестердің сипаттамасы,схема құру	3,5 сағ
Бағдарламалық өнімнің ең оңтайлы жолын талдау	Бағдарламалық өнімнің құрылымын зерттеу	40 сағ
Бағдарламалық жасақтаманың құрылымын зерттеу	БЖ барлық негізгі құралдарымен танысу	30 сағ
БӨ телекоммуникация саласындағы пайдасын негіздеу	БӨ телкоммуникация саласындағы маңыздылығын зерттеу	160 сағ
Бағдарламалық жасақтаманың орындалуының жалпы күрделілігі	246 сағ	

## 5.2 Бағдарламалық жасақтаманы зерттеуге кеткен шығындарды есептеу

Бұл кезеңде қажетті материалдық ресурстардың шығындары есептеледі, олар өз кезегінде негізгі және көмекші шығындарға, еңбек шығынына, электр қуатына, амортизацияға және әлеуметтік салыққа жұмсалған жалпы шығындарға, сондай-ақ басқа да шығындарға бөлінеді.

Материалдық ресурстардың құнын есептеу төменде келтірілген 5.2 кестеде сипатталған.

5.1 кесте- Материалдық ресурстарға кететін шығындар

Материалдық ресурстың атауы	Өлшем бірлігі	Тұтынылатын материалдың мөлшері	Бірлік бағасы, тг	Суммасы, тг
А4 қағаз	Дана	1	1800	1800
Картридж	Дана	1	1500	1500
БАРЛЫҒЫ				3300

Әр түрлі материалдық ресурстарға жұмсалатын шығындардың жалпы мөлшерін ( $\Sigma_{\text{м}}$ ) төменде келтірілген формула бойынша анықтаймыз:

$$\Sigma_{\text{м}} = \sum_{i=1}^n P_i \cdot B_i, \quad (5.1)$$

мұндағы  $P_i$ -  $i$ -түріндегі материалдық ресурстардың шығыны, натурал бірліктер;

$T_{si}$  - материалдық ресурстардың  $i$ -ші түрінің бірлігінің бағасы, тг;

$i$  - материалдық ресурстың түрі;

$n$  - материалдық ресурстар түрлерінің саны.

$$\Sigma_{\text{қағаз}} = 1 \cdot 1800 = 1800 \text{ тг}$$

$$\Sigma_{\text{картридж}} = 1 \cdot 1500 = 1500 \text{ тг}$$

$$\Sigma_{\text{жалпы}} = 1800 + 1500 = 3300 \text{ тг}$$

Қажетті құрал-жабдықтар мен бағдарламалық қамтамасыздандыруға кететін шығындарды есептеу 5.3 кестеде көретілген нысанда жүзеге асырылады.



5.3 кесте - Жоба үшін қажет жабдықтар мен бағдарламалық қамтамасыздандыруға кететін шығындарды есептеу

Құрылғы мен бағдарламалық қамтамасыздандырудың атауы	Бірліктің бағасы теңгемен	Өлшем бірлігі	Данасы	Сумма, тг
Acer Aspire 3 A315-54K NX.NEEER.01G ноутбугі	375 000	дана	1	375 000
A4TECH N-708X сымсыз компьютер тінтуірі	14 000	дана	1	14 000
KYOCERA FS-1040 принтері	45 000	дана	1	45 000
Iphone смартфоны	158 000	дана	1	158 000
БАРЛЫҒЫ	592 000,00			

Төмендегі формула бойынша жабдықтар мен бағдарламалық қамтамасыздандыруға жұмсалатын шығындардың жалпы мөлшерін анықтаймыз:

$$\text{Ш}_{\text{құрылғы}} = \sum_{i=1}^n P_i \cdot B_i \quad (5.2)$$

мұндағы  $P_i$ -  $i$ -түріндегі материалдық ресурстардың шығыны, натурал бірліктер;

$B_i$  -  $i$ -ші типтегі жабдықтың бірлігінің бағасы, тг;

$i$  - жабдық түрі;

$n$  - жабдық түрлерінің саны.

$$\text{Ш}_{\text{ноутбук}} = 1 \cdot 375000 = 375000 \text{ тг}$$

$$\text{Ш}_{\text{тінтуір}} = 1 \cdot 14000 = 14000 \text{ тт}$$

$$\text{Ш}_{\text{принтер}} = 1 \cdot 45000 = 45000 \text{ тт}$$

$$\text{Ш}_{\text{смартфон}} = 1 \cdot 158000 = 158000 \text{ тт}$$

$$\text{Ш}_{\text{жалпы}} = 375 000 + 14 000 + 45 000 + 158 000 = 592 000 \text{ тт}$$

Бағдарламалық жасақтама жасау үшін электр энергиясын тұтынатын техника қолданылатындықтан, 5.4 кестеде келтірілген электр энергиясының өзіндік құнын есептеу қажет.

#### 5.4 кесте-Электр қуатына жұмсалатын шығындар

Құрылғы атауы	Номиналды қуат , кВт	Электр қуатын пайдалану коэффициенті	Бағдарламалық қамтамасыздан-дыруды әзірлеуге арналған жабдықтың жұмыс уақыты, сағ	Электр қуатының бағасы, тг / кВт· сағ	Бағасы , тг
Acer Aspire 3 A315-54K NX.NEEER .01G ноутбугі	0,08	0,8	230	18,4	270,8
Iphone смартфоны	0,014	0,8	28	18,4	5,7
KYOCERA FS-1040 принтері	0,304	0,8	12	18,4	53,7
БАРЛЫҒЫ					330,20

Электр энергиясына жұмсалатын ( $Ш_э$ ) жалпы шығын мына формула бойынша есептеледі:

$$Ш_э = \sum_{i=1}^n M_i \cdot K_i \cdot T_i \cdot B_i , \quad (5.3)$$

мұндағы  $M_i$  -  $i$ -ші электр жабдықтарының номиналды қуаты, кВт;  
 $K_i$  -  $i$ -ші электр жабдығының қуатты пайдалану коэффициенті (болжанған  $K_i = 0.7, 0.9$ );  
 $T_i$  -  $i$ -ші жабдықтың бағдарламалық жасақтаманы жасаудың барлық кезеңіндегі жұмыс уақыты;  
 $B_i$  - электр энергиясының бағасы, тг / кВтсағ;  
 $i$  - электр жабдықтарының түрі;  
 $n$  - электр жабдықтарының саны.

$$Ш_{эноутбук} = 0,08 \cdot 0,8 \cdot 240 \cdot 18,4 = 270,8 \text{ тг}$$

$$Ш_{эсмартфон} = 0,014 \cdot 0,8 \cdot 28 \cdot 18,4 = 5,7 \text{ тг}$$

$$Ш_{эпринтер} = 0,304 \cdot 0,8 \cdot 12 \cdot 18,4 = 53,7 \text{ тг}$$

$$Ш_{\text{жалпы}} = 270,8 + 5,7 + 53,7 = 330,2 \text{ тг}$$

«Еңбекке ақы төлеу» бабы үшін бағдарламалық жасақтаманы әзірлеумен айналысатын және жұмыс істейтін барлық қызметкерлердің еңбек ақысы қосылуы керек. Еңбек шығындары 5.5 кестеде келтірілген нысан бойынша есептеледі.

5.5 кесте – Еңбек ақы шығындары

Қызметкерлер санаты	Бағдарламалық жасақтаманың күрделілігі, адам × сағ	Сағаттық төлем ,тг / сағ	Сумма, тг
Программист	246	956,4	235 274
Жоба басқарушысы	8	1081,43	8 651
БАРЛЫҒЫ			243 925,00

Еңбек ақыға жұмсалатын шығындардың жалпы құны ( $Ш_{\text{еңбек}}$ ) төмендегі формула бойынша анықталады:

$$Ш_{\text{еңбек}} = \sum_{i=1}^n CT_i \cdot K_i, \quad (5.4)$$

мұндағы  $CT_i$  -  $i$ -ші қызметкердің сағаттық төлемі, тг;

$K_i$ - бағдарламалық жасақтаманың күрделілігі, адамдар × сағ;

$i$  - қызметкерлер санаты;

$n$  - бағдарламалық жасақтаманы жасауға қатысатын жұмысшылар саны.

Қызметкердің сағаттық ақысын мына формула бойынша есептеуге болады:

$$CT_i = \frac{EA_i}{ЖУК_i}, \quad (5.5)$$

$$CT_i = \frac{EA_i}{ЖУК_i} = \frac{180000}{168} = 1071,43$$

$$CT_{\text{бас}} = \frac{EA_{\text{бас}}}{ЖУК_{\text{бас}}} = \frac{210000}{168} = 1250$$

ЖУҚ - бұл қызметкердің жұмыс уақытының қоры, ол айына шамамен 168 сағатты құрайды.

Бағдарламалық жасақтаманы әзірлеудің күрделілігі 5.1-кестеге сәйкес анықталады.

«Әлеуметтік салық» бабына бағдарламалық жасақтаманы әзірлеуге қатысқан барлық қызметкерлердің ( $\Pi_{\text{еңбек}}$ ) еңбек құнынан 9,5 % мөлшерінде есептелетін сома кіреді. Есептеу кезінде зейнетақы шегерімдеріне ( $\Pi_{\text{еңбек}}$  мөлшерінің 10%) әлеуметтік салық салынбайтындығын ескеру қажет (мөлшерлемелер 2020 жылға арналған).

$$\Pi_{\text{зш}} = \Pi_{\text{еңбек}} \cdot 0,1 = 235\,274 \cdot 0,1 = 23\,527$$

$$\Pi_{\text{еңбек}}(\text{з.ш. қосқан жағдайда}) = \Pi_{\text{еңбек}} - \Pi_{\text{з.ш}} = 235\,274 - 23\,527 = 211\,747$$

$$C_{\text{с.прог}} = \Pi_{\text{еңбек}}(\text{з.ш. қосқан жағдайда}) \cdot 0,095 = 211\,747 \cdot 0,095 = 20\,115,96$$

Жоба басқарушысы үшін еңбек ақы шығындарын есептеу.

$$\Pi_{\text{зш}} = \Pi_{\text{еңбек}} \cdot 0,1 = 8654 \cdot 0,1 = 865$$

$$\Pi_{\text{еңбек}}(\text{з.ш. қосқан жағдайда}) = \Pi_{\text{еңбек}} - \Pi_{\text{з.ш}} = 8654 - 865 = 7789$$

$$C_{\text{с.басқ}} = \Pi_{\text{еңбек}}(\text{з.ш. қосқан жағдайда}) \cdot 0,095 = 7789 \cdot 0,095 = 739,95$$

$$C_{\text{с.жалпы}} = C_{\text{с.прог}} + C_{\text{с.басқ}} = 20\,115,96 + 739,95 = 20\,855,91$$

«Негізгі құралдардың амортизациясы» бабында бағдарламалық жасақтаманы әзірлеу кезінде қолданылатын жабдықтар мен бағдарламалық қамтамасыздандырудың (БҚ) құнынан алынатын амортизациялық аударымдардың сомасы келтірілген. Амортизациялық аударымдар 5.6 кестеде көрсетілген.

### 5.6 кесте - Негізгі қорлардың амортизациясы (НҚ)

Құрылғы атауы	Құрылғы мен бағдарламалық қамтамасыздандыру құны, тг	Жылдық амортизация коэффициенті, %	Жабдықтар мен бағдарламалық қамтамасыздандырудың тиімді жұмыс уақытының қоры, сағат /жыл	БЖ әзірлеуге арналған жабдықтар мен бағдарламалық қамтамасыздандыру жұмыс уақыты, сағ, тг	Сумма, тг
Acer Aspire 3 A315-54K NX.NEEEER.01 G ноутбүгі	375 000	22	1820	246	8 414
A4TECH N-708X сымсыз компьютер тінтуірі	14 000	32,3	1820	246	473,07
KYOCERA FS-1040 принтері	45 000	15,3	1820	8	22
Iphone смартфон	158 000	18	1820	22	420
БАРЛЫҒЫ					9330,07

Амортизациялық аударымдардың жалпы сомасы мынадай формула бойынша анықталады:

$$Ш_{AM} = \sum_{i=1}^n \frac{K \cdot H_{Ai} \cdot T_i}{100 \cdot T_{\text{тыімді}}}, \quad (5.6)$$

мұндағы  $\Phi_i$ - НҚ  $i$ -түрінің құны, тг;  
 $H_{Ai}$  -НҚ амортизациясының жылдық нормасы, %;  
 $T_i$ -БҚ әзірлеудің барлық кезеңіндегі  $i$ -түріндегі НҚ жұмыс уақыты, сағ.;  
 $T_{\text{тиімді}}$  -  $i$ -түріндегі НҚ бір жылдағы жұмыс уақытының тиімді қоры, сағ/жыл;  
 $i$  – НҚ түрлері;

n - НҚ саны.

НҚ құнын анықтау кезінде БҚ жеткізу және монтаждау, орнату шығындарын ескеру қажет. Бұл шығындар НҚ сатып алу шығындарының 10-25% мөлшерінде қабылдануы мүмкін.

НҚ амортизациясының жылдық нормалары ҚР Салық кодексі бойынша қабылданады немесе НҚ пайдалы пайдаланудың ықтимал мерзімін ескере отырып анықталады:

$$H_{Ai} = \frac{100}{T_{Ni}}, \quad (5.7)$$

$$H_{Ai} = \frac{100}{6} = 16,6\%$$

$$H_{Ai} = \frac{100}{4} = 25\%$$

$$H_{Ai} = \frac{100}{9} = 11,1\%$$

мұндағы  $T_{Ni}$ -і түріндегі НҚ пайдаланудың ықтимал мерзімі, жыл.

НҚ пайдалы қолданудың ықтимал мерзімі 3 жылдан 10 жылға дейін болуы мүмкін.

$$III_{AM2} = \frac{375000 \cdot 16,6 \cdot 246}{100 \cdot 1820} = 8414 \text{ тг}$$

$$III_{AM3} = \frac{14000 \cdot 25 \cdot 246}{100 \cdot 1820} = 473,07 \text{ тг}$$

$$III_{AM4} = \frac{45000 \cdot 11,1 \cdot 8}{100 \cdot 1820} = 22 \text{ тг}$$

$$III_{AM2} = \frac{158000 \cdot 22 \cdot 22}{100 \cdot 1820} = 420 \text{ тг}$$

$$III_{AM_{жалпы}} = 8414 + 473,07 + 22 + 420 = 9330,07 \text{ тг}$$

"Өзге шығындар" бабына коммуналдық төлемдерді, лицензиялау мен сертификаттауға арналған шығындарды, жарнамаға арналған шығындарды, кеңселік және өзге де шаруашылық шығындарды қоса алғанда, жалдау ақысына арналған шығынтар кіреді.

#### 5.6.1 кесте - Жалдау ақысы

Үй-жайдың ауданы, м <sup>2</sup>	1м <sup>2</sup> құны,тг	1 ай төлемі, тг	Жалға алу ұзақтығы(ай өлшемінде)	Сумма, тг
46	3200	165 000	1	165 000

Интернет шығындары 5.6.2-кестеде келтірілген.

#### 5.6.2 кесте - Интернет шығындар

1 айға төлемі, тг	Ай саны	Сумма, тг
6 000	1	6 000

Төменде көрсетілген формула бойынша қосымша шығындардың жалпы сомасын анықтаймыз ( $Ш_{\text{қосымша}}$ ):

$$Ш_{\text{қосымша}} = Ш_{\text{жалдау}} + Ш_{\text{интернет}}, \quad (5.8)$$

$$Ш_{\text{қосымша}} = 165000 + 6000 = 171000 \text{ тг}$$

Жеке баптар бойынша алынған деректер негізінде 5.7 кестеде келтірілген нысан бойынша БӨ зерттеуге кететін шығындар жиынтығы жасалды.

#### 5.7 кесте –БӨ зерттеуге арналған шығындар жиынтығы

Шығындар бабы	Сумма, тг	%
Материалдық шығындар	3 300,00	0,31
БҚ мен құрылғыға жұмсалатын шығындар	592 000,00	56,714
Электр қуатына жұмсалатын шығындар	330,20	0,03
Еңбек ақыға жұмсалатын шығындар	243 925,00	23,3
Әлеуметтік қажеттіліктерге жұмсалатын шығындар	20 855,91	1,99
Негізгі қорлар амортизациясы	9 330,07	0,89
Қосымша шығындар	171 000,00	16,3
Жиынтық бойынша БАРЛЫҒЫ	1 043 412,27	100

Бағдарламалық өнімді әзірлеудің өзіндік құнының құрылымы 5.1 суретте көрсетілген.



5.1 сурет - БӨ шығын жиынтығы

### 5.3 Бағдарламалық өнімнің бағасын анықтау

БӨ ықтимал (шарттық) бағасының шамасы тапсырыс берушінің (тұтынушының) және орындаушының экономикалық мүдделеріне жауап беретін деңгейде орындалу тиімділігі, сапасы мен мерзімдері ескеріле отырып белгіленуі тиіс. Қолданбалы БӨ үшін шарттық баға ( $B_{ш}$ ) мынадай формула бойынша есептеледі:

$$B_{ш} = Ш_{БӨ} \cdot \left(1 + \frac{P}{100}\right), \quad (5.9)$$

мұндағы  $Ш_{БӨ}$  – бағдарламалық өнімді әзірлеу құны (4-кестеден), тг;

$P$  - бағдарламалық жасақтаманың пайдалылықтың орташа деңгейі, ол 25% құрайды.

$$B_{ш} = Ш_{БӨ} \cdot \left(1 + \frac{P}{100}\right) = 1043412,27 \cdot (1 + 0,25) = 1304265,07 \text{ тг}$$

Әрі қарай сату бағасы қосымша құн салығын (ҚҚС) ескере отырып анықталады, ҚҚС ставкасы заңмен Қазақстан Республикасының Салық кодексімен белгіленеді. 2020 жылға ҚҚС ставкасы 12% деңгейінде белгіленген. Сату бағасы ҚҚС есебімен мына формула бойынша есептеледі:

$$B_{с.б} = B_{ш} + B_{ш} \cdot КҚС, \quad (5.10)$$



$$B_{camy} = 1\,304\,265,07 + 1\,304\,265,07 \cdot 0,12 = 1\,304\,265,07 + 156\,511,8 = \\ = 1\,460\,776,88 \text{ тт}$$

Сонымен, бағдарламалық өнімнің құны - 1 043 412,27 теңге;

Пайда – 417 364,608 теңге;

Шартты бағасы - 1 460 776,88 теңге.

Сонымен, жоба өзін 2,5 жылда ақтап шығады. Байланыс жүйесін жобалаудағы техникалық – экономикалық бөлімді қорытындылайтын болсам, қызмет көрсету сапасын үнемі бақылап отыру кез келген байланыс операторы үшін қажетті шара болып табылады. Бағдарламаны іске асыру 1 460 777 теңге көлемінде капиталдық салымдар қажет. Жыл ішіндегі шығындар сомасы нақты өндірістік өзіндік құнды немесе жылдық эксплуатациялық шығындарды құрайды, менің жағдайымда эксплуатациялық шығындар 1 043 412,27 теңгені құрады. Эксплуатациялық шығындар құрамындағы ең үлкен үлес – еңбекақы қоры 49%, сондай – ақ үстеме шығындар – жалпы шығындардың 36% құрайды. Экономикалық тиімділікті есептеуді жасай отырып, біз бұл жобаның тиімділігі, күрделі салымдар мен ағымдағы шығындардың ең қолайлы комбинациясы беретін артықшылығы бар деп қорытынды жасауға болады.

## Қорытынды

Чат - бот бірнеше түрлі міндеттерді шешуге мүмкіндік береді, мекеме жұмысы сапасының көрсеткіші болып табылады, әр түрлі функцияларды іске асыруға мүмкіндік береді, өзіне мақсатты аудиторияның қосымша назарын аударады. Алайда, желіні әзірлегеннен және орналастырғаннан кейін бот жұмысы мүлдем аяқталмайды. Интернет жағдайын талдау дер кезінде жаңартылмайтын және пайдаланушыларды өзекті және уақтылы ақпаратпен қамтамасыз етпейтін қосымшалар өз аудиториясын тез жоғалтады. Сондай-ақ, чат-бот бойынша орналасу мәселесі өте өткір : қажетті ақпаратты тез табу мүмкін болмаса, ол ақпараттың бот кітапханасында орналасу қажеті жоқ. Чат ботты құру кезінде бірінші кезекте оның құрылымы мен ақпараттық құрамдастарына назар аудару керек. Бірақ тағы бір маңызды аспект: боттың пайдаланушымен тірі адам тәріздес байланыс орнатуы.

Бұл дипломдық жұмыстың мақсаты "Python бағдарламалық қамтамасыздандыруы арқылы нейрожеліні құру " болып табылады. Дипломдық жұмысты орындау нәтижесінде мен оқу кезінде алған білімдерімді бекіттім, сондай-ақ интернет-жобаларды әзірлеу саласында жаңа практикалық білім алдым. PHP бағдарламалау тілінде және NLTK деректер қорын басқару жүйесі әзірленген.

Чат – ботты жобалау процесі төмендегі кезеңдерден тұрды:

- қолданыс аймағын шолу және талдау;
- әзірленетін қосымшаға қойылатын талаптарды нақтылау;
- бот макетін әзірлеу;;
- деректер базасын құру;
- әзірленген макет негізінде бот үлгісін беттеу;
- боттың клиенттік және әкімшілік бөліктері үшін интерфейстерді әзірлеу.

Жұмыстар аяқталғаннан кейін біз seo тұрғысынан оңтайландырылған, оның алдына қойылған міндеттерді табысты орындайтын дайын чат-ботты алдық.

Бұл жоба әрбір пайдаланушыдан келген сауалдарға жауап беріп ,олармен берік қарым- қатынас орнатып ,мекеме жұмысшыларының уақытын үнемдеп, олардың жұмыс сапасын жоғарлатып және қосымша пайдалушыларды жинақтау мақсатында құрылған.

## Әдебиеттер тізімі

1 Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с. Ларри Ульман Основы программирования на PHP [Электронный ресурс]/ Ларри Ульман— Электрон. текстовые данные.— М.: ДМК Пресс, 2007.— 286 с.

2 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489>.

3 Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87530>.

4 Мэтиз, Э. Изучаем PYTHON. Программирование игр, визуализация данных, веб-приложения / Э. Мэтиз. - СПб.: Питер, 2017. - 496 с.

5 Коржинский С. Н. Настольная книга Web-мастера: эффективное применение HTML, CSS и JavaScript. – М.: КноРус, 2011. – 416с.

6 Бенкен Е.С. PHP, MySQL, XML: программирование для Интернета. – С. Пб.: BHV, 2008. – 570 с.

7 Россум Г., Дрейк Ф.Л. Дж., Откидач Д.С. и др. Язык программирования Python; -, 2011. - 463 с.

8 Олескин, А. В. Сетевые структуры в биосистемах и человеческом обществе / А.В. Олескин. - М.: Едиториал УРСС, Либроком, 2015. - 304 с.

9 Омату, Сигеру Нейроуправление и его приложения. Книга 2 / Сигеру Омату. - М.: Радиотехника, 2000. - 615 с.

10 Редько, В.Г. Эволюция, нейронные сети, интеллект: Модели концепции эволюционной кибернетики / В.Г. Редько. - Москва: СИНТЕГ, 2017. - 224 с.

11 Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. - М.: Горячая линия - Телеком, 2013. - 384 с.

12 Рышард, Тадеусевич Элементарное введение в технологию нейронных сетей с примерами программ / Тадеусевич Рышард. - М.: Горячая линия - Телеком, 2011. - 535 с.

13 Тархов, Д.А. Нейросетевые модели и алгоритмы. Справочник / Д.А. Тархов. - М.: Радиотехника, 2014. - 643 с.

14 «HTML5 + CSS3. Основы современного веб-дизайна» / Кириченко А. В., Хрусталева А. А. – СПб: «Наука и Техника», 2018. – 352 с., ил.

15 Белоус Е. Как чат боты создают ценность для вашего бизнеса //

Энциклопедия маркетинга [Электронный ресурс]. — Режим доступа: <http://www.marketing.spb.ru/lib-comm/dm/bot.htm>

16 Кузнецов В. В. Перспективы развития чат-ботов//Успехи современной науки. — 2016. — № 12, 16–19 стр. «HTML и CSS. Разработка и дизайн веб-сайтов» /Джон Дакетт; [пер. с англ. М. А. Райтмана]. – Москва: Эксмо, 2019. - 480 с.: ил. - (Мировой компьютерный бестселлер).

17 А.А.Лукьяница. Применение адаптивных методов для обработки экспериментальных данных. В сб.: Нейроинформатика-2009. XI Всероссийская научно-техническая конференция. Лекции по нейроинформатике, с.126-162. НИЯУ МИФИ, 2009.

18 Г.А.Осоков. Практические аспекты нейровычислений в экспериментальной физике. В сб.: Нейроинформатика-2009. XI Всероссийская научно-техническая конференция. Лекции по нейроинформатике, с.163-206. НИЯУ МИФИ, 2009.

19 В.А.Головко. От многослойных персептронов к нейронным сетям глубокого доверия: парадигмы обучения и применение. В сб.: Нейроинформатика-2015. XVII Всероссийская научно-техническая конференция с международным участием. Лекции по нейроинформатике, с.47-84. НИЯУ МИФИ, 2015.

20 «Руководство по API чат» <https://habr.com/ru/post/262247/>

## **А қосымшасы**

(міндетті түрде)

### **Техникалық тапсырма**

Бағдарламалық өнімнің атауы (БӨ): "WABot".

БӨ-ді әзірлеу мақсаты: байланыс орнату мен деректерді жаңарту үшін интернетке қол жетімді сұраныстарға жауап беретін осы тақырып бойынша деректер базасы орнатылған чат-бот құру.

Бағдарламалық қамтамасыздандыру идеологиясы: пайдаланушылардың белгілі бір саласында көмек беру қорын таңдауда бағалау болып табылады.

Клиент интерфейсі үшін қолданылатын әзірлемелер: HTML, CSS, JavaScript, Angular.

БӨ моделін таңдау: құрылымдық модель – бағдарламаның басқа бөліктерінде пайдаланылуы мүмкін жекелеген блоктарын және функцияларын (қайталанатын және қайталанбайтын) рәсімдеуге мүмкіндік береді. Функциялар мен рәсімдер кодтарындағы өзгерістер бағдарлама кодының басқа бөліктерін өзгертуге әкелмейді.

Сипаттама: тест модулі – пайдаланушының таңдауы тікелей өтетін басты модуль. Бұл модульді іске қосу пайдаланушыны тіркегеннен кейін жүзеге асырылады. Пайдаланушыны тіркеу модулі-пайдаланушы тізімді ашар алдында тіркеу үшін өзінің жеке деректерін толтыратын модуль. Оның деректері мен жіберілген өтінімнің нәтижесі таңдалғаннан кейін базаға енгізіледі. Анықтама модулі-БӨ пайдалану бойынша анықтама. Автор модулі-пайдаланушы туралы ақпарат береді. Өтінім нәтижелерін қалыптастыру модулі-осы модуль пайдаланушы өткен өтінімді қарау нәтижесін қалыптастырады. Пайдаланушылардың базасы және олардың нәтижелері-барлық пайдаланушылар туралы деректерді және олардың өтінімді қарау нәтижелері туралы мәліметтерді қамтитын база.

БӨ құру архитектурасын таңдау: Реккуренттік нейрожелі.

БӨ құрылымын өңдеу әдісін таңдау: Шығыс әдісі.

Бағдарламалау тілін таңдау: php.

Болжамды аудитория (түрі, жасы және т. б.): 18 жастан бастап.

БӨ жалпы көлемі, Мб: 25 Мб.

**Б қосымшасы**  
(міндетті түрде)  
Бағдарлама листингі

```
>>> import urllib2
>>> f=urllib2.urlopen('http://strelkacard.ru/api/cards/status/?cardnum=123456
78901&cardtypeid=3ae427a1-0f17-4524-acb1-a3f50090a8f3')
>>> print(f.read())
{"cardactive":false,"balance":100100,"baserate":3000,"cardblocked":false,"n
umoftrips":0}
>>> import requests
>>> CARD_TYPE_ID = '3ae427a1-0f17-4524-acb1-a3f50090a8f3'
>>> card_number='12345678901'
>>> payload = {'cardnum':card_number, 'cardtypeid': CARD_TYPE_ID}
>>> r = requests.get('http://strelkacard.ru/api/cards/status/', params=payload)
>>> print("Get info for card %s: %d %s" % (card_number, r.status_code,
r.text))
Get          info          for          card          12345678901:          200
{"cardactive":false,"balance":100100,"baserate":3000,"cardblocked":false,"numoftr
ips":0}
>>> r.json()
{'u'cardactive': False, u'balance': 100100, u'numoftrips': 0, u'baserate': 3000,
u'cardblocked': False}
>>> r.json()["balance"]
100100
!/usr/bin/python
import logging
import requests
logger = logging.getLogger(__name__)
def get_status(card_number):
    payload = {'cardnum':card_number, 'cardtypeid': CARD_TYPE_ID}
    r = requests.get('http://strelkacard.ru/api/cards/status/', params=payload)
    logger.info("Get info for card %s: %d %s" % (card_number, r.status_code,
r.text))
    if r.status_code == requests.codes.ok:
        return r.json()
    raise ValueError("Can't get info about card with number %s" %
card_number)
def get_balance(card_number):
    r = get_status(card_number)
    return r["balance"]/100.
>>> import checker
```

*Б қосымшасының жалғасы*

```
>>> checker.get_balance('12345678901')
1001.0
logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.INFO)
#!/usr/bin/python
from telegram import Updater, User
import logging
import checker
# Enable Logging
logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.INFO)
logger = logging.getLogger(__name__)
logger.setLevel(logging.DEBUG)
# Dictionary to store users by its id
users = {}
class UserInfo:
    def __init__(self, telegram_user):
        self.user = telegram_user
        self.cards = {}
    def add_card(self, card_number):
        card = CardInfo(card_number)
        self.cards[card_number] = CardInfo(card_number)
class CardInfo:
    def __init__(self, card_number):
        self.card_number = card_number
    def balance(self):
        logger.info("Getting balance for card %s" % self.card_number)
        json = checker.get_status(self.card_number)
        return json['balance']/100.
def log_params(method_name, update):
    logger.debug("Method: %s\nFrom: %s\nchat_id: %d\nText: %s" %
        (method_name,
         update.message.from_user,
         update.message.chat_id,
         update.message.text))
def help(bot, update):
    log_params('help', update)
    bot.sendMessage(update.message.chat_id
```

*Б қосымшасының жалғасы*

```
, text="""Supported commands:
/help - Show help
/addcard - Add a card to the list of registered cards
/removecard - Remove a card to the list of registered cards
/getcards - Returns balance for all registered cards""")
def add_card(bot, update, args):
    log_params('add_card', update)
    if len(args) != 1:
        bot.sendMessage(update.message.chat_id, text="Usage:\n/addcard
1234567890")
    return
    card_number = args[0]
    telegram_user = update.message.from_user
    if not users.has_key(telegram_user.id):
        users[telegram_user.id] = UserInfo(telegram_user)
    user = users[telegram_user.id]
    if not user.cards.has_key(card_number):
        user.add_card(card_number)
        bot.sendMessage(update.message.chat_id, text="Card %s has been
added successfully" % (card_number))
    else:
        bot.sendMessage(update.message.chat_id, text="Card %s has been
already added. Do nothing" % (card_number))
def remove_card(bot, update, args):
    log_params('remove_card', update)
    if len(args) != 1:
        bot.sendMessage(update.message.chat_id, text="Usage:\n/removecard
1234567890")
    return
    card_number = args[0]
    telegram_user = update.message.from_user
    if not users.has_key(telegram_user.id):
        bot.sendMessage(update.message.chat_id, text="There are no cards
saved for you")
    return
    user = users[telegram_user.id]
    if user.cards.has_key(card_number):
        user.cards.pop(card_number)
        bot.sendMessage(update.message.chat_id, text="Card %s has been
successfully removed" % (card_number))
```



```
else:
    bot.sendMessage(update.message.chat_id, text="Card %s has not been
added. Do nothing" % (card_number))
def get_cards(bot, update):
    log_params('get_cards', update)
    telegram_user = update.message.from_user
    if not users.has_key(telegram_user.id) or
len(users[telegram_user.id].cards) == 0:
        bot.sendMessage(update.message.chat_id
, text="There are no saved cards for you. Please add a card by typing
/addcard CARD_NUMBER")
        return
    user = users[telegram_user.id]
    cards = user.cards
    response = ""
    for card in cards.values():
        if len(response) != 0:
            response += '\n'
        response += "Card %s balance: %.2f"%(card.card_number,
card.balance())
    bot.sendMessage(update.message.chat_id, text=response)
def main():
    updater = Updater(TOKEN)
    # Get the dispatcher to register handlers
    dp = updater.dispatcher
    # Add handlers for Telegram messages
    dp.addTelegramCommandHandler("help", help)
    dp.addTelegramCommandHandler("addcard", add_card)
    dp.addTelegramCommandHandler("removecard", remove_card)
    dp.addTelegramCommandHandler("getcards", get_cards)
    updater.start_polling()
    updater.idle()
if __name__ == '__main__':
    main()
$ python simple_strelka_bot.py
2016-03-06 12:02:11,706 - telegram.dispatcher - INFO - Dispatcher started
2016-03-06 12:02:11,706 - telegram.updater - INFO - Updater thread started
2016-03-06 12:02:26,123 - telegram.bot - INFO - Getting updates:
[645839879]
2016-03-06 12:02:26,173 - __main__ - DEBUG - Method: get_cards
```

From: {'username': u'username', 'first\_name': u'Firstname', 'last\_name':  
u'Lastname', 'id': 12345678}