

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Мамандығы 5B070400 – Есептеу техникасы және бағдарламалық  
қамтамасыз ету

Студент Мауленбай Серік

Тақырыбы: Жоғарғы оқу орындарындағы тіркеуші кеңсеге арналған  
ақылды чат бот

### ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ

Диплом жобасын жасаушы Мауленбай Серіктің алдына жоғарғы оқу орындарындағы тіркеуші кеңсеге арналған ақылды чат бот құру тапсырмасы қойылған.

Дипломдық жобада студенттердің сабақ кестесін өзгертуге электронды кезекке тұру және анықтама алудың өтініш берудің электронды нұсқасы қарастырылған.

Жұмыс барысындағы басты мәселелер: студенттердің анықтама алу үшін қажетті ақпараттарды талдау, тіркеуші кеңсенің жұмыс уақытын талдау, жобадағы бағдарламалық қамтамасыз етудің алгоритмдерін іске асыру, дамыту болып табылады. Бұл қазіргі заманауи ең өзекті және келешегі бар технологияларды қолданудағы маңызды мәселелердің бірі.

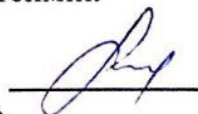
Менің пікірімше, диплом жазушы студент Мауленбай Серік алдына қойылған тапсырманы толығымен орындады және ақпараттық жүйелердің заманауи технологияларын меңгергендігін көрсетті.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейінде барлық талаптарға сәйкес деп есептей отырып Мауленбай Серікке 5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ

«Программалық инженерия»

кафедрасының лекторы «8» сәуір 2019 жыл

 А. К. Адильбекова



Университет:	Satbayev University
Название:	Серік Мауленбай.docx
Автор:	Серік Мауленбай
Координатор:	Алия Адильбекова
Дата отчета:	2019-05-03 17:17:32
Коэффициент подоби́я № 1: [?]	<b>8,9%</b>
Коэффициент подоби́я № 2: [?]	<b>7,9%</b>
Длина фразы для коэффициента подоби́я № 2: [?]	25
Количество слов:	4 267
Число знаков:	34 624
Адреса пропущенные при проверке:	
Количество завершенных проверок: [?]	1

Ғылыми жетекші \_\_\_\_\_  А. К. Адильбекова

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Мауленбай Серік Сағатұлы

«Жоғарғы оқу орындарындағы тіркеуші кеңсеге арналған ақылды чат бот»

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

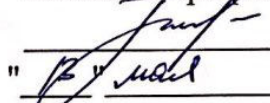
СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

ПИ кафедра меңгерушісі  
тех. ғыл. кандидаты, доцент,  
ассистент-профессор

 Р. Юнусов  
" 12 " маусым 2019ж.

**ТҮСІНДІРМЕ ЖАЗБА**

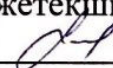
дипломдық жобаға

Тақырыбы: «Жоғарғы оқу орындарындағы тіркеуші кеңсеге арналған ақылды  
чат бот»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы

Орындаған

Мауленбай С. С.

Ғылыми жетекші тех. ғыл. магистрі,  
лектор  А.К. Адильбекова  
" 12 " маусым 2019 ж.

Алматы 2019



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

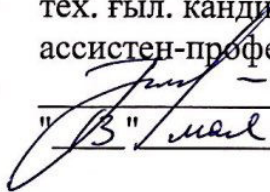
Программалық инженерия кафедрасы

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

**БЕКІТЕМІН**

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,  
ассистент-профессор

 Р. Юнусов  
"Б" мейл 2019 ж.

**Дипломдық жоба орындауға  
ТАПСЫРМА**

Білім алушы: Мауленбай Серік Сағатұлы

Тақырыбы: «Жоғары оқу орындарындағы тіркеуші кеңсеге арналған ақылды чат бот»

Университет академиялық мәселелер жөніндегі проректорының бұйрығымен  
«14» наурыз 2018ж. № 1841-б шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі «13» мамыр 2019 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Жоғарғы оқу орындарындағы тіркеуші кеңсеге арналған ақылды чат бот құру, деректерді тіркеу жүйелерін ұйымдастыру

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) негізгі бөлім;

б) жобалау бөлімі;

в) қолданылған бағдарламалық қамтамалар;

г) қолданбалы бөлім;

д) А Қосымшасы – техникалық тапсырма.

е) Б Қосымшасы – бағдарлама мәтіні.

Жобаның презентациялық 13 слайды ұсынылған.

Ұсынылған негізгі әдебиеттер: 10 атау.

**Дипломдық жобаны орындау  
КЕСТЕСІ**

<b>Бөлімдердің атаулары, зерттелген мәселелердің тізімі</b>	<b>Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі</b>	<b>Ескерту</b>
1. Диплом жұмысының жоспар-құрылымын құру.	14.01.2019	<i>орындалды</i>
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	<i>орындалды</i>
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім беру бойынша есеп беру жазбасын дайындау	01.02.2019	<i>орындалды</i>
4. Дипломның екінші бөлімі – жобалау сызбаларын дайындау.	15.02.2019	<i>орындалды</i>
5. Жобаны тестілеуден өткізу.	18.03.2019	<i>орындалды</i>
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2019	<i>орындалды</i>

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

<b>Бөлімдер атауы</b>	<b>Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)</b>	<b>Қолтаңба қойылған мерзімі</b>	<b>Қолы</b>
Нормалық бақылаушы	Таурбекова А. А. лектор	<i>04.05.19</i>	<i>[Signature]</i>
Бағдарламалық бөлім	Қалдыбеков С. Б. сениор-лектор	<i>06.05.19</i>	<i>[Signature]</i>

Ғылыми жетекші \_\_\_\_\_ *[Signature]* Адильбекова А. К

Тапсырманы орындауға қабылдап алған студент \_\_\_\_\_ *[Signature]* Мауленбай С. С

Күні

*«3» маусым 2018 ж.*

## Аңдатпа

Дипломдық жұмыс университет студенттерінің тіркеуші кеңсемен жұмыс істеуге жеңілдік беретін чат бот жүйесі. Ол университеттен анықтама алуға электронды ұсыныс тастау және сабақ кестесін ауыстыру үшін электронды кезекке тұру қызметін атқарады. Чат бот арқылы кезекті қадағалап студенттер өздерінің уақытын үнемдей алады. Бұл жүйені қолдану үшін арнайы қосымшаны жүктеудің қажеті жоқ. Тек телеграм мен ғаламтор болса жеткілікті.

Жоба үш бөлімнен тұрады: бірінші бөлімде ұқсас жүйелерге талдау жасалынып, екінші бөлімінде чат боттың жұмыс істеу принциптері UML-де жобаланып, деректер қоры құрылымы келтірілді; үшінші бөлімде қолданылған бағдарламалық қамтамалар бойынша есеп берілген.

Жобаның түсініктеме жазбасы жалпы 42 бет, оның ішінде 10 сурет, 2 міндетті қосымшадан тұрады. Жұмысты жазу кезінде 10 әдебиеттер мен мақалалар қолданылды.

## Аннотация

Дипломная работа система чат бот, позволяющая студентам университета работать с офисом регистратора. Он выполняет функции подачи электронных предложений на получение справки от университета и постановки на электронную очередь для замены расписания занятий. Студенты могут сэкономить время, чтобы отслеживать очередной путь через Чат бот. Для использования этой системы нет необходимости загружать специальное приложение. Достаточно, чтобы было только телеграмма и интернет.

Проект состоит из трех разделов: в первой части анализируется аналогичная система, во второй части принципы работы чат бота проектируются в UML, приведена структура базы данных; в третьей части представлен отчет по используемому программному обеспечению.

Пояснительная записка проекта состоит из 42 страниц, из них 10 рисунков, 2 обязательных приложения. При написании работы использовались 10 литературы и статьи.



## **Annotation**

This chat bot system allows University students to work with the Registrar's office. It performs the functions of submitting electronic proposals for obtaining a certificate from the University and placing on the electronic queue to replace the schedule of classes. Students can save time to track the next path through the Chat bot. There is no need to download a special application to use this system. Enough to have only a telegram and the Internet.

The project consists of three sections: the first part analyzes a similar system, the second part of the principles of the chat bot are designed in UML, shows the structure of the database; the third part presents a report on the software used.

Explanatory note of the project consists of 42 pages, including 10 figures, 2 mandatory annexes. When writing the work used 10 literature and articles.

## МАЗМҰНЫ

КІРІСПЕ	8
1. Негізгі бөлім	9
1.1 Telegram	9
1.2 API технологиясы	9
1.3 Чат бот технологиясымен танысу	10
1.4 Ұқсас жобаларды шолу	15
2. Жобалау бөлімі	17
2.1 Use Case диаграммасы	17
2.2 Тізбек диаграммасы	18
2.3 Кооперация диаграммасы	19
2.4 Күй диаграммасы	20
3. Қолданылған бағдарламалық қамтамалар	22
3.1 Python	22
3.2 Visual code	23
3.3. Деректер қоры	25
3.4 Чат бот қосымшасын құру	27
Қорытынды	31
Пайдаланылған әдебиеттер тізімі	32
А қосымшасы. Техникалық тапсырма	33
Б қосымшасы. Бағдарлама мәтіні	35



## КІРІСПЕ

Дипломдық проектінің өзектілігі жоғарға оқу орнының студенттердің анықтама алуын және сабақ кестесін өзгертуге электронды кезекке тұру болып табылады. Проекттің қызметі мобильді мессенджер, телеграм бот арқылы жүзеге асырылатын болады. Бот клиенттермен байланысқа түсіп, келген сұраныстарға жауап қайтару қызметін атқарады. Осы арқылы студенттер өз уақыттарн үнемдей алатын болады.

Дипломдық жобаның мақсаты:

- жоғарғы оқу орнының студенттердің уақытын үнемдеу;
- жоғарғы оқу орнындағы тіркеуші кеңсенің жұмысын жеңілдету.

Чат бот – бұл мәтінді немесе дауысты пайдаланып, адамның қарым-қатынасын бейнелейтін сөйлесу бағдарламасы. Чат боттар берілген алгоритмде жұмыс істеу арқылы тапсырмаларды автоматтандыруға көмектесе алады.

Осының арқасында, боттар мессенджерден шықпай-ақ пайдалануға болатын қалта көмекшілері болды. Олар жылдам командалардың көмегімен қарапайым тапсырмаларды шешуге мүмкіндік береді, барлық осы бағдарламалар орнатуды қажет етпейді және сіздің девайсыңыздың жадында жеке орын алмайды.

Сөйлесу боттары телефон жадысын жүктейтін көптеген қосымшаларды алмастырады. Чат-бот – бұл әлеуметтік желі, хабарлар мен хабарландырулар жіберу және қажет болса веб-сайт қызметі деп сенімді айта аламыз. Сөйлесу бот – бұл мәтінді немесе дауысты пайдаланып, адамның қарым-қатынасын бейнелейтін сөйлесу бағдарламасы. Сөйлесу боттары берілген алгоритмде жұмыс істеу арқылы тапсырмаларды автоматтандыруға көмектесе алады.

Чат боттарында бірнеше артықшылық бар:

- бір терезеде жұмысты ыңғайлы ету. Өртүрлі қойындылар мәзірі арасында ауысудың қажеті жоқ;
- сөйлесу боттар аз қозғалысты пайдаланады және Интернет жылдамдығы төмен болса да жұмыс істей алады.



## **1 Негізгі бөлім**

### **1.1 Телеграм**

Телеграм – көптеген форматтағы хабарламалар мен медиа файлдарды алмасуға мүмкіндік беретін кросс-платформа хабаршысы. Павел Дуровты қаржыландырған бірнеше американдық және неміс фирмаларының қуаттылығымен жұмыс істейтін жабық коды бар жеке меншік сервер бөлігі болып табылады. Ол жыл сайын шамамен \$13 млн GNU GPL лицензиясы бойынша [1].

Қызметтің ай сайын белсенді пайдаланушылары 2018 жылдың наурыз айының соңында 200 миллионнан астам адамды құрайды. 2017 жылдың тамызында «Телеграм» телеарнасында Павел Дуровтың хабарлауынша, тұтынушылардың саны күн сайын 600000-нан асып отырады екен.

Ромир зерттеу нәтижесіне сүйінетін болсақ, 2018 жылдың ақпан айына дейін ресейлік телеграмма пайдаланушылары тәулігіне 10-11 минут жұмсап отырады. Пайдаланушылардың көп бөлшегі 18-24 жастағы ресейлік жасөспірімдерге тиесілі. Мәскеуде Telegram жалпы алғанда екі есе танымал, әсіресе 35-44 жастағы аудитория арасында көп суранысқа ие болып табылады.

Диалогтар мен топтардағы стандартты хабар алмасудан басқа сіз файлдардың шектеусіз санын сақтауға, арналарды (микроблогтарды) сақтауға, хабарлаушыға боттарды жасауға және пайдалануға мүмкіндік береді.

### **1.2 API технологиясы**

Telegram Bot API құжаттамасында екі диаметрльды бөлінеді жаңартуларды алудың қарама-қарсы жолдары: көмегімен немесе вебхуктер орнату [6]. Кіріс жаңартулары сервер оны өңдегенше сақталады, бірақ 24 сағаттан артық емес. Жаңартуларды алу тәсіліне қарамастан, жауап ретінде нысанды аламыз JSON сериалданған Update.

Бірінші және ең қарапайым нұсқасы-мерзімді жаңа ақпараттың болуы тұрғысынан Telegram серверлерін сұрау болып табылады. Мұның бәрі Long Polling арқылы жүзеге асырылып отырады. Қысқа уақыт және барлық жаңартулар бірден ботқа жіберіледі.

Қарапайым, дегенмен өте сенімді емес. Біріншіден, Telegram серверлері мезгіл-мезгіл қатесін қайтарып отырады, себебі кейбір боттар сатыға түседі. Екіншіден, бір уақытта бірнеше іске қосылған болса боттар, қателіктерге тап болу ықтималдығы күшейеді. Вебхуктер кішкене басқаша жұмыс істейді. Вебхук орнатумен енді чат хабарлама келсе, онда Telegram өзі бұл туралы айтады. Қажеттілік болмаса мезгіл-мезгіл серверлер сұрайды, сол арқылы боттардың құлауының себебі жоғалады. Алайда бұл боттарды іске қосу

жоспарланып отырған машинаға толық веб-серверді орнату қажеттілігін төлеуге тура келеді. Сол үшін жұмыс жеке SSL сертификаты болуы керек (Secure Sockets Layer), өйткені Telegram тек HTTPS бойынша жұмыс істейді.

Telegram Bot API-мен жұмыс істеу үшін құжаттама зерттелді, онда барлық әдістер мен берілетін параметрлер сипатталған, барлық жауаптар JSON форматында. Сұхбат-ботты жазу барысында сыналған және келесі әдістер мен түрлері қолданылды:

– Getupdates әдісі арқылы жаңартуларды алу үшін пайдаланылады long polling. Жауап Update объектілерінің массиві түрінде қайтарылады.

Бот API-ның ең ерекше ерекшеліктерінің бірі пернетақталарды реттеуге арналған. Сервер жауап жіберген кезде алдын ала анықталған жауап опциялары бар арнайы пернетақтаны көрсету үшін пәрменді жіберуге болады (ReplyKeyboardMarkup қараңыз). Хабарламаны алғаннан кейін «Telegram» клиенті пайдаланушыға пайдаланушы пернетақтасын көрсетеді. Пернені басу дереу серверге тиісті пәрменді жібереді. Осылайша, роботты пайдаланушымен өзара әрекеттесуді едәуір жеңілдетуге болады.

### **1.3 Чат бот технологиясымен танысу**

Чат бот – хабардың ішінде жұмыс істейтін бағдарлама болып табылады. Мұндай бағдарлама сұрақтарға жауап беруге және оларды өз бетінше сұрауға қабілетті. Сөйлесу боттар жалпы мәселелерді шешу үшін әр түрлі салаларда қолданылады [7].

Бизнес Insider болжамына сәйкес, 2020 жылға қарай компаниялардың 80% -ы сұхбат боттарын пайдаланады.

Сөйлесу боттарды қолдау қызметтерінде пайдаланып, парольді өзгерту сияқты қарапайым мәселелерді шешуге көмектеседі.

Сұхбат боттарын ақпарат іздеу үшін пайдалануға болады. Мысалы, ауа-райы болжамы, оқиғалар плакаты, және т.б. Телеграммалық хабаршыда ауа райы туралы сөйлесетін немесе сыйлық таңдауға көмектесетін он мыңдаған боттарды табуға болады.

Саяхат индустриясы сұхбат боттарын алғаш пайдаланатындардың бірі болды. Шақыру боттары іздеу сұрауларына және пайдаланушы артықшылықтарына негізделген бағыттар, рейстер, мейрамханалар ұсынуы мүмкін. Сатып алғаннан кейін бағдарлама жиі қойылатын сұрақтарға жауап беру арқылы тұтынушыларды қолдауды қамтамасыз етеді.

Сөйлесу боттары жұмысқа орналасу және жұмысқа орналастыру кезінде жұмыс берушілер мен жұмыс іздеушілерге көмектесе алады. 2016 жылы жаз айында FirstJob жұмыс іздестіру агенттігі тек қана бос орындарды ұсынып қана қоймай, сондай-ақ бос лауазымға үміткермен әңгімелесуді жүргізетін «Мия» бот-боттарын шығарған болатын. Ресейде Связной компаниясының Superjob компаниясынан сұхбат жүргізді.

Маркетингші Эстер Крабдфорд EtherBot сұхбат бетін жасады, ол стипендияның өкілдеріне ханымның мансаптары, оның қабілеттері мен қызығушылығы туралы айтып берді.

Сөйлесу боттарын жеке көмекші ретінде пайдалануға болады. Хештагты ойлап тапқыш Крис Messina Facebook-да жазғандардың барлығына жауап беретін, MessinaBot бот көмекшісі пайдаланады.

Боталардың телеграммадағы жұмыс механикасы программистерге түсінікті, бірақ біз оны түсіндіруге тырысамыз. Сонымен, бет бұл, шын мәнінде, тек "рупор және рычагтар", ол арқылы сіздің жеке серверде әр түрлі бағдарламалау тілдерінде жазылған жеке бағдарлама жұмыс істейді. Сондықтан бота арқылы осындай бағдарламаны басқару жүріп жатыр деп айту дұрыс болар еді.

Пайдаланушы мен бот арасындағы өзара іс-қимыл келесідей:

– бот пайдаланушысы оған команданы береді - > Бот сіздің серверіңізге команданы береді - > бағдарлама сіздің серверіңізде боттан алынған сұранысты өңдейді - > Сервер ботқа жауап береді - > Бот пайдаланушыға қосымша экранына жауап береді.

Және бұл цикл түймелерге басқанда және кез келген жеделхат-ботпен өзара әрекеттескен кезде бір рет қайталанады.

Сөйлесу бот – бұл мәтінді немесе дауысты пайдаланып, адамның қарым-қатынасын бейнелейтін сөйлесу бағдарламасы. Сөйлесу боттары берілген алгоритмде жұмыс істеу арқылы тапсырмаларды автоматтандыруға көмектесе алады. Олар пайдаланушымен диалогқа қатысады, сұраныстарын қанағаттандырады, сұрауларға жауап береді немесе олардың жауаптарымен көңіл көтереді. Алғашқы бағдарламалар 1966 жылы пайда болды. Элисаньң виртуалды әңгімелесушісі терапевтпен диалогты жеткілікті түрде дәлелдеді. 2010 жылы жылдам хабаршылардың танымалдығы артып келе жатқанда, сұхбат боттары жаңа өмір алды. Олардың көпшілігі танымал хабаршы платформаларында жұмыс істейді: Facebook Messenger, Telegram, Viber, VKontakte, Skype, Slack. Боттар жеке бағдарламалар ретінде жұмыс істей алады немесе іздеу жүйелерінің функционалдық мүмкіндіктеріне ене алады.

Сөйлесу боттары көп тапсырмалы болып табылады деуге болады. Пайдаланушының сұрауы бойынша олар пайдалы ақпарат береді: ауа райы болжамдары, айырбас бағамдары, плакаттар, сөздерді аудару, соңғы жаңалықтар, дұрыс ұшуды таңдауға, билет жазуға және таксиге тапсырыс беруге көмектеседі. Көмекші боттар алдағы оқиғаларды еске түсіреді. Пошта қызметі боттардан пакетті трек жолымен қадағалап отырады. Сөйлесу боттары кітаптарды, фильмдерді және музыканы табуға және жүктеуге қолайлы болып табылады. БАҚ сұхбат боттарын лездік хабаршылар арқылы материал мен жаңалықтарды оқырмандарға жеткізу үшін пайдаланады. Мұндай чат боттарында BBC, Forbes, The Wall Street Journal, Medusa, Kommersant сияқты ірі басылымдар бар. Банктердің және төлем жүйелерінің чат боттарының көмегімен қаржы транзакцияларын, ақша аударымдарын жүргізуге және теңгерімді тексеруге болады.

Медициналық боттар, мысалы, қытайлық іздеу қозғалтқышының Baidu әуенімен, науқаспен сұхбат алуға, дәрігермен кездесуге және маманға диагноз қоюдың симптомдарының тізімін бере алады. Оқу боттарында шет тілдерін үйренуге көмектеседі. Көптеген әйгілі сұхбат боттары көңіл көтеруі мүмкін. Олар онлайн ойындар, викториналар және жұмбақтар ретінде жұмыс, күлкілі суреттер, SIFCO, баға, әзілдер, жұлдыз және жорамал жіберіңіз, боттардан, тараптар ұзақ философиялық диалогтар асырылуы мүмкін.

Қазақстандық хабаршы DAR VIS интерфейсінде сөйлесу боттары сияқты көпфункционалды боттар да бар. Ол Қазақстанның бірнеше қалаларында ауа-райы болжамы, энциклопедиялық ақпараттар және іс-шаралар плакатын ұсынады. Бот сөзді немесе сөз тіркесін әлемнің 100 тіліне аударып алады. Банктік картаны чат боттарына қосу арқылы ұялы байланыстың ақысын төлеуге, Опау картасын толтыруға, онлайн ойындарда виртуалды валютаны сатып алуға, гүлдер тапсырысына (болашақта азық-түлік тапсырысына және тұрақ төлеміне қол жеткізуге болады) ақша аударуға болады.

Бот адамды алмастыра алмайды, бірақ ол көптеген қарапайым операцияларды автоматтандыруға, техникалық қолдау қызметтері мен қоңырау орталықтарын алмастыруға, танымал сұрауларға дер кезінде жауап беруге және арқаулық ақпарат беруге мүмкіндік береді. Бот бизнеске көмек ретінде сату бойынша көмекші ретінде жұмыс істей алады. Пайдаланушымен диалог барысында бағдарлама опциялардың тізімін бірте-бірте қысқартатын ең өзекті өнімдер мен қызметтерді ұсынады. Боттар заң кеңесімен жақсы жұмыс істейді. Компанияларда сұхбат боттары HR және заң бөлімшелерінің кейбір функцияларын орындай алады. Олар бос лауазымдарға өтінімдерді қабылдап, үміткерлерді іздестіруге, қызметкерлердің кестесін жоспарлауға және маңызды тапсырмаларды еске салуға, сапарларды ұйымдастыруға, курстарға тіркелуге және стандартты есептер алуға мүмкіндік алады.

Сөйлесу боттарын қарапайым және өзін-өзі үйренуге болады. Біріншісі түймені және кескіндерді пайдаланып пайдаланушымен өзара әрекеттесе алады немесе мәтіндік пәрмендерді алады. Бот мәтінді немесе сөйлеуді таниды, негізгі сөздерді бөліп көрсетіп, оларды топтарға аударады. Содан кейін ол осы бағдарламаға сәйкес жауап береді. Мұндай сұхбат боттарында көбінесе командалар мен негізгі сөздер тізімі бар нұсқаулық бар. Боттардың осы түріне DAR VIS кіреді. Сөйлесу боттарының екінші түрі жасанды интеллект принципі бойынша жұмыс істейді, пайдаланушылармен қарым-қатынас жасауда, диалогтарды талдауда және пайдаланушылардың жеке ерекшеліктеріне түзетулермен айналысады. Осы қағидаға сәйкес, Apple-дан Siri және Yandex-тен Алиса дауыстық көмекшілері жұмыс істейді. Сөйлесу боттарымен дерексіз тақырыптар бойынша диалог жүргізе аласыз.

Чат боттарында бірнеше артықшылық бар:

– бір терезеде жұмысты ыңғайлы ету. Әртүрлі қойындылар мәзірі арасында ауысудың қажеті жоқ;

– сөйлесу боттар аз қозғалысты пайдаланады және Интернет жылдамдығы төмен болса да жұмыс істей алады.



Боттар барлық платформаларда хабаршылармен жұмыс істейді. Сөйлесу боттарын жасаушылар Android және iOS үшін жаңа қосымшаларды әзірлеуді қажет етпейді және пайдаланушылар оларды жүктеп алудың қажеті жоқ.

Сөйлесу боттарынан ақпаратты: телефон нөмірлерін, суреттерді, геодтарды, жол кодтарын, фотосуреттерді және бейне файлдарды тасымалдауға болады. Оларды өңдеу арқылы бот көптеген тапсырмаларды орындай алады, мысалы, ең жақын кафені немесе қонақүйді жіберілген геотагына жауап ретінде таба алады.

Бір топ адамдар бір уақытта сұхбат боттарымен жұмыс істей алады. Боттың DAR VIS бірегей ерекшелігі бар – «Жалпы қазандық». Пайдаланушылар ақша тауып, бірлескен сатып алуларға немесе шығындарға ақы төлей алады.

Боттар – хабарларды автоматты түрде өңдеу және жіберу үшін жасалған арнайы Telegram шоттары. Пайдаланушылар жүйелі немесе топтық сөйлесулер арқылы жіберілген хабарларды пайдаланып, боттармен өзара әрекеттеседі. Боттың логикасы HTTPS сұранымдары арқылы біздің бот API-імізге басқарылады.

Боттарды пайдалану мысалдары келтірілген:

- басқа қызметтермен интеграциялау. Мысалы, бот түсініктемелерді жіберу немесе ақылды үйді басқара алады. Немесе, мысалы, әрекетті немесе оқиғаны орындаған кезде ескертулерді жіберіңіз (мысалдар: GitHub боткасы, Image Bot);

- коммуналдық құралдар мен құралдар. Бот ауа-райы көрсетеді, мәтінді аударады немесе сұрауыңыз бойынша алдағы оқиғалар туралы ескертеді;

- бір және бірнеше ойындар. Бот сізбен дойбы немесе шахмат ойнайды, викториналар өткізеді және т.с.с.;

- әлеуметтік қызметтер. Робот сіздің ортақ мүдделеріңізге және қызығушылығыңызға негізделген әңгімелесуші таба алады. (Мысалы: HotOrBot);

- қалаған нәрсе. Боттарды кез келген нәрсеге бағдарламалауға болады.

Жоғарыда аталып өткендей, роботтар – олар жасаған кезде телефон нөмірін талап етпейтін арнайы шоттар. Шын мәнінде, бұл тіркелгілер қашықтағы серверде жұмыс істейтін қызметіңізге арналған интерфейстің рөлін атқарады.

Роботтар туралы ең қызықты нәрсе – оларды жасау үшін сізге төмен деңгейлі MTPROTO және шифрлау әдістерін үйренудің қажеті жоқ, роботпен байланыс әдеттегі HTTPS интерфейсі арқылы оңайлатылған Telegram API әдістерімен ұйымдастырылады. Біз оны Bot API деп атадық.

Роботтар онлайн режимінде және онлайн режимінде емес, орнына «робот» сөзі көрсетіледі.

Роботтар үшін серверлерде шектеулі орын бөлінеді - барлық хабарламалар өңделгеннен кейін белгілі бір уақыттан кейін жойылады.

Роботтар өздері пайдаланушымен байланысқа кіре алмайды. Пайдаланушы роботты топқа қосу керек немесе онымен диалогты бастау үшін

бірінші болып табылады. Ол үшін telegram.me/ <bot\_username> пішінінің сілтемелерін немесе пайдаланушы атымен іздеуге болады.

Роботтың пайдаланушысы «ботта» аяқталуы керек (мысалы, @TriviaBot, @gadalka\_bot).

Конференцияға қосылған кезде, әдепкі бойынша, робот барлық хабарларды қабылдамайды (құпиялылық режимін қараңыз).

Роботтар ешқашан ұйықтамайды, тамақ ішпейді немесе шағымданады (егер оларды керісінше бағдарламалай берсеңіз).

Телеграмма роботтарында көптеген ерекше функциялар бар – мысалы, арнайы пернетақта, әдепкі командаларға арналған қосымша интерфейстер, сыртқы байланыстар және топтарға арналған құпиялылық режимдері.

Пайдаланушылар өзіңіздің ботыңызбен кез келген сөйлесуден немесе топтан тікелей хабар алмасу өрісімен сөйлесуге болады. Мұны істеу үшін боттың пайдаланушы атын және іздеу сұрауын жазыңыз.

Сұрауды алғаннан кейін, бот кез келген нәтижені қайтара алады. Пайдаланушы біреуін басқанда, ол бірден сөйлесуге шығады. Осылайша, сұхбат бөлмелері, топтар немесе арналардағы ботты мазмұннан сұрауға болады.

Бот API-ның ең ерекше ерекшеліктерінің бірі пернетақталарды реттеуге арналған. Сервер жауап жіберген кезде алдын ала анықталған жауап опциялары бар арнайы пернетақтаны көрсету үшін пәрменді жіберуге болады (ReplyKeyboardMarkup қараңыз). Хабарламаны алғаннан кейін «Telegram» клиенті пайдаланушыға пайдаланушы пернетақтасын көрсетеді. Пернені басу дереу серверге тиісті пәрменді жібереді. Осылайша, роботты пайдаланушымен өзара әрекеттесуді едәуір жеңілдетуге болады.

Қазіргі уақытта эмодзи және мәтін кілтті көрсету үшін пайдаланылуы мүмкін. Мынадай пернетақталардың мысалдары:

Пайдаланушылардың боттармен жұмыс істеуін жеңілдету үшін барлық әзірлеушілерден бірнеше қарапайым пәрмендерге қолдау көрсетуді сұраймыз. Telegram қолданбасының интерфейсінде осы командаларға арналған жылдам сілтемелер (жылдам сілтемелер) болады.

/start – пайдаланушымен байланысқа шығады (мысалы, сәлемдесу хабары жіберіледі). Осы пәрменге қосымша аргументтерді де жібере аласыз (сыртқы байланыстыруды қараңыз).

/help – пәрменді пайдаланып хабарды көрсетеді. Бұл сіздің ботыңыз туралы қысқа хабарлама және қол жетімді пәрмендердің тізімі болуы мүмкін.

/Параметрлер – (мүмкіндігінше) оларды өзгертуге болатын ықтимал параметрлер мен пәрмендердің тізімін қайтарады.

Сіз роботпен байланысуды бастасаңыз, пайдаланушы Бастау түймешігін көреді. Боттың профиль бетінде Анықтама және Параметрлер сілтемелері де болады.

Telegram бот – жасанды интеллект функциялары бар бағдарламалық жасақтаманы басқаратын Telegram тіркелгісі. Олар басқа да қызметтерге үйрету, ойнау, іздеу, еске салу, басқа қызметтермен біріктіру немесе тіпті басқа қызметтерге командаларды беруі мүмкін.

Сөйлесу боттары – әлеуметтік қолданушылармен өзара әрекеттесудің танымал тәсілі. желімен және пайдаланушымен қарым-қатынас жасай алатын хабаршылармен қажетті ақпаратты ұсынып, әрі қарай әрекет ету нұсқаларын ұсынамыз.

Сөйлесу боттары телефон жадысын жүктейтін көптеген қосымшаларды алмастырады және жиі сөндіреді.

Чат-бот – бұл әлеуметтік желі, хабарлар мен хабарландырулар жіберу және қажет болса веб-сайт қызметі деп сенімді айта аламыз.

#### **1.4 Ұқсас жобаларды шолу**

Telegram-да қарапайым юзерлерден тек "bot" деген префиксі бар боттардың бір жалпы түрі қолданылады. Боттар өздері бірнеше бағытқа бөлінеді:

- чат-боттар. Пайдаланушы берген тақырыпқа қарым-қатынас имита;
- бот-информаторлар. Боттардың жеке түрі, олардың басты мақсаты – пайдаланушыны белгілі бір оқиғалар (жаңалықтар, іс-шаралар, Жарияланымдар және т.б.) туралы хабардар ету;
- ойын боттары. Түрлі ойындар ойнауға болатын боттар. Көп жағдайда, бұл әр түрлі ойындардың мәтіндік нұсқалары Combats.ru;
- боттар-асистенттер. Негізгі веб-нұсқаға қосымша ретінде түрлі онлайн сервистермен әзірленген боттар.

Шын мәнінде, нақты бөлу жоқ, өйткені кейбір боттар бірден бірнеше механиктерді қамтиды және көптеген пайдаланушы тапсырмаларды табысты орындайды. Олардың көмегімен аударуға, оқуға, тестілеуге, ақпаратты іздеуге, ойын ойнауға және тіпті басқа сервистерді пайдалануға және жаһандық желіге шығатын заттармен өзара әрекеттесуге болады (қазір танымал "интернет-заттар"). Жеделхаттағы барлық боттар тегін, бірақ 2017 жылы Павел Дуров осындай бағдарламаларды теңшеу және ақылы пайдалану мүмкіндігін жоққа шығарды.

Осының арқасында, боттар мессенджерден шықпай-ақ пайдалануға болатын қалта көмекшілері болды. Олар жылдам командалардың көмегімен қарапайым тапсырмаларды шешуге мүмкіндік береді, барлық осы бағдарламалар орнатуды қажет етпейді және сіздің девайсыңыздың жадында жеке орын алмайды.

Күн сайын 10 пайдалы жеделхат:

- @splittybot-шығындарды басқаруға көмектеседі;
- @getmusicbot - әрбір Меломан үшін қажетті музыкалық ойнатқыш;
- @delorean\_bot Бот еске салу. Оған хабарлама жазыңыз және күнін/уақытын көрсетіңіз, ол сізге еске салады;

- @thewinebot-сіздің виртуалды сомельніз сізді таң қалдыратын шарапты таңдауға көмектеседі;
- @movie\_bot-да, бірақ фильмдер туралы;
- @instasave\_bot-Инстаграмдағы кез келген фотосуретті жүктеуге мүмкіндік береді;
- @caffé\_bot и @barman\_bot;
- @zodiac\_bot — бот әлі де гороскоптарға сенетіндер үшін);
- @chgk\_bot-ойын әуесқойлары " не ? Қайда? Қашан?»;
- @channelanalyticsbot-кез келген жеделхаттың метриктерін талдауға мүмкіндік беретін бот.



## 2 Жобалау бөлімі

### 2.1 Use Case диаграммасы

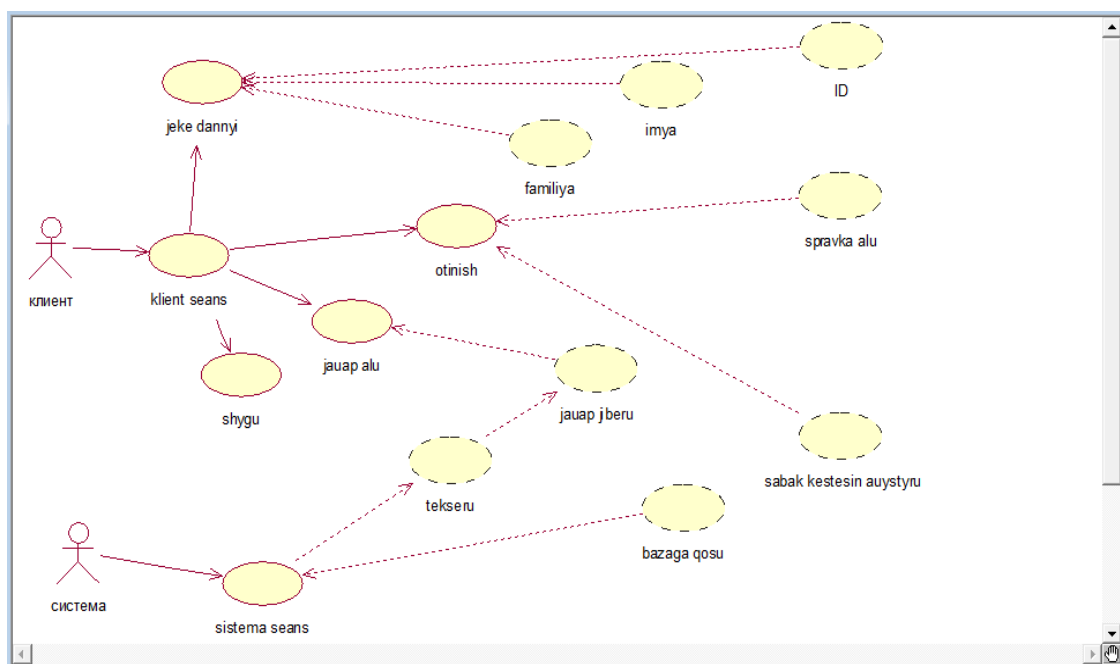
Use Case – бұл белгілі актерге (Actor) арналған, белгіленген бір нәтиже бере алатындай, бағдарлама жүйесімен орындалатын іс-қимылдар тізбегінің сипаттамасы. Ал, прецедент негізінен моделдегі берілген мәндердің тәртіптік структурасын жасайды, сонымен қатар прецеденттер кооперацияның қатысуымен таратылады [4].

Актер (Actor) – бұл берілген жүйедегі негізгі деген элементтермен байланыста болатын негізгі объектінің рөлі болып саналады. Актер мен қолданушының салыстырмалы түрде айырмашылығы келесідей: қолданушы жүйені толықтай пайдаланатын шынайы объект. Ол бірнеше рөлде ойнауы мүмкін болады, соған байланысты ол бір емес, бірнеше актер болуы мүмкін.

Пайдалану нұсқаларының диаграммасы (ағылш. UML-де актерлер мен прецеденттер арасындағы қатынастарды көрсететін және тұжырымдамалық деңгейде жүйені сипаттауға мүмкіндік беретін прецеденттер моделінің құрамдас бөлігі болып табылатын диаграмма.

Прецедент – модельдеуші жүйенің мүмкіндігі (оның функциясының бір бөлігі), оның арқасында пайдаланушы нақты, өлшенетін және оған қажетті нәтиже ала алады. Прецедент жүйенің жеке сервисіне сәйкес келеді, оны пайдалану нұсқаларының бірін анықтайды және пайдаланушының жүйемен өзара іс-қимылының типтік тәсілін сипаттайды. Пайдалану нұсқалары әдетте жүйеге қойылатын сыртқы талаптардың спецификациясы үшін қолданылады.

Use Case диаграммасы 2.2-суретте көрсетілген.



2.1-сурет – Use Case диаграммасы

## 2.2 Тізбек диаграммасын құру

Тізбек диаграммасы уақыт бойынша берілген объектілердің өзара ұйымдасу операцияларының жасалу ретін безендіреді. Ол жерде жоспарлармен қарастырылған мүмкіндіктердің орындалу процесіндегі өзара алмасатын объектілердің хабарлама тізбектерінің реттелген класстар және объектілер бейнеленеді [4].

Тізбектелген диаграммалар жалпы түрде Logical View пакетіндегі көзделген қолдану нұсқаларының жүзеге асырылуымен бірге белгіленеді. Тізбек диаграмма негізгі төрт элементтен тұрады:

- прецеденттегі артындағы мәтінінің іс-қимылы. Ол жоғарыдан төменге жазылады, және сол жақтан басталады. Сол терезеде іс-қимыл сипатталуы болып, орындалатын жұмыс уақытындағы белгілі ақпараттар қызмет етеді;

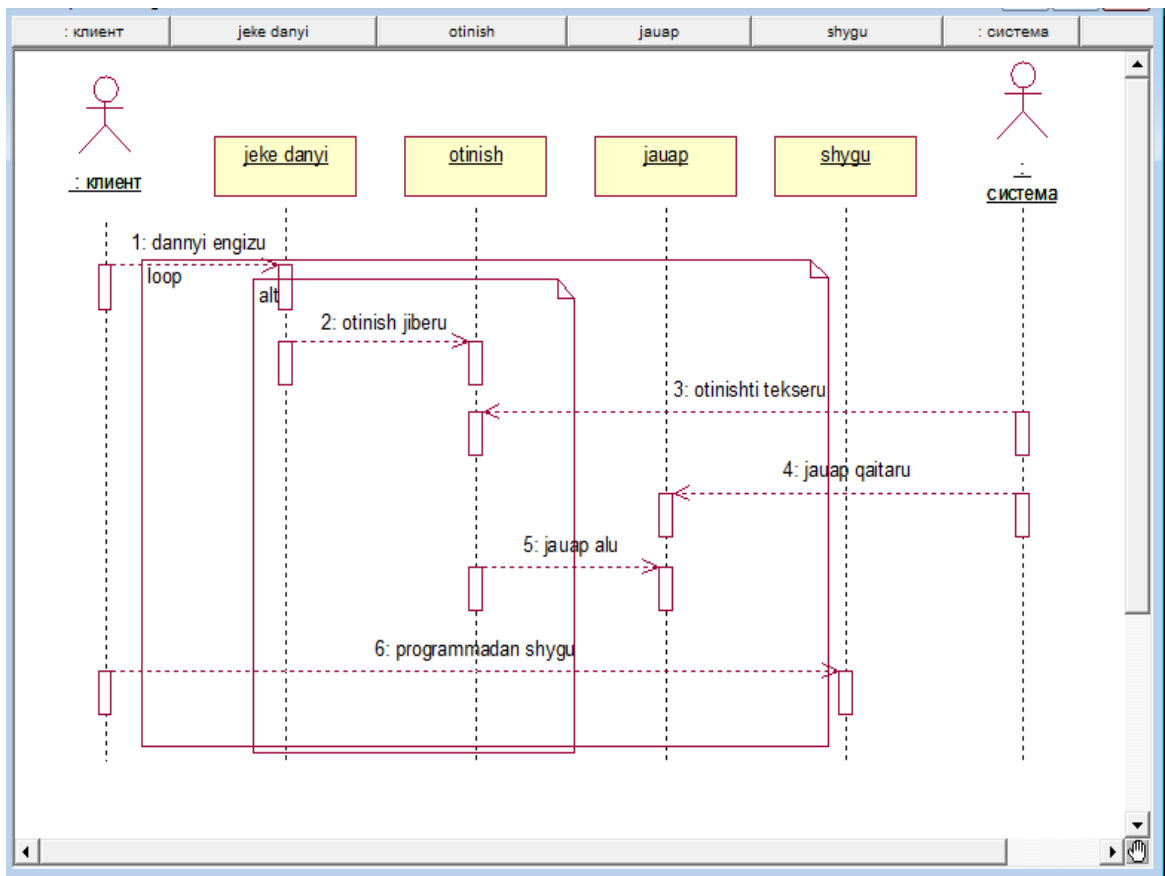
- объектілер "объект-класс" кейпінде аты болмаса объект данасының нөмірі және класс объектісінің атауы жазылады;

- хабарландыру, бір бағытта көрсетілген белгілі бір объектіден келесі объектіге бағытталған іс-қимыл туралы мәлімет хатынан тұрады. Берілген бір уақытта орындалуы мүмкін, және осы іс-әрекеттегі берілген жүйенің қайтарымды реакциясы болуы мүмкін болып келеді;

- әдістері (операциялар). Тікбұрыш түрінде болады. Олар үздік сызықпен бейнеленген. Яғни, берілген әдістерге кіретін сол объектілер болып табылады. Тік бұрыштың ұзындығы артынан қушыда басқару нүутесін көрсетуде қолданса болады: Тікбұрыш аяқталатын немесе бітетін әдіс толықтай нүктесіне дейін басқарумен игеріледі. Бұл, берілген үшбұрыштар циклге арналған объекті түзуі деп саналады.

Диаграмма (грек. (diagramma) - бейне, сурет, сызба) – графикалық мәліметтерді ұсыну желілік отрезками немесе геометриялық тұлғалармен, мүмкіндік беретін тез арақатынасын бағалау бірнеше шамаларды. Визуализация техникасының түрлі тәсілдерін қолдана отырып, ақпараттың геометриялық символдық бейнесі болып табылады. Кейде диаграммаларды ресімдеу үшін жазықтыққа бағытталған үш өлшемді визуализация қолданылады, бұл диаграммаға ерекшелік сызықтарын береді немесе ол қолданылатын аймақ туралы жалпы түсінік береді. Мысалы: ақша сомаларымен байланысты қаржылық диаграмма бумадағы купюралар немесе стопкадағы монеталар санын білдіреді; жылжымалы құрам санын салыстыру диаграммасы — бейнеленген поездардың әртүрлі ұзындығын және т. б. Өзінің көрнекілігі мен пайдалану ыңғайлылығының арқасында диаграммалар тек бухгалтерлердің, логистердің және басқа да қызметкерлердің күнделікті жұмысында ғана емес, сонымен қатар әр түрлі ұйымдардың клиенттері мен менеджерлері үшін презентациялар материалдарын дайындау кезінде де жиі қолданылады.

Тізбектер диаграммасы 2.2-суретте көрсетілген



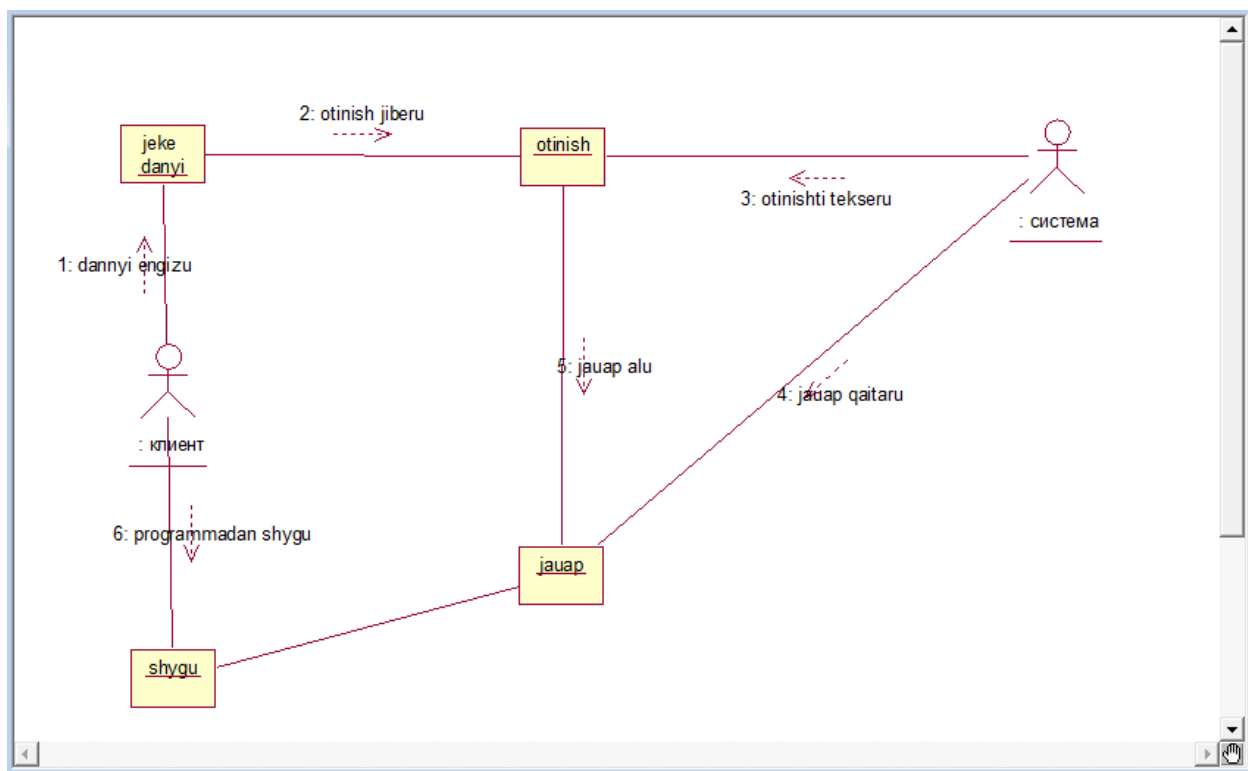
**2.2-сурет – Тізбек диаграммасы**

### 2.3 Кооперация диаграммасын құру

Кооперация диаграммасы дегеніміз – бұл хаттарды жіберетін және қабылдайтын субъектілерді құрылымдық жақтан жан-жағынан ұйымдасуын айқындайтын әрекеттесу диаграммасы. Кооперация диаграммалары жүйе жұмысының барысында субъектілердің өзара әрекеттесуін баяндайды. Мұндай диаграммалар жүйе жоспарының сценарийлерін бейнелейді [4]. Объект атыуның асты сызылады және әрдайым ұсынылады, ал қасиеттері таңдалып көрсетіледі.

Кооператив диаграммалар белгілі оқиғаларда пайдалы субъектілерде істелген өзгерістердің зардаптары керек бағалауда және қандай басқа субъектілерге әсер етеді.

Кооперация сызбалары жүйе жұмысы барысында субъектілердің өзара қарым-қатынасын бейнелейді. Мұндай сызбалар жүйе тәртібінің жоспарларын модельдейді. Кооперация диаграммасы 2.3-суретте көрсетілген.



2.3-сурет – Кооперация диаграммасы

## 2.4 Күй диаграммасын құру

Күй диаграммасы – нысанның бір қалпынан басқасына ауысуын айыратын, және іс-әрекеттері қалпыларын алмастырумен ескертілінген, оқиғалардың, болмаса хабарлаулардың объекттік жағдайларының графикалық түрдегі ұсынысы[4].

Диаграмма күйі – әр түрлі температурада (қысымда) қорытпаның фазалар күйінің тепендігін, олардың шоғырлануына байланысты етіп көрсететін диаграмма. Объектінің өз қалпын уақыт бойында қай бағытта өзгертетінін түсіну үшін жағдайдың өзгеруінің спецификациясы бар. Субъектінің жағдайы оның атрибуттарының, және қатынас атрибуттарының мағынасымен сипатталады.

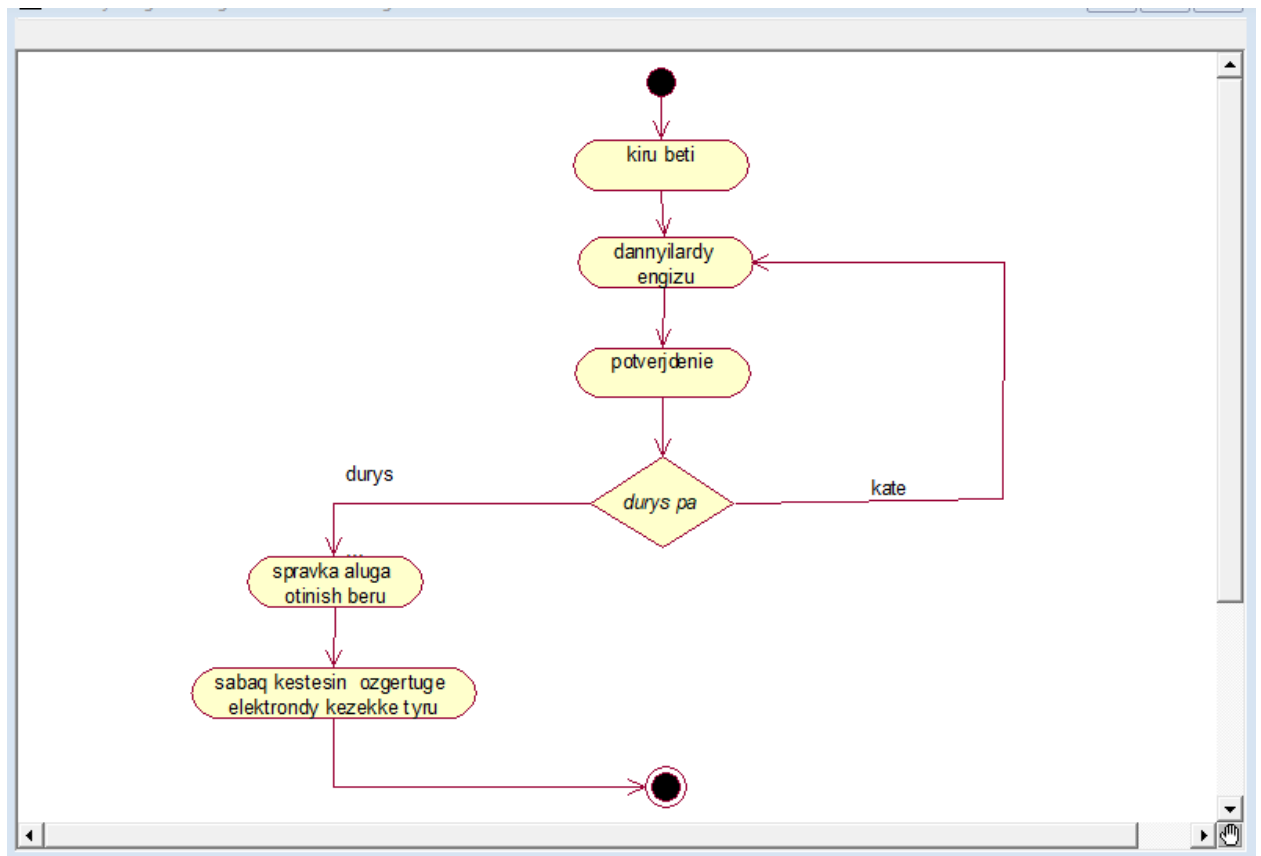
Күй спецификациясы класс атрибутын сипаттайды. Тәртіп спецификациясы класс оталарын анықтайды, олардың кейбірі нысан күйін басқаға өзгерте алады.

Объектілер жағдайларын моделдеу күйлер диаграммасының арқасында жүзеге асады. Күйлер сызбасы бұл күйлер мен өзгерістер графы. Күйлер моделі системаға маңызды кластар үшін сызылады.

Объектілер белгілі бір нүктеден шығып, соңғы нүктеге жеткенше барлық нүктелерден өтеді, және соңғы нүктеге жетеді.

Күй диаграммасы 2.4-суретте көрсетілген.





2.4-сурет – Куй диаграммасы

## 3 Қолданылған бағдарламалық қамтамалар

### 3.1 Python

Python – бұл жалпы мақсаттағы бағдарламалау тілі, бірінші кезекте бағдарламашының өнімділігін арттыруға, ол жазған кодқа қарағанда. Қарапайым тілмен айтқанда, Python іс жүзінде ештеңе жазу (веб-үстел қолданбалары, ойындар, автоматтандыру бойынша скрипттер, есептеудің кешенді жүйелері, жүйелер және т.б.) елеулі проблемаларсыз [2].

Сонымен қатар, кіру шегі төмен, ал код көбінесе қысқа және ньюм ешқашан жазмаған адамға да түсінікті. Кодтың қарапайымдылығы үшін,

Python-да жазылған бағдарламаларды одан әрі сүйемелдеу.

Java немесе C++ салыстырғанда оңайырақ және жағымды. Ал бизнес тұрғысынан бұл шығындарды қысқарту және өнімділікті арттыру қызметкерлердің еңбек.

Python интерпретаторы сөзсіз артықшылығы іс жүзінде барлық платформаларда және операциялық жүйелерде іске асырылған.

Бірінші осындай тіл болды Си, бірақ оның түрлі машиналарда деректер түрлері жадтың әртүрлі санын алып, бұл кейбір 5 шын мәнінде көшірілетін бағдарламаны жазу кезінде кедергі. Python осындай кемшілік жоқ.

Сондай - ақ, маңызды сызықтілдің кеңеюі, оған беріледі Гвидо ван Россумның өзі жазғандай, тіл болды кеңейтілген ретінде ойластырылған. Бұл мүмкіндік бар дегенді білдіреді, барлық мүдделі бағдарламашылармен тілді жетілдіру.

Интерпретатор с тілінде жазылған және бастапқы код кез келген үшін қол жетімді манипуляция. Қажет болған жағдайда, сіз оны өз бағдарлама және кіріктірілген қабық ретінде пайдалану.

Python және бағдарламаны компилирлеу арқылы оның толықтырулар, алу жаңа мүмкіндіктері бар" кеңейтілген " интерпретатор.

Келесі қадір-қасиетке қосылатындар санының көп болуы қамтамасыз ететін Модульдер мүмкіндіктер. Мұндай Модульдер Си және Python-да жазылады және мүмкін барлық білікті бағдарламашылар әзірлеген. В мысал ретінде келесі модульдерді келтіруге болады:

– Numerical Python – кеңейтілген математикалық мүмкіндіктер, бүтін векторлармен және матрицалармен манипуляциялар сияқты;

– Tkinter – графикалық сызықтарды пайдаланып қосымшаларды құру кең таралған негізіндегі пайдаланушы интерфейсі (GUI);

– X-Windows Tk интерфейсінде;

– OpenGL – көлемді графикалық кітапхана пайдалану Open Graphics library екі және үш өлшемді нысандарды модельдеу Silicon Graphics Inc. Бұл стандарт, оның ішінде осындай Microsoft Windows 95 OSR 2 сияқты кең таралған операциялық жүйелер 98 және Windows NT 4.0.

Автор байқаған жалғыз кемшілік – бұл Python бағдарламаны орындаудың салыстырмалы төмен жылдамдығы, бұл оның түсіндірілуіне байланысты. Дегенмен, бұл өте ақталады бағдарламаларды жазу кезінде тілдің артықшылықтары өте қиын емес орындау жылдамдығы.

Python (орыс тіліндегі жалпы аты piton) әзірлеуші өнімділігін және кодты оқуға қабілеттілігін арттыруға бағытталған жоғары деңгейлі жалпы мақсаттағы бағдарламалау тілі болып табылады. Python ядросының синтаксисі минималды болып табылады. Сонымен қатар, стандартты кітапхана көптеген пайдалы функцияларды қамтиды.

Python құрылымдық, объектілі-бағытталған, функционалды, императивті және аспект-бағдарланған бағдарламаларды қолдайды. Негізгі архитектуралық ерекшеліктер – динамикалық теру, автоматты түрде жадты басқару, толығымен инспекциялау, ерекшеліктерді өңдеу механизмі, көп айналымды есептеуді қолдау, жоғары деңгейдегі деректер құрылымдары. Бағдарламаларды модульдерге бөлу арқылы қолданады, олар өз кезегінде пакеттерге біріктірілуі мүмкін.

Python-тің тірек жүзеге асуы белсенді қолданылатын платформалардың көпшілігін қолдайтын CPython аудармашы болып табылады. Ол еркін Python Software Foundation лицензиясы бойынша таратылады, ол оны кез-келген қосымшаларда, соның ішінде меншік нысандарында шектеусіз пайдалануға мүмкіндік береді. JVM үшін компиляция, CLR, LLVM, басқа да тәуелсіз іске асыру мүмкіндіктері бар аудармашы жүзеге асырылады. PyPy жобасы JIT жинағын пайдаланады, бұл Python бағдарламаларының жылдамдығын едәуір арттырады.

Python – белсенді түрде дамып келе жатқан бағдарламалау тілі, тілдік қасиеттерді қосу/ауыстырудың жаңа нұсқалары екі жарым жылда бір рет пайда болады. Тіл ресми стандарттауға ұшырамаған, стандарттың рөлі шын мәнінде тіл авторының бақылауымен әзірленген CPython арқылы орындалады.

### **3.2 MySQL технологиясы**

MySQL – еркін дерекқорды басқару жүйесі. MySQL-ті Швеция MySQL AB-ді сатып алған Sun Microsystems-мен жұтып қойған сауда маркасына құқықты алған Oracle корпорациясы әзірледі және қолдады. Өнім GNU General Public License, сондай-ақ өзінің коммерциялық лицензиясы бойынша таратылады. Бұдан басқа, әзірлеушілер лицензияланған пайдаланушылар үшін теңшелетін функцияларды жасайды. Осы бұйрықтың арқасында репликалау механизмі ең ертерек нұсқаларында пайда болды [8].

MySQL – бұл шағын және орта қосымшаларға арналған шешім. WAMP, AppServ, LAMP серверлеріне және Denver, XAMPP, VertrigoServ серверлерінің портативті жинақтарына қосылған. Әдетте, MySQL жергілікті немесе қашықтағы клиенттерге қол жеткізетін сервер ретінде пайдаланылады, бірақ

тарату пакеті дербес бағдарламаларда MySQL-ты қосуға мүмкіндік беретін ішкі сервер кітапханасын қамтиды.

MySQL DBMS-дің икемділігі көптеген кесте түрлерінің қолдауы арқасында қолданады: пайдаланушылар толық мәтінді іздеуді қолдайтын MyISAM типтік кестелерді және жеке жазбалар деңгейінде операцияларды қолдайтын InnoDB кестелерін таңдай алады. Сонымен қатар, MySQL ДББЖ кестелердің жаңа түрлерін жасау принциптерін көрсететін EXAMPLE кестелерінің арнайы түрімен келеді. Ашық архитектура және GPL-лицензиялау арқасында MySQL-де кестелердің жаңа түрлері үнемі пайда болады.

2008 жылдың 26 ақпанында Sun Microsystems MySQL АВ-ді 1 млрд. Долларға сатып алды, 2010 жылдың 27 қаңтарында Oracle Sun Microsystems-ді 7,4 млрд. Долларға сатып алды және MySQL-ты ДББЖ желісіне қосқан.

MySQL әзірлеуші қоғамдастығында Drizzle, OurDelta, Percona Server және MariaDB секілді әртүрлі кодтар салалары жасалды. Осы филиалдардың бәрі Oracle компаниясы тарапынан Күнді басып алу кезінде болған.

MySQL-тің пайда болуы туралы төмен деңгейлі ескі ISAM пайдаланылды кестелер,: MySQL патенттелген компаниясы mSQL пайдалануға әрекет ретінде пайда. Нәтижесінде жаңа SQL интерфейсі әзірленді, бірақ API mSQL-дан мұраланған. «MySQL» атауының қайдан келгені анық емес. («Менің ....» ағылшын) немесе Менің қызы құрметіне (менің FIN.), Майкл Widenius жүйесін жасаушылардың бірі. «... Менің» компаниясының барлық дерлік жетістіктері префикс басталды, өйткені, не: Developers екі параметрлерін беріледі.

MySQL дельфинінің логотипі Сакила деп аталады. Ол пайдаланушылардың ұсынған «дельфиндік атауларының» үлкен тізімінен таңдалды. «Сакила» атауы Open Source әзірлеуші Амброуз Твебазеге жіберілді.

MySQL-те қос лицензиялау бар. MySQL GPL лицензиясының шарттарына сәйкес таратылуы мүмкін. Алайда, GPL шарттарына сәйкес, кез келген бағдарлама MySQL кітапханаларын (немесе басқа GPL кодын) қолданса, ол GPL лицензиясына сәйкес таратылуы керек. Бұл бағдарламалардың бастапқы кодын ашқысы келмейтін әзірлеушілердің жоспарларынан өзгеше болуы мүмкін. Мұндай жағдайларда коммерциялық лицензия беріледі, ол сондай-ақ сапалы қызмет көрсетуді қамтамасыз етеді. Тегін бағдарламалық қамтамасыз ету үшін, Oracle белгілі бір Oracle тізімінен лицензия бойынша таратылатын бағдарламалық жасақтамамен қатар, MySQL-ды пайдалану мен таратуды нақты рұқсат беретін ережелерге бөлек ерекшелік береді.

MySQL платформаларының көп саны: AIX, BSDi, FreeBSD, HP-UX, Linux, MacOS, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003, WinCE, Windows Vista, Windows 7 және Windows 10. OpenVMS-ге MySQL порты бар. Ресми ДББЖ веб-сайтына қотару үшін тек қана бастапқы кодтар ғана емес, сондай-ақ арнайы операциялық жүйелер үшін жинақталған және оңтайландырылған MySQL ДББЖ-ның дайын орындалатын модульдері де қарастырылғанын атап өту маңызды.

### 3.3 Visual Studio Code

Visual Studio Code – Microsoft, Windows, Linux және macOS үшін әзірленген бастапқы коды редакторы. Веб-платформаны және бұлт қолданбаларын әзірлеу үшін «жеңіл» код редакторы ретінде орналастырылған. Git синтаксисті бөлектеу, IntelliSense және құралдарды қайта өңдеу құралдары бар құралдарды қамтиды. Бұл теңшелім үшін ыңғайлы мүмкіндіктер: реттелетін тақырыптар, пернелер тіркесімі және конфигурация файлдары. Ол тегін таратылады, ашық бастапқы бағдарламалық жасақтама ретінде әзірленген, бірақ дайын құрылымдар меншік лицензиясымен бөлінеді [3].

Visual Studio коды Electron негізделген, Node.js пайдаланып Blink қозғалтқышында жұмыс істейтін үстел үсті қосымшаларын жасауға мүмкіндік беретін негіздеме. Редактор Electron негізделген болса да, ол Atom редакторын пайдаланбайды. Оның орнына, ол Visual Studio Online үшін әзірленген Monaco веб-редакторын орындайды.

Visual Studio коды – бастапқы кодының редакторы. Ол бірқатар бағдарламалау тілдерін, синтаксисті бөлектеуді, IntelliSense, рефакторинг, отладтау, кодты шарлау, Git қолдауы және басқа мүмкіндіктерді қолдайды. Visual Studio кодының көптеген ерекшеліктері графикалық интерфейс арқылы қол жетімді емес, олар пәрмендер немесе JSON файлдарының палитрасы арқылы жиі пайдаланылады (мысалы, пайдаланушы параметрлері). Пәрмен палитрасы - пернелер тіркесімі арқылы шақырылатын пәрмен жолы ұқсастығы.

Visual Studio сонымен қатар құжатты, жаңа жол таңбаларын және ағымдағы құжаттың бағдарламалау тілін сақтағанда код бетін ауыстыруға мүмкіндік береді.

2018 жылдан бастап Visual Studio Code үшін ашық бастапқы коды Python кеңейтімі пайда болды. Ол әзірлеушілерге кодты редакциялау, түзету және тестілеу үшін жеткілікті мүмкіндіктер ұсынады.

2019 жылдың наурыз айынан бастап өнімдегі ендірілген пайдаланушы интерфейсі арқылы бірнеше мың кеңейтімді «бағдарламалау тілдері» санатына ғана жүктеуге және орнатуға болады.

Visual Studio коды пайдалану деректерін (телеметрияны) жинайды және оны Microsoft-қа жібереді, бірақ бұл функция өшірілген (Файл>Параметрлер>Параметрлер опциясы, «телеметрия.жаселеметрия»). Деректермен қамтамасыз ету міндетті емес және жеке деректерді беруден бас тартса да, дербестендіру сияқты кейбір мүмкіндіктер мұндай деректерді пайдаланып өшіруге болмайды. Деректер құпиялылық туралы мәлімдемеге сәйкес Microsoft корпорациясының бақыланатын еншілес компанияларына, филиалдарына және құқық қорғау органдарына жіберілуі мүмкін.

### 3.4 Деректер қоры

Деректер қоры – бір-бірімен белгілі бір тәртіппен құрылымдық тұрғыдан байланысқан, машинамен оқылатын, нақты қолданбалы сала үшін көкейтесті, ақпараттық технология үшін жарамды нысандағы нақты дүние объектілері сипаттамаларының, сондай-ақ ЭЕМ-нің жадындағы немесе машинамен оқылатын сақтағыштағы (магнитті ленталардағы, дискеттердегі, ықшам дискілердегі) ақпарат жазбаларының жиынтығы. Деректер қоры қоғамның барлық ақпараттық қорларымен тығыз байланысты, солардың негізінде жасалады, ал олардың көбісі ақпараттың тек құжаттық көздерімен бірге пайдаланылады. Деректер қорын жасау үшін ең алдымен мүдделілік туғызған объектілер (мысалы, жеткізушілер, тұтынушылар, кәсіпорындар, қызметкерлер, т. б.) айқындалады, сонан соң әрбір объекті үшін оның өзгешеліктерін сипаттайтын деректердің жиынтығы бөліп көрсетіледі. Мысалы, кәсіпорынның өзгешеліктеріне оның атауы, мекен-жайы, банкідегі шоты, шығаратын өнімінің түрпаты, т.б. жатады [5].



### 3.4 Чат бот қосымшасын құру

Телеграмада ботты құру үшін бағдарламаның қандай мақсаттары орындалатынын анықтаймыз: хабарламаларға жауап беру, валютаны айырбастау немесе басқа функцияларды орындау [9]. Егер бағдарламалау тілін білмесеңіз-қиын емес. Қарапайым роботтарды онсыз да жазуға болады. Жұмысты бастау үшін, @BotFather-ге жазылыңыз және оны іске қосыңыз, әрі қарай – нұсқауларды орындаймыз:

Командалық жолда /newbot жазыңыз (жаңа бота жасайды).

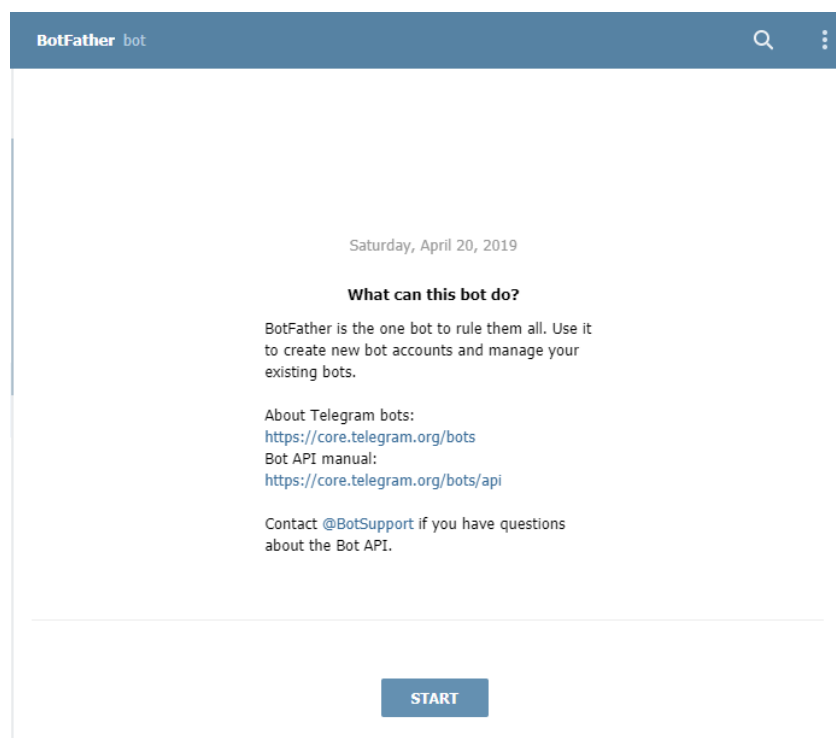
BotFather сіздің бала деп атауды ұсынғанға дейін күтіңіз. Кез келген атауды ойлап табыңыз, бірақ ол "bot" - ге аяқталуы керек екенін есте сақтаңыз. Бұл жерде сіздің бота бетіңізді (аватарды) қосып, оны сипаттауға болады.

BotFather бірегей белгіні алыңыз.

Оны кез келген мәтіндік файлға көшіріңіз және жоғалмауы үшін сенімді жерде сақтаңыз (бұл мүмкін емес, себебі белгіні есте сақтау қажет емес).

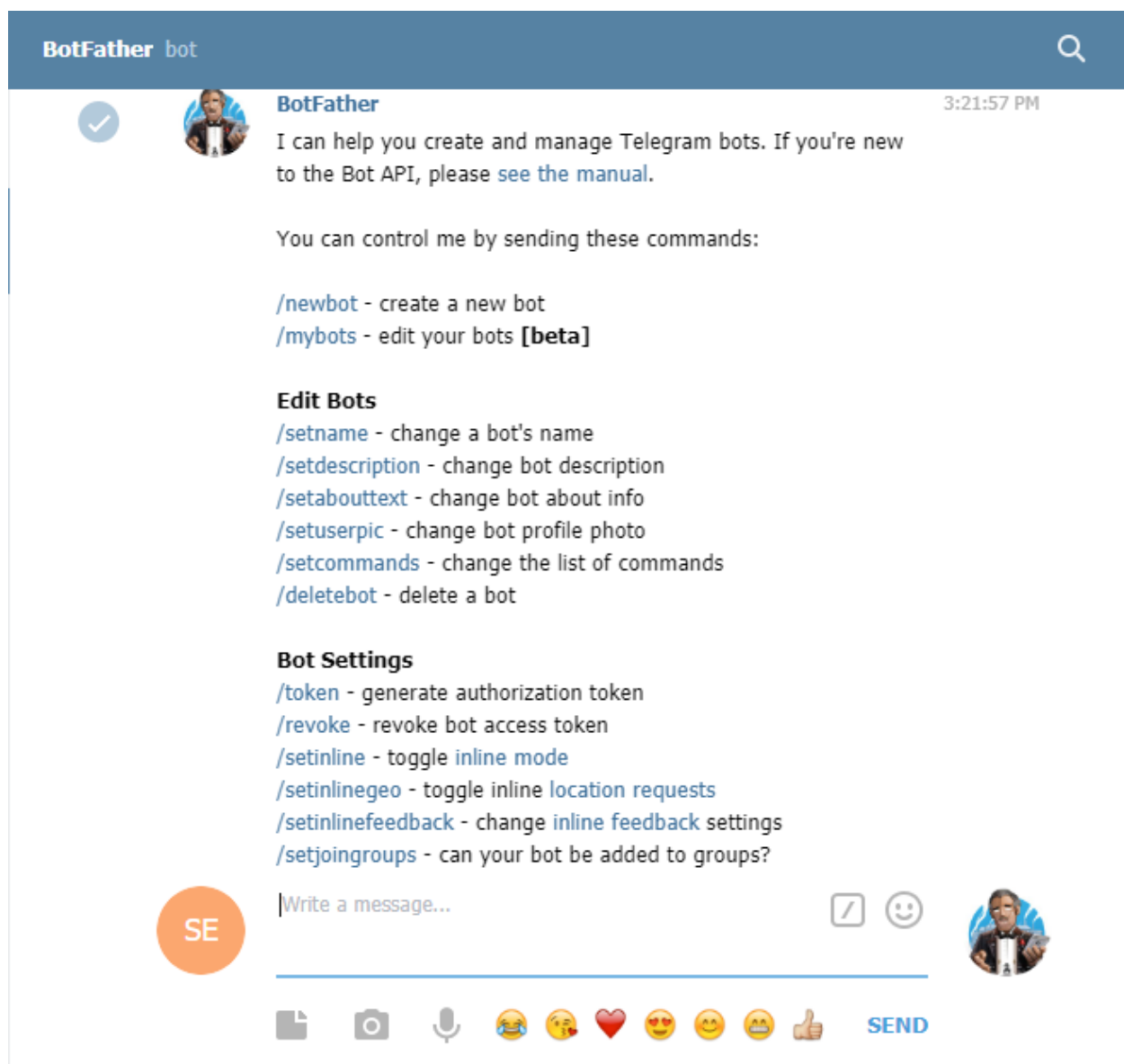
Бұл әрекеттер сізге өз боттарыңызды жасауға көмектеседі, бірақ оны қандай да бір функцияларды орындауға үйрету үшін сізге бағдарламалау тілінде код жазу немесе Raquebot сияқты арнайы бағдарламалардың мүмкіндіктерін пайдалану қажет болады. Бұл қызмет ерекше проблемаларсыз функционалдық роботтарды құруға көмектеседі.

Чат бот құру үшін алдымен телеграм әлеуметтік желісінен токен алу керек. Оны алу үшін Telegram әлеуметтік желісіндегі BotFather ботына жүгіндік. Ол 3.1-суретте көрсетілген.



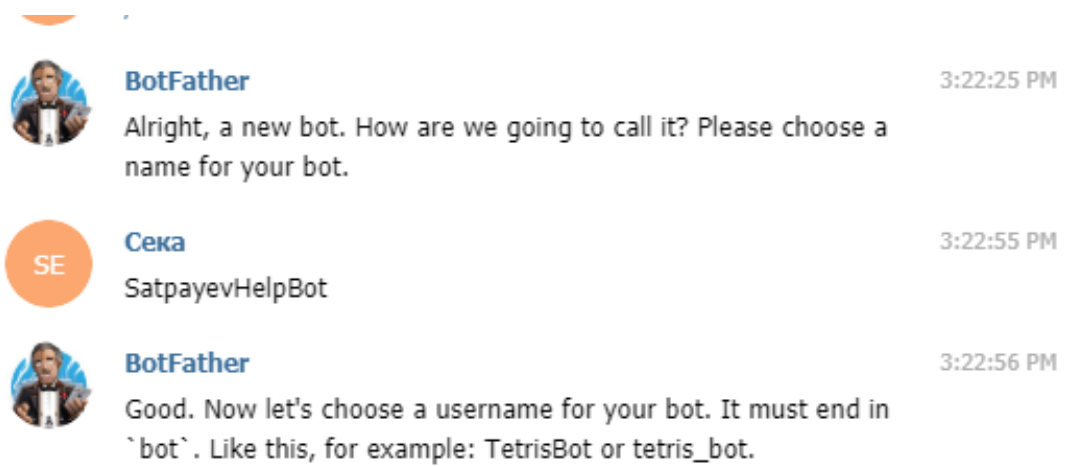
3.1-сурет – BotFather-дің бастапқы беті

BotFather боты көптеген қызметтер ұсынады. Соның арасындағы бізге керегі жаңа бот функциясы. Ол 3.2-суретте көрсетілген.



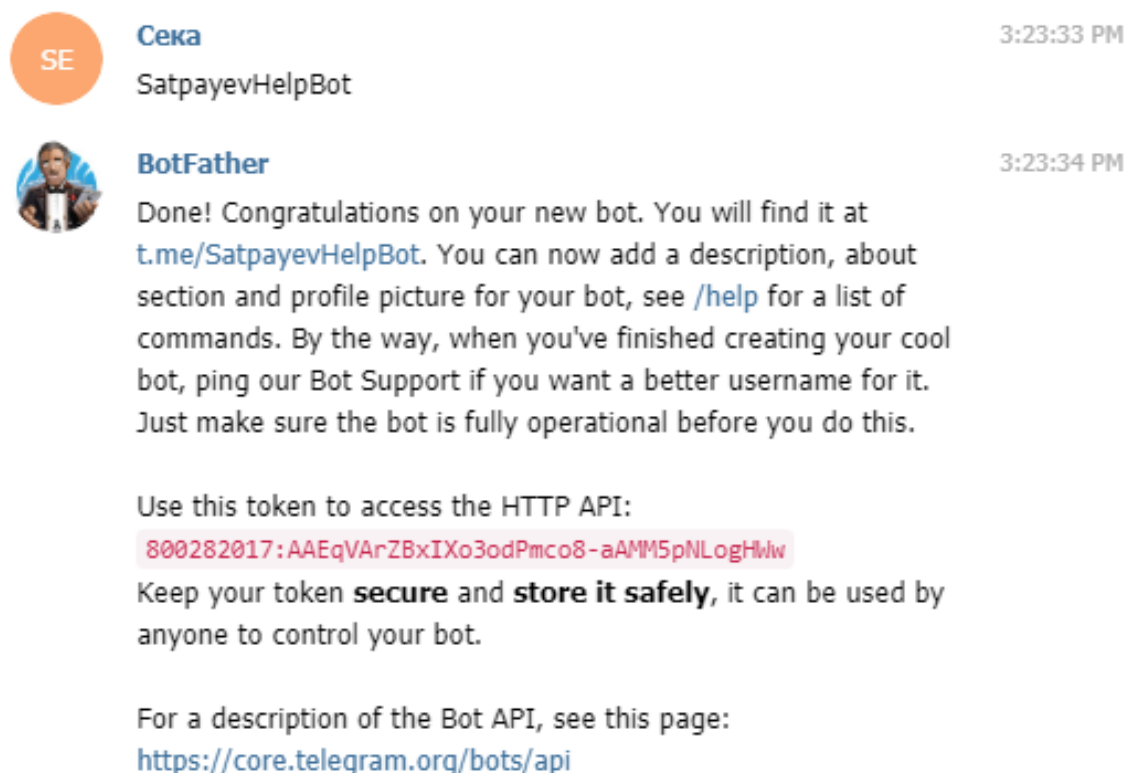
### 3.2-сурет – BotFather ботының қызметтері

Бізге BotFathet боты токен беру үшін біз құргалы жатқан боттың атауын беруіміз қажет. Ол 3.3-суретте көрсетілген.



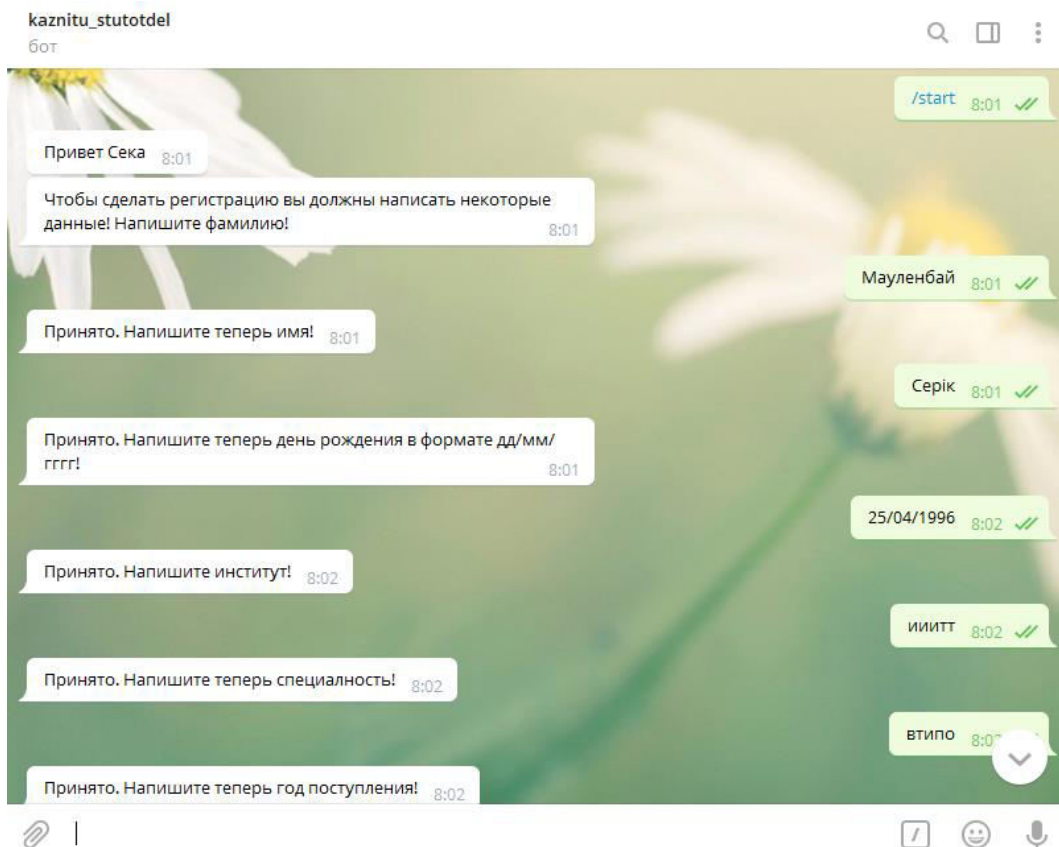
### 3.3-сурет – Құрғалы жатқан боттың атауын беру

Белгілі әрекеттерді орындағаннан кейін телеграм әлеуметтік желісіне токен алдық. Ол 3.4-суретте көрсетілген.

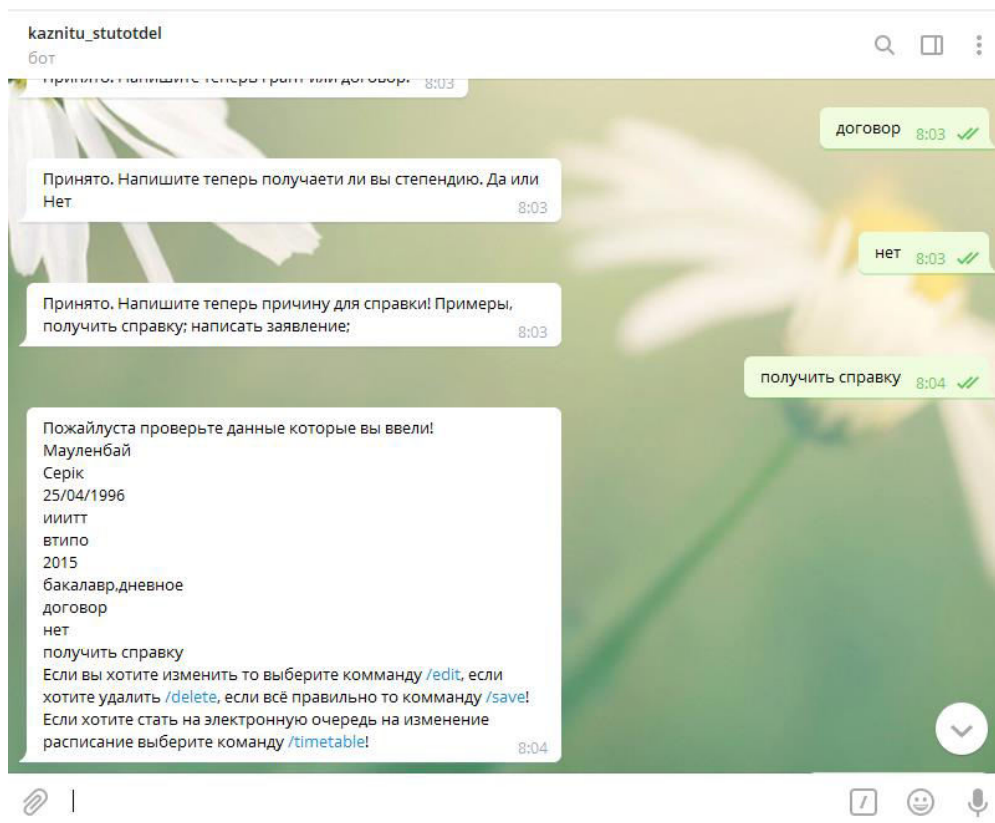


### 3.4-сурет – Токен алу

Чат боттың бастыпқы және соңғы беті 3.5-суретте және 3.6-суретте көрсетілген.



### 3.5-сурет – Чат боттың бастапқы беті



### 3.6-сурет – Чат боттың соңғы беті

## ҚОРЫТЫНДЫ

Боттар – түрлі функцияларды орындайтын және олардың пайдаланушылардың өмірін жеңілдететін арнайы бағдарламалар. Жеделхаттардың платформалары үшін жазылған, олар түрлі функцияларды орындауға арналған: жаңалықтар алудан бастап ақпарат іздеуге және тіпті акциялар саудасына дейін. Боттың басты міндеті оған пайдаланушы енгізген командадан кейін автоматты жауап болып табылады. Бұл ретте, тікелей интерфейс арқылы жеделхат арқылы жұмыс істей отырып, бағдарлама тірі юзердің іс-әрекетін имитациялайды, соның есебінен мұндай ботаны пайдалану әлдеқайда ыңғайлы және түсінікті.

Сондықтан, Интернет арқылы бизнесті дамытатын көптеген компаниялар боттардың мүмкіндіктерін бірнеше себептер бойынша пайдаланады:

- олар мақсатты аудиториямен коммуникацияның кезекті арнасын іске қосуға мүмкіндік береді;

- олар бір типтегі сұрақтарға жауап беретін адаммен өзара әрекеттеседі.

Telegram бот – жасанды интеллект функциялары бар бағдарламалық жасақтаманы басқаратын Telegram тіркелгісі. Олар басқа да қызметтерге үйрету, ойнау, іздеу, еске салу, басқа қызметтермен біріктіру немесе тіпті басқа қызметтерге командаларды беруі мүмкін. Чат боттың өнімділігі артуы үшін интернет желісі жоғары, яғни, Wi-Fi желісі немесе кабель арқылы қосылған болуы шарт емес. Чат бот жоғары жылдамдықты интернетті талап етпейді.

Бұл жүйе студенттерге қолайлы болу үшін құрастырылған. Студенттердің уақытын үнемдейтін және анықтама алудың заманауи түрі десекте болады. Студенттердің сабақ уақытында сабақ кестесін ауыстырудың кезегіне алаңдамауына мүмкіндік береді. Жәнеде анықтама алу үшін даныйларын енгізеді, содай кейін анықтама дайын болғанда барып алып кетуіне болады.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Телеграм // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/Telegram>
2. Python // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/Python>
3. Visual code // Сайттың электронды нұсқасы [https://ru.wikipedia.org/wiki/Visual code](https://ru.wikipedia.org/wiki/Visual_code)
4. UML// Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/UML>
5. База данных // Сайттың электронды нұсқасы [https://ru.wikipedia.org/wiki/База\\_данных](https://ru.wikipedia.org/wiki/База_данных)
6. Telegram bot API // Сайттың электронды нұсқасы <https://telegramzy.ru/api-bot>
7. Что такое чат-боты и зачем они нужны // Сайттың электрондық нұсқасы <https://informburo.kz/cards/что-такое-chat-boty-i-zachem-oni-nuzhny.html>
8. MySQL // Сайттың электрондық нұсқасы <https://ru.wikipedia.org/wiki/MySQL>
9. Инструкция: Создание бота для Telegram без навыков программирования // Сайттың электрондық <https://vc.ru/selectel/22593-howto-bot-selectel>
10. Самоучитель Python // Сайттың электрондық нұсқасы <https://pythonworld.ru/samouchitel-python>

## **А Қосымшасы** (міндетті)

### Техникалық тапсырма

#### **А.1 Жұмыстың мақсаты мен міндеттері**

##### **А.1.1 Жүйенің мақсаты және міндеті**

Жүйенің мақсатын келесідей топтауға болады:

– бұл чат бот технологиясын пайдаланып университетіміздегі сабақ кестесін ауыстыру және анықтама алу үшін электронды кезекке тұруды қамтамасыз ету;

– жоғарғы оқу орнындағы студенттердің уақытын үнемдеу;

– жоғарғы оқу орнындағы тіркеуші кеңсенің жұмысын жеңілдету;

Чат бот келесі функцияларды қамтамасыз етуі керек:

– тапсырыс берушінің тұтынушыларынан өтініштерді қабылдау процесін автоматтандыру;

– чат пен сұхбат арасындағы ақпараттық өзара іс-қимыл тетігін құру;

– пайдаланушыдан өтінішті қабылдау;

– кез келген дербес компьютер немесе мобильді құрылғы (смартфон) арқылы жұмыс жасау.

##### **А.1.2 Қолдану саласы**

Жоба Сәтбаев Университетінің студенттеріне арналған. Және оның келесідей мүмкіндіктері бар:

– студенттермен байланысқа түсу;

– студенттердің данныйларын қабылдау;

– электронды кезек құру;

– электрондық кезектің жақындауы туралы ақпарат беріп отыру.

Сонымен қатар бұл жобаны кез келген жоғарғы оқу орындары, кәсіпорындар, ІТ бөлімшелер қолдана алады.

#### **А.2 Жүйенің сипаттамасы**

Бұл жүйе студенттерге қолайлы болу үшін құрастырылған. Студенттердің



уақытын үнемдейтін және анықтама алудың заманауи түрі десекте болады. Студенттердің сабақ уақытында сабақ кестесін ауыстырудың кезегіне алаңдамауына мүмкіндік береді. Жәнеде анықтама алу үшін даннйларын енгізеді, содай кейін анықтама дайын болғанда барып алып кетуіне болады.

### **А.3 Өндеуге талаптар**

#### **А.3.1 Функциональдық мінездемеге талаптар**

Чат бот дұрыс жұмыс жасауы үшін мына талаптар жүзеге асырылуы керек: авторизацияның, үлкен трафик кезінде қосымшаның дұрыс жұмыс істеуі, өтінім құру жұмыстарының дұрыс орындалуы, жүйені бақылап отыру, қолданушы интерфейсінің ыңғайлы және түсінікті болуы.

#### **А.3.2 Өнімділігі**

Чат боттың өнімділігі артуы үшін интернет желісі жоғары, яғни, Wi-Fi желісі немесе кабель арқылы қосылған болуы шарт емес. Чат бот жоғары жылдамдықты интернетті талап етпейді.

#### **А.3.2.3 Қатені өңдеу**

Жүйедегі барлық қателер өңделуі тиіс және қате туралы хабарлар қолданушыларға көрсетілуі керек.

## Б Қосымшасы (міндетті)

### Бағдарлама мәтіні

```
import config
import telebot
import telegram
import sqlite3
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters
bot = telebot.TeleBot(config.token)
updater = Updater('762592341:AAHafkeSv6y5GUYhuDbZraQd8LYuSV6tIU')
dispatcher = updater.dispatcher
connection = sqlite3.connect('order.db',check_same_thread=False)
data1=[]
function = 0
editing = 0
error = 0
user_id1=0
def start(bot, update):
    global user_id1
    global data1
    global function
    function = 0
    data1 = []
    update.message.reply_text(
        'Привет {}'.format(update.message.from_user.first_name))
    bot.sendMessage(chat_id=update.message.chat_id,
        text="Чтобы сделать регистрацию вы должны написать некоторые
данные! Напишите фамилию!")
    user_id1=update.message.from_user.id
def register(bot,update):
    user_id=update.message.from_user.id
    global data1
    global function
    global editing
    global user_id1
    # Get the text the user sent
    text = update.message.text
    # Run it through the summarizer
    if len(text)>2 and function != 3:
        if text[0]=='a' and text[1]=='d' and text[2]=='m':
            snd=int(text[3:])-1
```

## Б Қосымшасының жалғасы

```
cursor = connection.cursor()
cursor.execute('SELECT * FROM KAZNITUstut_otdel')
ac=0
for row in cursor:
    if ac == snd:
        bot.send_message(chat_id=int(row[0]),text="Ваша очередь пришла.
Просим зайти!")
    else:
        ac=ac+1
if len(data1)==0 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь имя!")
elif len(data1)==1 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь день рождения в формате
дд/мм/гггг!")
elif len(data1)==2 and function == 0 and user_id1==user_id:
    print(data1)
elif len(data1)==3 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь специальность!")
elif len(data1)==4 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь год поступления!")
elif len(data1)==5 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь программу обучения(бакалавр,
магистр, дневное, вечерное)!")
elif len(data1)==6 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь грант или договор!")
elif len(data1)==7 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Принято. Напишите теперь получаете ли вы степендию. Да или Нет")
elif len(data1)==8 and function == 0 and user_id1==user_id:
```

## Б Қосымшасының жалғасы

```
data1.append(text)
bot.sendMessage(chat_id=update.message.chat_id,
                text="Принято. Напишите теперь причину для справки! Примеры,
получить справку; написать заявление;")
elif len(data1)==9 and function == 0 and user_id1==user_id:
    data1.append(text)
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Пожайлуста проверьте данные которые вы
ввели!\n"+data1[0]+\n"+data1[1]+\n"+data1[2]+\n"+data1[3]+\n"+data1[4]+\n"
+data1[5]+\n"+data1[6]+\n"+data1[7]+\n"+data1[8]+\n"+data1[9]+\n"+"Если
вы хотите изменить то выберите команду /edit, если хотите удалить /delete,
если всё правильно то команду /save! Если хотите стать на электронную
очередь на изменение расписание выберите команду /timetable!")
elif len(data1)==10 and function == 1 and user_id1==user_id:
    if text.upper() == "ДА":
        data1 = []
        bot.sendMessage(chat_id=update.message.chat_id,
                        text="Принято. Ваши данные удалены")
    elif text.upper == "НЕТ":
        bot.sendMessage(chat_id=update.message.chat_id,
                        text="Принято.")
    function = 0
else:
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Извините, произошла ошибка. Можете написать
сначала!")
elif len(data1) == 10 and function ==2 and user_id1==user_id:
    if editing == 0:
        editing = int(text)
        if editing > 0 and editing <11:
            bot.sendMessage(chat_id=update.message.chat_id,
                            text="Принято. Напишите новое значние этого поля!")
        else:
            editing = 0
            bot.sendMessage(chat_id=update.message.chat_id,
                            text="Извините, произошла ошибка. Можете написать
сначала!")
    elif editing > 0 and editing < 11 and editing != 3:
        data1[editing-1]=text
        bot.sendMessage(chat_id=update.message.chat_id,
                        text="Пожайлуста проверьте данные которые вы
ввели!\n"+data1[0]+\n"+data1[1]+\n"+data1[2]+\n"+data1[3]+\n"+data1[4]+\n"
```

## Б Қосымшасының жалғасы

```
+data1[5]+\n"+data1[6]+\n"+data1[7]+\n"+data1[8]+\n"+data1[9]+\n"+"Если
вы хотите изменить то выберите команду /edit, если хотите удалить /delete,
если всё правильно то команду /save! Если хотите стать на электронную
очередь на изменение расписание выберите команду /timetable!")
    function = 0
elif function == 3 :
    if text=="1234567":
        #bot.sendDocument(chat_id=chat_id, document=open('1.pdf', 'rb'))
        #bot.sendDocument(chat_id=chat_id, document=open('kaznitu_stutotdel.txt',
'rb'))
        #bot.sendDocument(chat_id=chat_id,
document=open('kaznitu_stutotdel_timetable.txt', 'rb'))
        sql = 'delete from KAZNITUstut_otdel'
        sql1 = 'delete from KAZNITUstut_otdel_timetable'
        cur = connection.cursor()
        cur1 = connection.cursor()
        cur.execute(sql)
        cur1.execute(sql1)
        open('kaznitu_stutotdel.txt', 'w').close()
        open('kaznitu_stutotdel_timetable.txt', 'w').close()
        bot.sendMessage(chat_id=update.message.chat_id,
            text="Данные очищены!")
        function=0
    else:
        bot.sendMessage(chat_id=update.message.chat_id,
            text="Извините, произошла ошибка. Можете написать сначала!")
def database(user_id,bot,update):
    global error
    cursor = connection.cursor()
    cursor.execute('SELECT * FROM KAZNITUstut_otdel')
    for row in cursor:
        if int(row[0])==user_id:
            error=error+1
    if error==0:
        connection.execute("INSERT INTO KAZNITUstut_otdel VALUES
(?)",(str(user_id),))
        connection.commit()
        cursor = connection.cursor()
        cursor.execute('SELECT * FROM KAZNITUstut_otdel')
        numberOfStudents= 0
        for row in cursor:
            numberOfStudents=numberOfStudents+1
```

## Б Қосымшасының жалғасы

```
bot.send_message(chat_id=update.message.chat_id,text="Ваша позиция в
очереди "+str(numberOfStudents)+" . Приём студентов начинается с 12:00. Время
ожидание 3 минуты. После этого заходит следующий. Спасибо за воспользование
наших новых технологий!")
else:
    bot.send_message(chat_id=update.message.chat_id,text="Вы уже
зарегистрировались, вы можете зарегистрироваться только завтра!")
#отправляем дальнейшую инструкцию
def database1(user_id,bot,update):
    global error
    cursor = connection.cursor()
    cursor.execute('SELECT * FROM KAZNITUstut_otdel_timetable')
    for row in cursor:
        if int(row[0])==user_id:
            error=error+1
    if error==0:
        connection.execute("INSERT INTO KAZNITUstut_otdel_timetable VALUES
(?)",(str(user_id),))
        connection.commit()
        cursor = connection.cursor()
        cursor.execute('SELECT * FROM KAZNITUstut_otdel_timetable')
        numberOfStudents= 0
        for row in cursor:
            numberOfStudents=numberOfStudents+1
        bot.send_message(chat_id=update.message.chat_id,text="Ваша позиция в
очереди "+str(numberOfStudents)+" . Приём студентов начинается с 12:00. Время
ожидание 3 минуты. После этого заходит следующий. Спасибо за воспользование
наших новых технологий!")
    else:
        bot.send_message(chat_id=update.message.chat_id,text="Вы уже
зарегистрировались, вы можете зарегистрироваться только завтра!")
#отправляем дальнейшую инструкцию
def dateConverter(bot,update):
    global function
    global data1
    global user_id1
    day,month,year=update.message.text.split('/')
    user_id=update.message.from_user.id
    if len(data1) == 2 and function == 0 and user_id1==user_id:
        textDate = str(day)+"/"+month+"/"+year
        data1.append(textDate)
        bot.sendMessage(chat_id=update.message.chat_id,
```

## Б Қосымшасының жалғасы

```
        text="Принято. Напишите институт!")
elif len(data1) == 10 and function == 2 and editing == 3 and user_id1==user_id:
    textDate = str(day)+"/"+month+"/"+year
    data1[2]=textDate
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Пожайлуста проверьте данные которые вы
ввели!\n"+data1[0]+\n"+data1[1]+\n"+data1[2]+\n"+data1[3]+\n"+data1[4]+\n"
+data1[5]+\n"+data1[6]+\n"+data1[7]+\n"+data1[8]+\n"+data1[9]+\n"+"Если
вы хотите изменить то выберите команду /edit, если хотите удалить /delete,
если всё правильно то команду /save! Если хотите стать на электронную
очередь на изменение расписание выберите команду /timetable!")
    function = 0
else:
    bot.sendMessage(chat_id=update.message.chat_id,
                    text="Извините, произошла ошибка. Можете написать сначала!")
def delete(bot,update):
    global data1
    global user_id1
    global function
    user_id=update.message.from_user.id
    if len(data1) != 10 and user_id1==user_id:
        bot.sendMessage(chat_id=update.message.chat_id,
                        text="Сначала заполните данные нажав команду /start!")
    else:
        function = 1
        bot.sendMessage(chat_id=update.message.chat_id,
                        text="Вы уверены что хотите удалить? Напишите ДА или НЕТ!")
def edit(bot,update):
    global data1
    global editing
    global function
    if len(data1) == 10:
        function = 2
        editing = 0
        bot.sendMessage(chat_id=update.message.chat_id,
                        text="Напишите номер поля который вы хотите изменить! \n 1.
Фамилия \n 2. Имя \n 3. День рождения \n 4. Институт \n 5. Специальность \n 6.
Год поступления \n7. Программа обучения(бакалавр, магистр, дневное,
вечерное) \n 8. Грант/договор \n 9. Получает степендию/ не получает \n 10.
Причина для справки")
    else:
        bot.sendMessage(chat_id=update.message.chat_id,
```



## Б Қосымшасының жалғасы

```
text="Сначала заполните данные нажав команду /start!")
def saving(bot,update):
    global data1
    global error
    error = 0
    user_id=update.message.from_user.id
    database(user_id,bot,update)
    if len(data1)==10 and error==0:
        main_row=data1[0]+", "+data1[1]+", "+data1[2]+", "+data1[3]+", "+data1[4]+",
"+data1[5]+", "+data1[6]+", "+data1[7]+", "+data1[8]+", "+data1[9]+".
        f = open('kaznitu_stutotdel.txt','a')
        f.write("\n" +main_row)
        f.close()
        start(bot,update)
    else:
        bot.sendMessage(chat_id=update.message.chat_id,
            text="Извините, произошла ошибка. Можете повторить
сначачла!")
def admin(bot,update):
    global function
    function = 3
    bot.sendMessage(chat_id=update.message.chat_id,
        text="Введите пароль!")
def timetable(bot,update):
    global data1
    global error
    error = 0
    user_id=update.message.chat_id
    database1(user_id,bot,update)
    if len(data1)==10 and error==0:
        main_row=data1[0]+", "+data1[1]+", "+data1[2]+", "+data1[3]+", "+data1[4]+",
"+data1[5]+", "+data1[6]+", "+data1[7]+", "+data1[8]+", "+data1[9]+".
        f = open('kaznitu_stutotdel_stutotdel.txt','a')
        f.write("\n" +main_row)
        f.close()
        start(bot,update)
if __name__=='__main__':
    connection.execute("CREATE TABLE IF NOT EXISTS
KAZNITUstut_otdel(user_id text)")
    connection.execute("CREATE TABLE IF NOT EXISTS
KAZNITUstut_otdel_timetable(user_id text)")
    start_handler = CommandHandler('start', start)
```

## Б Қосымшасының жалғасы

```
delete_handler = CommandHandler('delete', delete)
edit_handler = CommandHandler('edit', edit)
save_handler = CommandHandler('save', saving)
admin_handler = CommandHandler('admin', admin)
timetable_handler = CommandHandler('timetable', timetable)
updater.dispatcher.add_handler(MessageHandler(Filters.text, register),group=0)
updater.dispatcher.add_handler(MessageHandler(Filters.text,
dateConverter),group=1)
dispatcher.add_handler(start_handler)
dispatcher.add_handler(delete_handler)
dispatcher.add_handler(edit_handler)
dispatcher.add_handler(save_handler)
dispatcher.add_handler(admin_handler)
dispatcher.add_handler(timetable_handler)
updater.start_polling()
updater.idle()
```



