

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

Анужингийн Сайнзолбоо

АЖИЛ ОЛГОГЧДЫН ӨГӨГДЛИЙН АНАЛИЗ
СИСТЕМ ДЭЭР СУУРИЛСАН ЧАТ БОТ
(Chat bot based on sytem analysis of employers' data)

Мэдээллийн технологи (D061303)
Бакалаврын судалгааны ажил

Улаанбаатар

2022 оны 03 сар

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

АЖИЛ ОЛГОГЧДЫН ӨГӨГДЛИЙН АНАЛИЗ СИСТЕМ
ДЭЭР СУУРИЛСАН ЧАТ БОТ

(Chat bot based on sytem analysis of employers' data)

Мэдээллийн технологи (D061303)
Бакалаврын судалгааны ажил

Удирдагч: _____ Б.Хуягбаатар доктор (Ph.D.)
Гүйцэтгэсэн: _____ А.Сайнзолбоо (18B1NUM1762)

Улаанбаатар

2022 оны 03 сар

Зохиогчийн баталгаа

Миний бие Анужингийн Сайнзолбоо “АЖИЛ ОЛГОГЧДЫН ӨГӨГДЛИЙН АНАЛИЗ СИСТЕМ ДЭЭР СУУРИЛСАН ЧАТ БОТ” сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ эсвэл ихэнхдээ Монгол Улсын Их Сургуулийн зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй.
- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг дипломын ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

УДИРТГАЛ	1
БҮЛГҮҮД	2
1. СЭДВИЙН ТАНИЛЦУУЛГА	2
1.1 Оршил	2
1.2 Зорилго	2
1.3 Зорилт	2
1.4 Алсын хараа	3
2. СИСТЕМИЙН СУДАЛГАА	4
2.1 Системийн судалгаа	4
2.2 Ижил төстэй системүүд	7
2.3 Технологийн судалгаа	9
3. СИСТЕМИЙН ШИНЖИЛГЭЭ	16
3.1 Бизнесийн үйл ажиллагааны шинжилгээ	16
3.2 Хэрэглэгч	17
3.3 Функционал шаардлага	17
3.4 Функционал бус шаардлага	18
3.5 Use case диаграм	19
4. СИСТЕМИЙН ЗОХИОМЖ	20
4.1 Өгөгдлийн сангийн диаграм	20
4.2 Өгөгдлийн элемент	21
4.3 Өгөгдлийн сангийн холбоосын тайлбар	24
5. ХЭРЭГЖҮҮЛЭЛТ, ҮР ДҮН	25
5.1 Хөгжүүлсэн байдал	25

НОМ ЗҮЙ	33
ХАВСРАЛТ.....	34
А. ҮЕЧИЛСЭН ТӨЛӨВЛӨГӨӨ	34
В. КОДЫН ХЭРЭГЖҮҮЛЭЛТ.....	35
В.1 Өгөгдөл цуглуулалт	35

ЗУРГИЙН ЖАГСААЛТ

2.1	Pizza Hut chat bot	7
2.2	WHO's chat bot	8
2.3	Python лого	9
2.4	BeautifulSoup лого	10
2.5	Өгөгдөл цуглуулалтын жишээний үр дүн	11
2.6	Cosine similarity утгын график	12
2.7	SBERT.net лого	12
2.8	cosine-similarity ашигласан жишээ	13
2.9	Чатботын амьралын мөчлөг	14
3.1	BPMN-1	16
3.2	BPMN-2	17
3.3	Use Case диаграм	19
4.1	Өгөгдлийн сангийн диаграм	20
5.1	Үндсэн процесс зураглал	26
5.2	Data set	32
A.1	Бакалаврын судалгааны ажлын үечилсэн төлөвлөгөө	34
B.1	Фолдерийн бүтэц	35

ХҮСНЭГТИЙН ЖАГСААЛТ

4.1	advertisement хүснэгт	21
4.2	category хүснэгт	23
4.3	location хүснэгт	23
4.4	contactInfo хүснэгт	24

Кодын жагсаалт

2.1	Python энгийн жишээ	9
2.2	BeautifulSoup жишээ өгөгдөл цуглуулалт	10
5.1	Data Link crawling	27
5.2	Өгөгдөл цуглуулах	28
5.3	Хуудаслалтыг задлах	29
5.4	CSV файлууд хадгалах	30
B.1	Бүх өгөгдлийг цуглуулах - dataScrapping.py	35
B.2	Нэг зарын өгөгдлийг цуглуулах - adScrape.py	37
B.3	Өгөгдлийн төрөл - classTypes.py	40
B.4	Scrape хийх функц - scrape.py	40

УДИРТГАЛ

Мэдээллийн технологи эрчимтэй хөгжиж буй өнөөгийн нийгэмд байгууллага үйл ажиллагаа явуулж эхэлсэн цагаасаа эхлэн өгөгдлийг үйлдвэрлэсээр байдаг. Тэдгээр өгөгдлийг байнга хадгалах нь өгөгдлийн сангийн нөөцөд хортой байдаг тул өгөгдөлд шинжилгээ хийж, тэдгээрээс шаардлагатай өгөгдлүүдийг түүвэрлэн хадгалах нь чухал юм.

Өнөөдөр бид дэлхий нийтээрээ хурдтай амьдралын хэмнэлд ажиллаж, амьдарч байна. Мөн зах зээлийн хөгжил, ажил олгогчийн эрэлт хэрэгцээ ажил хайгчийн хүсэл онирхлыг соновчтой бөгөөд хурдан холбож өгөх нь нэн шаардлагатай. Өнөөгийн байдлаар энэ эрэлт хэрэгцээг хангасан тодорхой шийдвэрлэсэн мэдээллийн систем хомс байна. Иймд энэхүү бакалаврын судалгааны ажлаар ажил олгогч болон ажил идэвхтэй хайгч хоёрыг түргэн шуурхай холбож өгөх чатбот системийг хөгжүүлж байна.

1. СЭДВИЙН ТАНИЛЦУУЛГА

1.1 Оршил

Энэхүү бакалаврын судалгааны ажлын хүрээнд “Ажил олгогчдын өгөгдлийн анализ систем дээр суурилсан чатбот” сэдвийн дагуу ажил хайгчдыг ажлын байрны мэдээллээр хангах Чатбот системийг хөгжүүлнэ. Ажлын байрны мэдээллийг Data Scraping аргын тусламжтайгаар, системд шаардлагатай мэдээллийг өгөгдлийн сангийн хэлбэрт оруулан бүтэцтэйгээр нэгтгэн авах бөгөөд үүнээс ажил хайгчдын дунд байдаг түгээмэл асуултуудын хариултыг өгнө. Мөн энэ системд машин сургалтын арга болох Language Understanding-ийг ашиглан хэрэглэгчийн асуултыг таамаглаж оновчтой хариулт өгөх боломжийг олгох юм.

1.2 Зорилго

Ажлын хайгчдын хэрэгцээт асуултад хариулж, ажлын байрны хүртээмжийг нэмэгдүүлэхэд энэхүү системийн гол зорилго оршино.

1.3 Зорилт

Дээрх зорилгод хүрэхийн тулд дараах зорилтуудыг тавьсан. Үүнд:

- Ашиглагдах технологиудыг сонгох, судлах
- Ижил төстэй системийн судалгаа хийх
- Системийн шинжилгээ хийх
- Системийг зохиомжлох
- Системийг хөгжүүлэх, сайжруулалт хийх

1.4 Алсын хараа

Ажлын байрны дэлгэрэнгүй мэдээллийг цуглуулснаар цаашид тэдгээрт шинжилгээ хийж хамгийн их эрэлттэй, өндөр цалинтай ажлын байр гэх зэрэг мэдээллүүдийг систем хэрэглэгчдэд хүргэх боломжтой юм.

2. СИСТЕМИЙН СУДАЛГАА

2.1 Системийн судалгаа

Сонгосон сэдэв болох “Ажил олгогчдын өгөгдлийн анализ систем дээр суурилсан чатбот” сэдвийн хүрээнд судалгаа хийхдээ чатбот системийн талаар болон өгөгдөл цуглуулгын аргын талаар судалсан. Үүний дараа ижил төстэй системийн болон ашиглагдах технологийн талаар судалгааг хийсэн болно.

2.1.1 Чатбот систем

Чатбот систем нь ихэвчлэн хэрэглэгчийн асуултыг хиймэл оюун ухааны тусламжтайгаар ойлгож, хариултыг автоматжуулах үндсэн зорилготой компьютерийн програм хангамж юм. Орчин үед хэрэглэгчдэд туслах үндсэн үүргийн дагуу чатбот системийг байгууллагууд олон янзаар ашиглах болсон. Тэдгээрээс дурдвал,

- Цэс дээр суурилсан чатбот (Menu-based chatbot)
- Түлхүүр үгийг танихад суурилсан чатбот (Keyword recognition-based chatbot)
- Машин сургалтын чатбот (Machine learning chatbot)

Цэс дээр суурилсан чатбот

Өнөөгийн зах зээлд хэрэгжиж буй чатботуудын хамгийн энгийн бөгөөд түгээмэл хэлбэр юм.[1]

¹ Хэрэглэгчийн асууж болох асуултуудыг урьдаас таамаглан хариултуудыг мод хэлбэртэйгээр бүтэцлэн хадгалдаг. Хэрэглэгч хүссэн хариултаа авахын тулд системийн хадгалсан хариултаар аялах хэрэгтэй болдог. Бусад чатботтой харьцуулбал, хариулт хязгаарлагдмал бөгөөд хэрэглэгчээс олон асуулт асууж цаг их шаарддагаараа сул талтай байдаг.

¹<https://www.engati.com/blog/types-of-chatbots-and-their-applications>

Түлхүүр үгийг танихад суурилсан чатбот

Энэхүү чатбот нь хэрэглэгчийн бичсэнийг уншиж тохиромжтой хариултыг өгдөг. Ингэхдээ өгүүлбэрийг хиймэл оюун ухааны нэг хэсэг болох эх хэлний боловсруулалт (Natural Language Processing)-ын тусламжтайгаар шинжилж түлхүүр үгийг таньж хариултыг өгдөг. Ижил төстэй олон асуултад хариулах эсвэл түлхүүр үг дутуу үед амжилтгүй болдог. Мөн хэрэглэгч хүссэн хариултаа олж чадахгүй байх болон үр дүн муутай хариулт өгсөн тохиолдолд цэс дээр суурилсан чатботыг хослуулан ашиглах нь найдвартай болдог бөгөөд түгээмэл шийдлүүдийн нэг байдаг.

Машин сургалтын чатбот

Энэ төрлийн чатбот нь өмнө хэрэглэгчийн харилцан яриан дээр хиймэл оюун ухаан болон машин сургалтын тусламжтайгаар шинжилгээ хийж, хэрэглэгчийн зан төлөв, асуултын хэв маягаас суралцдаг. Ингэснээрээ чатботод хэрэглэгчийн зарцуулах цаг эрчимтэйгээр буурах буюу хариултаа авах алхам багасгах ба хэрэглэгчийн туршлага (UX) нь түүнийгээ даган өсөх нь энэхүү чатботын үндсэн зорилго болно.

Чатботыг сонгох

Машин сургалтын чатбот нь илүү уян хатан хэрэглэгчдэд ээлтэй чатботыг бий болгодог боловч хөгжүүлэхэд цаг хугацаа их шаардагдах ба машин өөрөө суралцахад мөн хугацаа шаардагддаг. Иймд системийн нөөц, шаардлагыг харгалзан үзэж энэхүү судалгааны ажлаар түлхүүр үг танихад суурилсан чатботыг хэрэгжүүлэхийг зориод байна.

2.1.2 Өгөгдөл цуглуулгын арга

Өгөгдөл цуглуулах (data scraping) нь хэрэглэгчдэд харагдаж буй өгөгдлийг олон янзын сувгаас цуглуулан хувийн орчинд хадгалан цаашид ашиглах боломжийг олгодог хамгийн үр дүнтэй автомат өгөгдөл олборлох арга юм. Ихэвчлэн өгөгдөл цуглуулах арга нь вэбсайтаас шаардлагатай өгөгдлийг цуглуулахад ашигладаг. Өгөгдөл цуглуулж буй хүнээс хамааран олборлосон өгөгдлийг таслалаар тусгаарлагдсан утгын (Comma-Separated Values) файл эсвэл

өгөгдлийн санд хадгалах боломжтой бөгөөд нэгэнт цуглуулсан их хэмжээний өгөгдөлд судалгаа шинжилгээ хийх, худалдаа, борлуулалтын хэрэгсэл болгох зэрэг олон төрлийн боломжийг олгодог.

Вебсайтаас өгөгдлийг олборлох хамгийн түгээмэл арга нь HTML parsing буюу HTML-ийг задлан шинжлэх юм. Энэ нь вебсайтын HTML болох сайтын үндсэн бүтцийг агуулгынх нь хамтаар хуулах бөгөөд авах гэж буй өгөгдлийн зан төрхийг нь зааж өгснөөр доторх агуулгыг хамгийн хялбар бөгөөд автомат байдлаар цуглуулдаг юм. Цуглуулга хийх 2 үндсэн арга байдаг. Үүнд:

- Өгөгдлийг цуглуулж, задлах (Data scraping)
- Өгөгдлийг олж илрүүлж, хаягийг цуглуулах (Data crawling)

Өгөгдлийг цуглуулж, задлах

Нэг үгээр хэлбэл өгөгдлийг цуглуулж, задлах нь зааж өгсөн хаягийн дагуу шаардлагатай өгөгдлийг задалж, хэрэгтэй агуулгыг хөгжүүлэгчдэд өгдөг бөгөөд хүссэн өгөгдлөө задлан авах боломжийг олгодгоороо давуу талтай. Өөрөөр хэлбэл өгөгдөл олборлох програм нь зорилго буюу даалгавараа мэдэж байгаа юм.

Өгөгдлийг олж илрүүлж, хаягийг цуглуулах

Энэхүү аргачлал нь хаяг тодорхой бус үед түүнийг олж илрүүлж шаардлагын дагуу хаягийг, зарим тохиолдолд өгөгдлийг цуглуулдаг. Системийн шаардлагын дагуу өгөгдлийг цуглуулах үед хаяг алгасах, дутуу өгөгдөл цуглуулахаас сэргийлдэг давуу талтай.

Ихэвчлэн энэхүү хоёр аргыг хослуулан ашигладаг бөгөөд шаардлагад нийцэх өгөгдлийг үлдээлгүй бүгдийг нь олоход *data crawling*-ийг ашиглах бол олсон өгөгдлийг задалж, шинжлэн өгөгдлийн санд хадгалах үйлдлийг *data scraping* хийдэг. Жишээлбэл, худалдааны сайтын бараа бүтээгдэхүүний өгөгдлийг цуглуулах гэж байгаа гэж үзвэл, барааны ангиллын хаягуудыг өөрчлөгдөх бүрт хадгалан өгөгдлийг цуглуулна. Өөрөөр хэлбэл нэг нь өөрчлөлт гарахыг ажиглаж вебсайтаар мөлхөж байх бол нөгөө нь шаардлагын дагуу бүх хэрэгтэй өгөгдлийг хэдийн цуглуулсан

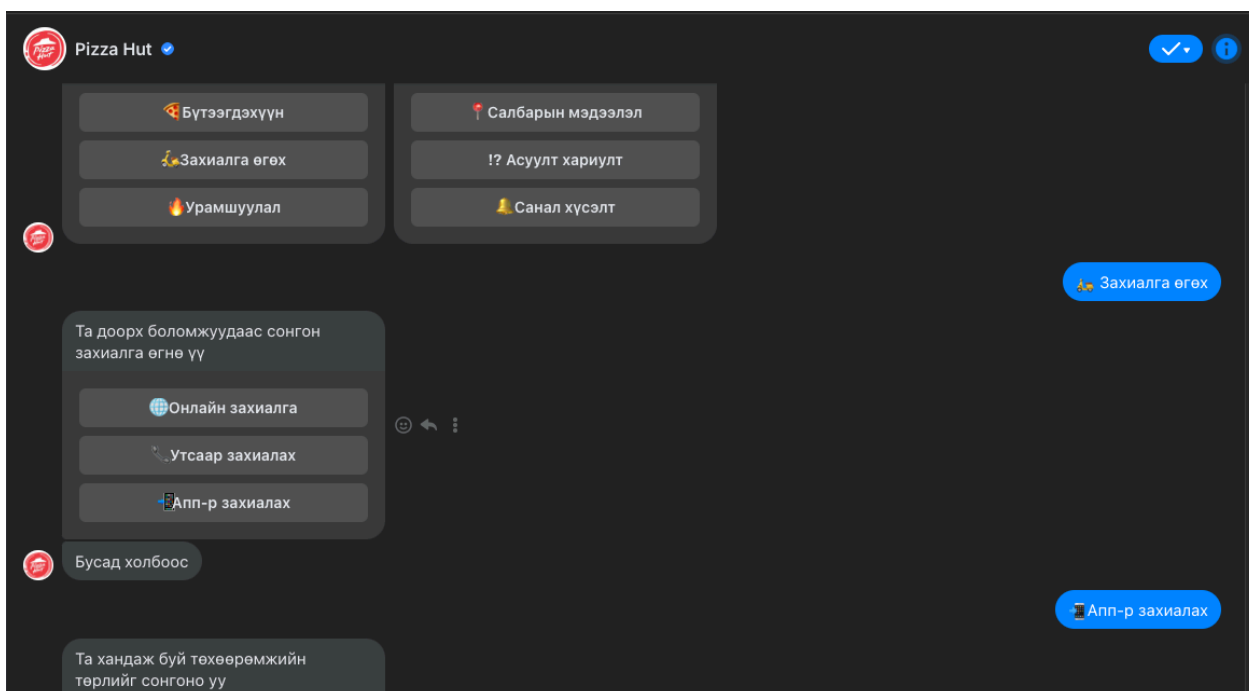
байна. Энэхүү бакалаврын судалгааны ажлын хүрээнд өгөгдлийг CSV файл үүсгэн хадгалж цаашид ашигласан болно.

2.2 Ижил төстэй системүүд

Гадаад ба дотоодын байгууллагуудын үйл ажиллагаандаа хэрэгжүүлдэг чатбот системүүдээс, *Domino's Pizza & Pizza Hut* болон *WHO's Chat bot* гэсэн гурван чатботыг сонгон авч судалгаанд оруулав.

2.2.1 *Domino's Pizza & Pizza Hut*

Domino's pizza хоолны газар нь захиалгын алхамаас эхлээд бүх мэдээллийг ганцхан *Facebook messenger chatbot* хангадаг. Чатбот эрчээ авч эхэлсэн шалтгаан нь хүмүүс, бусад хүмүүсийг хүлээлгүйгээр үйлчилгээ авах, тусламж авах зэрэг үйлчилгээг зэрэг нэвтрүүлсэнтэй холбоотой билээ. Үүний нэгэн адилаар Монголд үйл ажиллагаа явуулж буй *Pizza Hut Mongolia* юм.

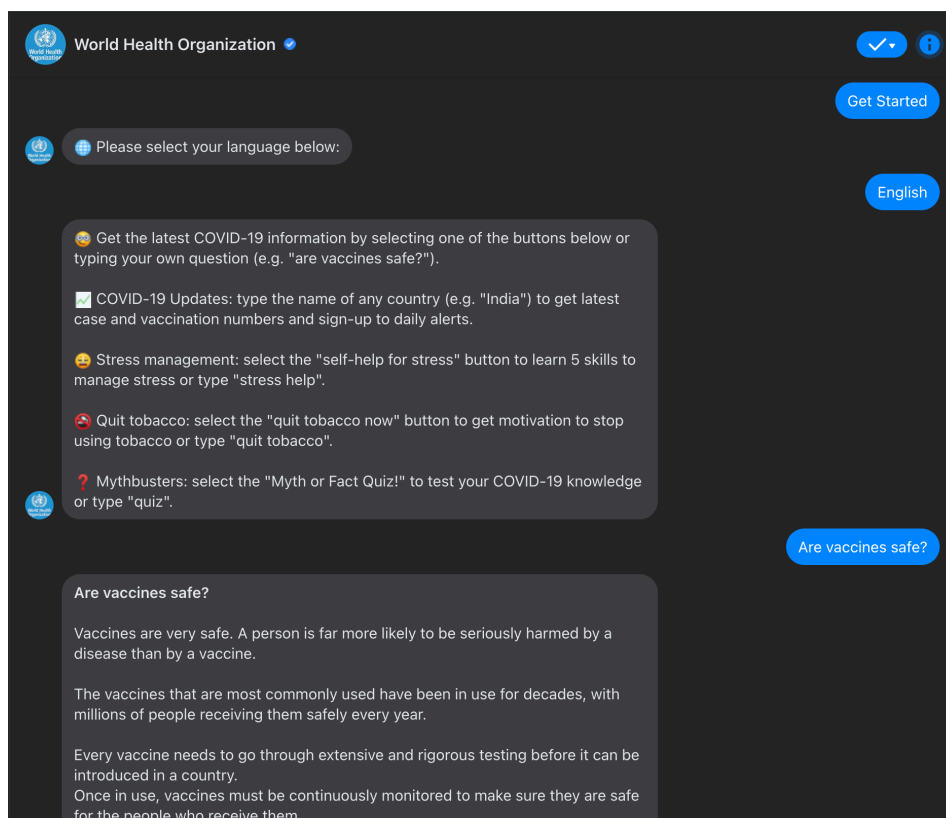


Зураг 2.1: Pizza Hut chat bot

Үйлчлүүлэгчдийн захиалга хүлээх хугацааг багасгахын тулд захиалгын үйл явцыг хурдасгаснаар тодорхой хэмжээнд нөлөөлж байгаа нь дээрх 2 жишээнээс харагдаж байна.

2.2.2 World Health Organization's Chat bot

Цар тахал болох коронавирусын эрчимтэй тархаж байх үед дэлхийн өнцөг булан бүрт оршин суугаа хүмүүст цар тахлын мэдээлэл, урьдчилан сэргийлэх арга, баталгаатай эх сурвалжийн мэдээллээр хангах зорилготой чатбот юм. Дэлхий нийтээр вакцинжуулалтын хөдөлгөөн өрнөж байх үеэр вакцины талаарх мэдээлэл, архаг хууч өвчинд нөлөөлөх талаар найдвартай, хамгийн сүүлийн үеийн албан ёсны мэдээллийг өгдөг. Хэдий халдварын тоо буурч, нийгэм өөрөө дасан зохьцож байгаа хэдий ч Дэлхийн Эрүүл Мэндийн байгууллага үүргээ гүйцэтгэж чухал эх сурвалжаар хангасаар байгаагийн шинж юм.



Зураг 2.2: WHO's chat bot

2.3 Технологийн судалгаа

АЖИЛ ОЛГОГЧДЫН ӨГӨГДЛИЙН АНАЛИЗ СИСТЕМ ДЭЭР СУУРИЛСАН ЧАТ БОТ-ЫГ хөгжүүлэхдээ өгөгдөл цуглуулгыг *python* хэлний сан болох *BeautifulSoup* HTML өгөгдөл задлах технологийг ашигласан бөгөөд чатбот системийн түлхүүр үг таних технологийг *Python* хэлний *Framework* болох *SentenceTransformers*-ийг сонгон хөгжүүлэлтийг хийсэн. Харин цуглуулсан өгөгдлийг *CSV* файлд хадгалан, *Microsoft Bot Framework*-ийг чатбот хөгжүүлэлтэд ашиглан судалгааг дараах байдлаар хийсэн болно.

2.3.1 *Python*

Python нь дээд түвшний маш олон төрлийн програмчлалыг өөртөө шингээсэн хэл юм. Хэлний сан болон *framework*-үүд нь тасралтгүй сайжирч, шинэчлэгдэж байдаг тул бүх л төрлийн програмчлалын аргуудыг гүйцэтгэж болдог. Орчин үед машин сургалт, хиймэл оюун ухаан болон эх хэлний боловсруулалтад(NLP) түгээмэл ашигладаг болсон бөгөөд веб хүртэл хийх боломжтойгоороо давуу талтай юм. Үүнээс гадна анхлан суралцаж буй хүмүүст ойлгоход хялбар *syntax*-ийн дүрэмтэй байдаг тул хэрэглэгчдийн тоо нь javascript, java хэлүүдтэй өрсөлдөхүйц байдаг.



Зураг 2.3: Python лого

```
1 x = 5
2 name = 'Sainzolboo'
3 print(x)
4 print(name)
```

Код 2.1: Python энгийн жишээ

Python програмчлалын хэл нь ойлгоход маш хялбар бөгөөд өөр дээрх функцууд нь шууд утгаараа ойлгомжтой байдаг. Syntax-ийн хувьд ; ашигладаггүй ба догол мөрөөр програмчлалын үндсэн схемийг гаргадагаараа онцлог хэл юм.

2.3.2 *BeautifulSoup*

Өгөгдөл цуглуулгын олон технологиудын нэг нь *BeautifulSoup* бөгөөд *python* програмчлалын хэлний сан юм. Энэ нь өгсөн вебсайтын хаяг (Url)-ийн дагуу бүх HTML өгөгдлийг агуулгын хамтаар нь хэрэглэгчид өгдөг. HTML хэл нь мод хэлбэртэй байдаг бөгөөд түүний хүүхэд элементүүдийн агуулгыг шаардлага болон түлхүүр үгийн дагуу цуглуулах зарчмаар ажилладаг.



Зураг 2.4: BeautifulSoup лого

Бакалаврын судалгааны ажлын сэдвийн дагуу ажлын байр олгогчдын мэдээлэл болон ажлын байрны мэдээллийг **zangia.mn**-ээс *BeautifulSoup* ашиглан цуглуулсан. Доорх кодын жишээнд бүх ажлын байрны ангилал болон шүүлтүүрийн агуулгыг цуглуулсан бөгөөд жишээнд зориулж зөвхөн эхний ангиллын мэдээллийг харуулав.

```
1 from bs4 import BeautifulSoup
2 import requests
3 from urllib.error import HTTPError
4
5 url = 'https://zangia.mn/'
```

```

6  try:
7      response = requests.get(url)
8      response.raise_for_status()
9  except HTTPError as error:
10     print(error)
11  soup = BeautifulSoup(response.text, 'html.parser')
12  navigatorList = soup.find_all('div', class_='filter')
13  print(navigatorList[0])

```

Код 2.2: BeautifulSoup жишээ өгөгдөл цуглуулалт

```

<div class="filter">
  <h3>
    Онцлох
  </h3>
  <div>
    <a href="job/list/x.1">
      Удирдах албан тушаалын ажлын байр
    </a>
  </div>
  <div>
    <a href="job/list/x.3">
      Англи хэлний 100%-н мэдлэг шаардах ажлын байр
    </a>
  </div>
  <div>
    <a href="job/list/x.2">
      Ажлын туршлага шаардахгүй ажлын байр
    </a>
  </div>

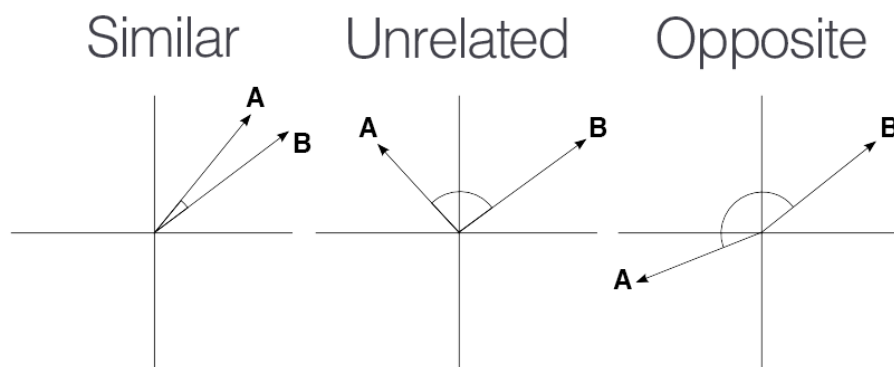
```

Зураг 2.5: Өгөгдөл цуглуулалтын жишээний үр дүн

2.3.3 SentenceTransformers

Python хэлний framework болох SentenceTransformers [2] буюу өгүүлбэр хувиргалт нь өгүүлбэр болон текстийн ижил төстэй байдал болон утгын хувьд адил байдлыг *cosine-similarity*² -ийн тусламжтайгаар тооцоолог. Энэхүү тооцооллыг цаашид өгүүлбэрийн ижил төстэй байдлыг харьцуулах, хайлт хийх, түүнд шинжилгээ хийх зэргээр ашиглаж болно. Доорх зурагт өгүүлбэрт хувиргалт хийж, шинжилгээний үр дүнгийн вектор хоорондын өнцгөөр хэрхэн тодорхойлогддог болох талаар харуулав.

²Cosine-similarity нь өгөгдлийн шинжилгээнд 2 тооны ижил төстэй байдлыг вектор үржвэрээр илэрхийлдэг.



Зураг 2.6: Cosine similarity утгын график

SentenceTransformers-ийг дэлхийн 100 гаруй хэл дээр урьдчилан бэлтгэн, сургасан эх хэлний боловсруулалт (NLP)-ын загваруудыг ашиглаж болдогоороо давуу талтай.



Зураг 2.7: SBERT.net лого

Чатбот системийн хувьд монгол хэлийг танин ашиглах боломжтой загвар болох *distiluse-base-multilingual-cased-v2[3]*-ийг ашиглан хийж гүйцэтгэв.

Хоёр өгүүлбэрийг *cosine-similarity* ашиглан ижил төстэй байдлыг илэрхийлэх жишээг доор харууллаа. Эх кодыг utf-8 формат танихгүй байсан тул зураг хэлбэрээр орууллаа.

```
tests.py > ...
1  from sentence_transformers import SentenceTransformer, util
2  model = SentenceTransformer(
3      'sentence-transformers/distiluse-base-multilingual-cased-v2')
4
5  emb1 = model.encode("Энэ бол улаан малгайтай муур юм.")
6  emb2 = model.encode("Энэ бол миний улаан малгайтай нохой.")
7
8  cos_sim = util.cos_sim(emb1, emb2)
9  print("Cosine-Similarity:", cos_sim)
10

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

/usr/local/bin/python3 /Users/zolboo/Desktop/bachelor/employmentAnalysis/project/sbert.py
zolboo@Sainzolboos-MacBook-Pro project % /usr/local/bin/python3 /Users/zolboo/Desktop/bachelor/employmentAna
Cosine-Similarity: tensor([[0.7287]])
zolboo@Sainzolboos-MacBook-Pro project %
```

Зураг 2.8: cosine-similarity ашигласан жишээ

2.3.4 Linear Regression - Шугаман Регресс

Linear Regression буюу шугамар регрессийн загвар нь шулуун шугамыг ашигладаг бол логик болон шугаман бус регрессийн загвар нь муруй шугамыг ашигладаг. Шугаман регресс нь бие даасан хувьсагч хэрхэн өөрчлөгдөхийг тооцоолох боломжийг олгодог. Хоёр тоон хувьсагчийн хоорондын хамаарлыг тооцоолоход шугаман регрессийг ашигладаг бөгөөд ихэвчлэн дараах нөхцөлд ашиглагддаг. Үүнд:

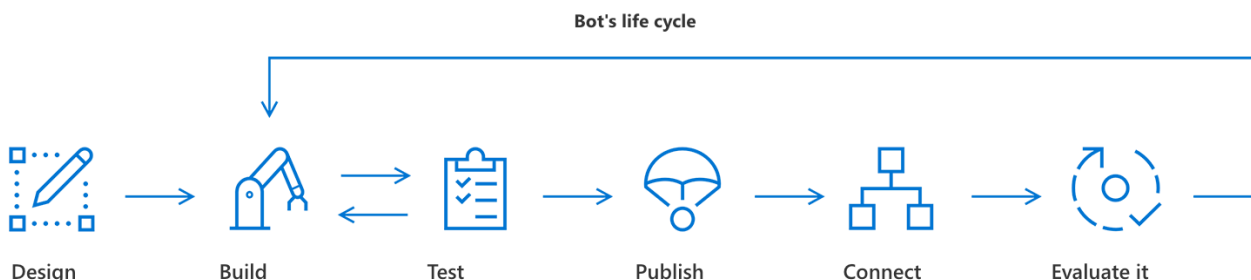
- Хоёр хувьсагчийн хоорондын нөлөөлөл
- Бие даасан хувьсагчийн тодорхой утга дахь хамааралтай хувьсагчийн утга
- Вариацийн нэгэн төрлийн байдал
- Ажиглалтын бие даасан байдал
- Нормал байдал

2.3.5 Comma Separated Values - CSV файл

CSV нь өгөгдлийн утгуудыг тусгаарлахад таслал ашигладаг текст файл юм. Файлын мөр бүр нь өгөгдөл байдаг бөгөөд харгалзах утгуудад текст файлыг бичих энгийн өгөгдөл хадгалах технологи юм. Их өгөгдөлтэй хялбар харьцах боломжийг олгодгоороо давуу талтай.

2.3.6 Microsoft Bot Framework

*Bot Framework*³ нь Microsoft-ийн *Azure Bot Service*-ийн тусламжтайгаар чатботыг турших, үүсгэх, удирдах, хэрэглээнд нэвтрүүлэх гэх мэт боломжуудыг нэг дор хангаж өгдөг. Энэхүү боломжуудын хүрээнд асуулт хариултыг зохицуулах, хэрэглэгчид зориулсан User Interface бүтээх, Language Understanding аргыг ашиглах гэх мэт үйлдлүүдийг хийх боломжтой. Bot бүтээх үйл явцыг Azure Bot Service болон Bot Framework нь ихэд хөнгөвчилж өгдөг бөгөөд доорх зурагт үзүүлсэн дарааллын дагуу Bot системийг бүтээдэг.



Зураг 2.9: Чатботын амьралын мөчлөг

Design

Design буюу загварчлах нь төслийн төлөвлөгөөг гаргах юм. Өөрөөр хэлбэл, системийн зорилго, үйл явц, хэрэглэгчийн хэрэгцээг сайтар судлах нь амжилттай Bot систем бүтээх чухал хэсэг юм.

Build

Бот системийг угсрах буюу хөгжүүлэх үйл явц юм. Энэ алхамд хөгжүүлэгч хэрэглэгчийн

³<https://dev.botframework.com/>

харагдах хэсгийг загварчлах бөгөөд хөгжүүлэлтийн орчин нь *Azure Portal*, JavaScript, Python болон C програмчлалын хэлүүдээс сонгож хөгжүүлэлтийг гүйцэтгэх явц юм. Мөн системийн шаардлагыг тодорхойлсны дагуу бот системийг өргөжүүлж ашиглах боломжтой бөгөөд тэдгээрээс дурдвал:

- Эх хэлний боловсруулалт (NLP)
- Асуулт хариултыг сайжруулан мэдлэгийн сан үүсгэх
- Хэрэглэгчийн интерфэйсийг сайжруулах

Test

Програм хангамжийн хөгжүүлэлтийн амьдралын мөчлөгийн адилаар тестийн үйл явцыг алгасаж болохгүй. Нэгэнт хэрэглэгчийн гарт бот системийг оруулахаас өмнө гарч болох алдаа дутагдлыг засан сайжруулах шаардлагатай. Иймд Bot системийг publish хийхээс өмнө заавал туршиж үзэх шаардлагатай. Энд Microsoft-ийн өөрсдийнх нь бие даасан програм болох *Bot emulator*-ийг ашиглан хөгжүүлэлтийн орчинд туршиж үзэх боломжийг олгодог.

Publish

Тестийн шатны дараа бот систем ашиглахад бэлэн болсон гэж үзсэн үед төсөл эсвэл чатботыг олон нийтэд ил болгох явдал юм.

Connect

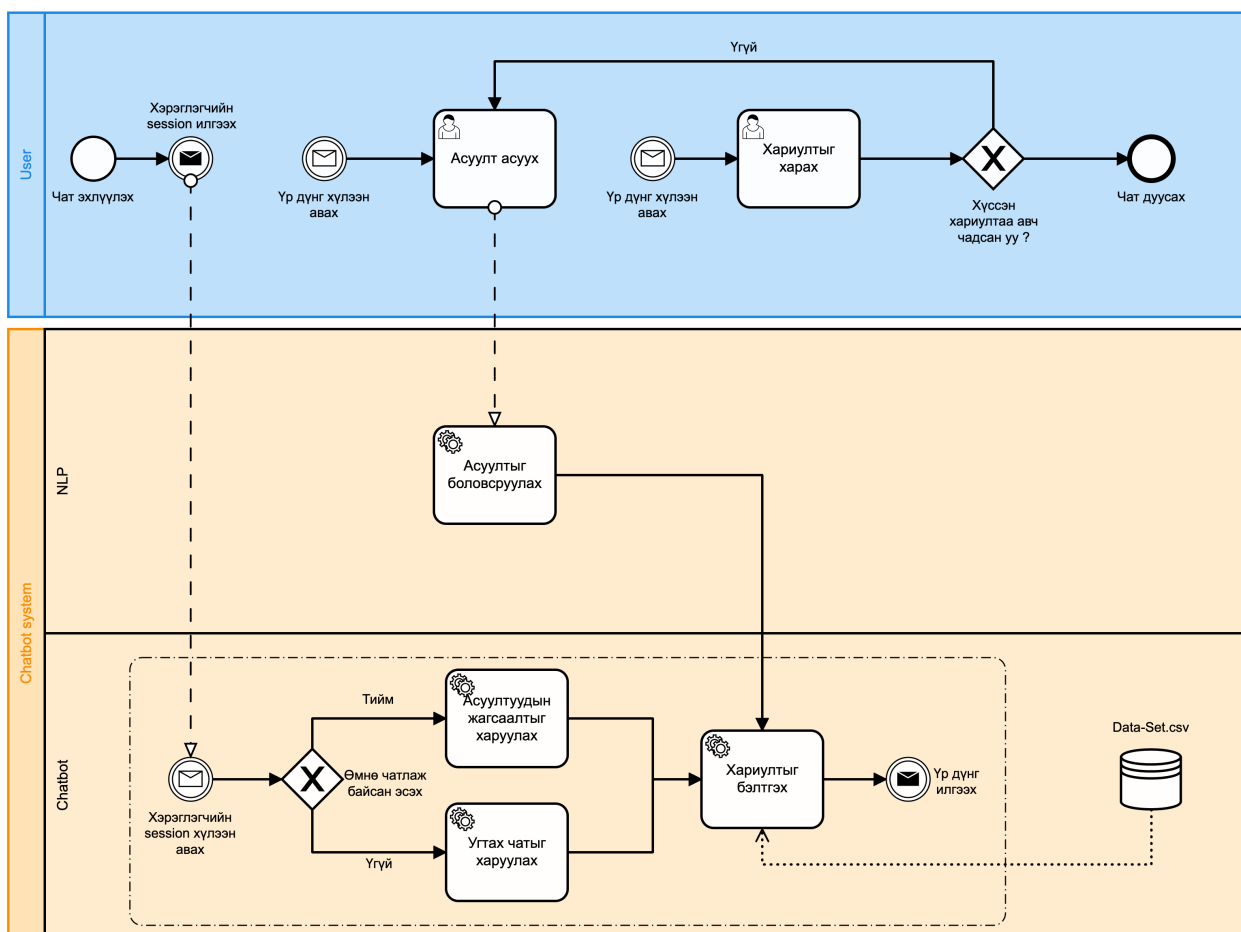
Bot системээ Facebook messenger, Microsoft Teams, Telegram, Skype гэх мэт чат сувгуудыг өөрийн Bot-той холбоно.

Ингэж бүх мөчлөгийг дууссаны дараа хөгжүүлэгч хэрэглэгчийн ашиглаж буй байдал дээр анализ хийж системийг дахин сайжруулах боломжтой бөгөөд буцаад угсрах үйл явцруу шилжих юм.

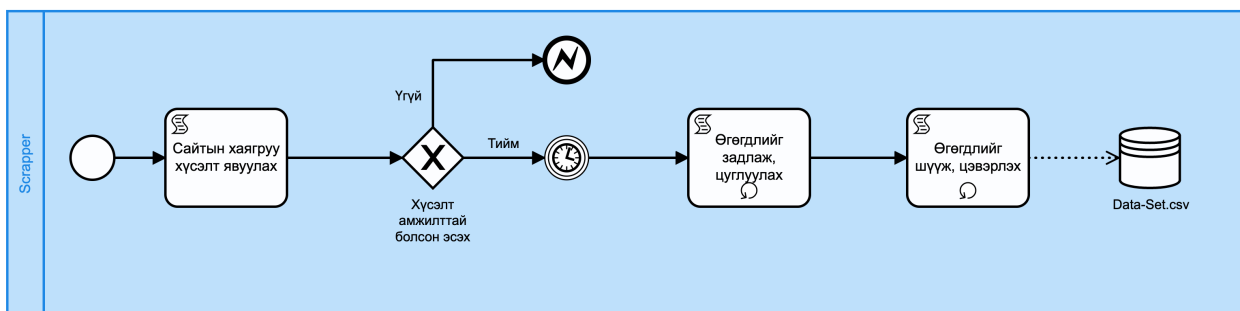
3. СИСТЕМИЙН ШИНЖИЛГЭЭ

3.1 Бизнесийн үйл ажиллагааны шинжилгээ

Бизнес процессийн модель нь чатбот системийн үндсэн процесс буюу үйл ажиллагааны явцыг BPMN-2.0 ашиглан дүрслэн харуулав [5]. Диаграмд дүрслэхдээ оролцогч талууд болох системүүдийг тус тусын *pool* дотор дүрсэлсэн бол дэд процесс буюу *subprocess*-ийг *lane*-д дүрсэлж хоорондын хамаарлыг харууллаа.



Зураг 3.1: BPMN-1



Зураг 3.2: BPMN-2

3.2 Хэрэглэгч

Чатбот системийг ямар ч хүн хэрэглэх боломжтой бөгөөд олон нийтэд нээлттэй байна. Системийн гол зорилго нь ажил хайж буй хэрэглэгчдэд ажлын байрны цогц мэдээллийг олгох зорилготой байх тул хэрэглэгчдийг дараах байдлаар тодорхойлж болно. Үүнд:

- Ажлын байр хайж буй хүн
- Хөгжүүлэгч

3.3 Функционал шаардлага

Дараах хэсэгт чатбот системд тавигдах функционал шаардлагуудыг харуулсан болно.

ФШ 1 Чатбот нь харилцан яриа эхэлмэгц хариу өгдөг байна.

ФШ 2 Чатбот нь ямар ч оролтод хариу өгнө.

ФШ 3 Хэрэв чатбот нь оролтод хариу өгч чадхааргүй байвал бусад асуултуудыг санал болгож ойлгомжгүй утга оруулсныг илэрхийлнэ.

ФШ 4 Чатботын санал болгох асуултууд нь цэс хэлбэртэй харагдана.

ФШ 5 Чатботын цэсэн дээр нэг товшилтоор асуултын хариултыг харуулдаг байна.

ФШ 6 Алхам бүрт үндсэн цэсрүү буцах сонголтыг харуулдаг байна.

ФШ 7 Чатботны хариулт нь текстэн хэлбэрээр хэрэглэгчид харагдана.

ФШ 8 Чатбот нь зөвхөн Монголоор бичсэн асуултад хариулт өгнө.

ФШ 9 Чатбот нь дэлгэрэнгүй мэдээллийг цэс хэлбэрээр сонгуулан харуулж чаддаг байна.

3.4 Функционал бус шаардлага

Бэлэн болон найдвартай байдал (Availability & Reliability)

ФБШ 01 Чатбот систем өдрийн аль ч цагт 99.999% ажиллагаатай байх ёстой.

ФБШ 02 Ямар ч хүсэлт ирсэн чатбот 100% хариу өгдөг байна.

Гүйцэтгэлтэй байдал (Performance)

ФБШ 03 Чатботын байршуулсан сувагт, шаардлагаас хамаарч ямар ч төхөөрөмжөөс хандаж болно.

ФБШ 04 Зарим тохиолдолд чатботын гүйцэтгэл нь хэрэглэгчийн интернет болон төхөөрөмжийн үйлдлийн системийн хувилбараас хамаарч болно.

Дэмжих чадвар (Supportability)

ФБШ 05 Чатботын эх кодыг *github* дээр нээлттэй эхийн систем хэлбэрээр байршуулна.

Хэрэгцээт байдал (Usability)

ФБШ 06 Чатбот нь хэрэглэхэд хялбар, ойлгомжтой байна.

ФБШ 07 Чатботны цэс нь ойлгомжтой цөөн үгээр илэрхийлэгдсэн байна.

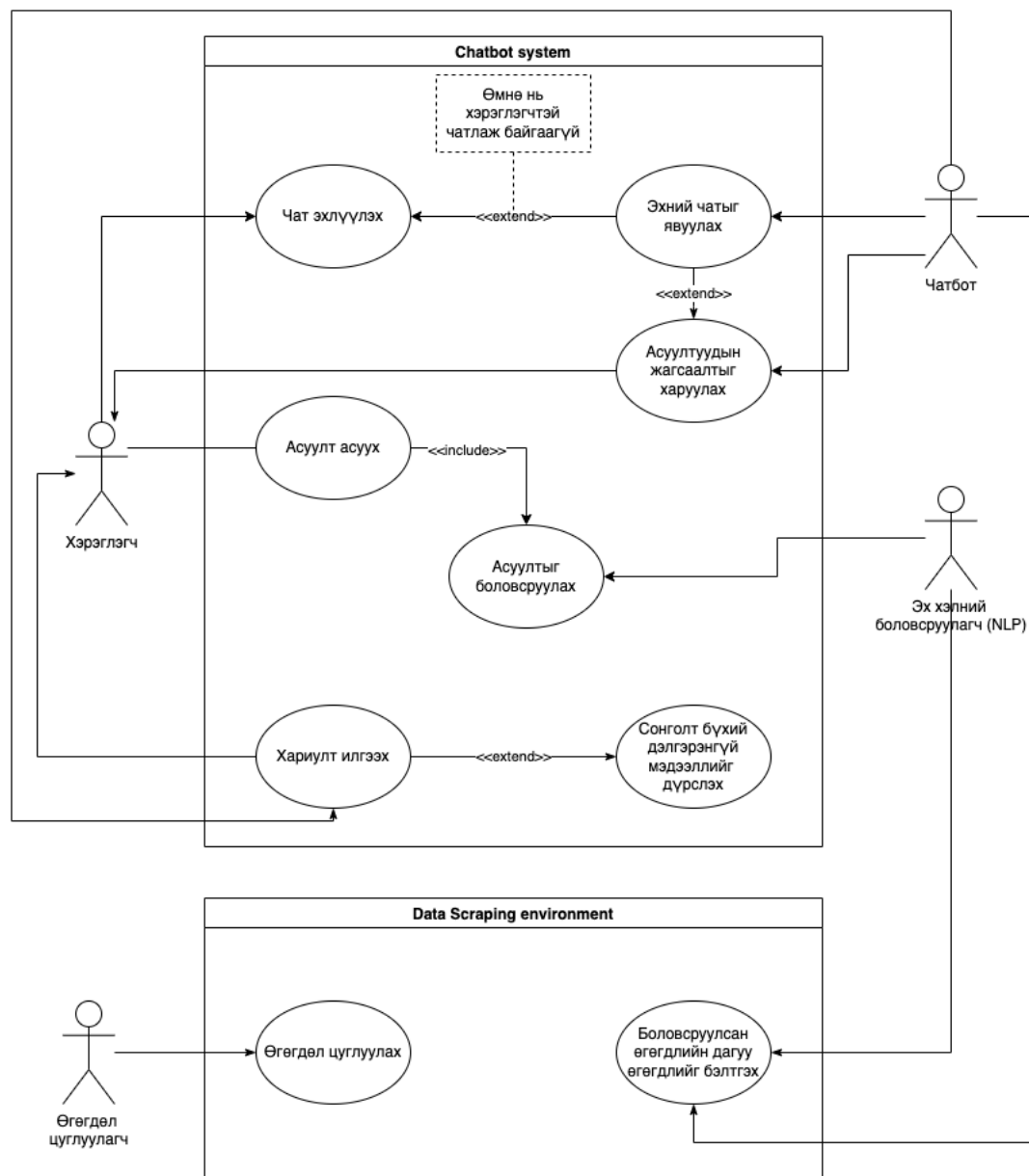
ФБШ 08 Чатботны цэсийн хэмжээ дарагдахуйц том байна.

Аюулгүй байдал(Security)

ФБШ 09 Чатбот системийн байршуулсан сувгийн стандартын дагуу хэрэглэгчийн мэдээллийг өгөгдлийн санд хадгалахгүй байна.

3.5 Use case диаграм

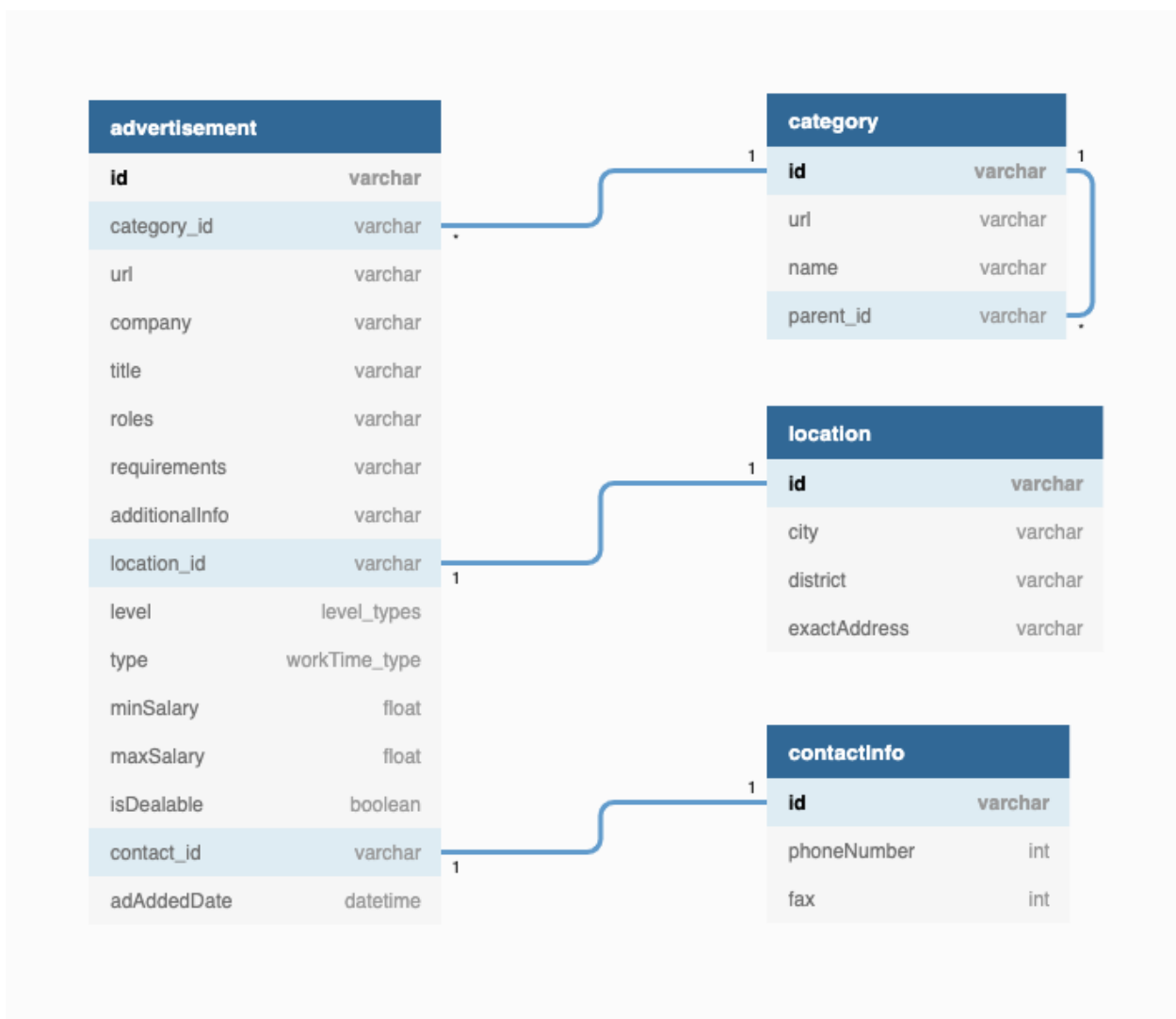
Чатбот системийн use-case диаграммыг байдлаар тодорхойлов [4].



Зураг 3.3: Use Case диаграм

4. СИСТЕМИЙН ЗОХИОМЖ

4.1 Өгөгдлийн сангийн диаграм



Зураг 4.1: Өгөгдлийн сангийн диаграм

4.2 Өгөгдлийн элемент

Чатбот системийн өгөгдлийн сангийн диаграмд харуулсан хүснэгтүүдэд агуулагдах мэдээлэл болон үүргийн талаар дэлгэрэнгүй тайлбарласан болно.

4.2.1 *advertisement* - Ажлын байрны зар

Ажлын байрны зар нь ямар категори буюу ангилалд, ямар холбоо барих хаягийн хамтаар хадгалагдаж буй мэдээлэл болон бусад дэлгэрэнгүй мэдээллийг харуулсан байна.

Table 4.1: advertisement хүснэгт

№	Баганын нэр	Түлхүүр өгөгдөл	Өгөгдлийн төрөл	Хоосон утга	Тайлбар
1	id	PK	varchar	not null	Ажлын байрны зарын дахин давтагдашгүй дугаар
2	category_id	FK	varchar	not null	Ажлын байрны зард хамаарах ангиллын дугаар
3	url		varchar	not null	Ажлын байрны зарын хаяг
4	company		varchar	not null	Ажил олгогч компани / хүн
5	title		varchar	not null	Ажлын зарын гарчиг
6	roles		varchar	null	Гүйцэтгэхүндсэн үүрэг
7	requirements		varchar	null	Ажлын байранд тавигдах шаардлага
8	additionalInfo		varchar	null	Нэмэлт мэдээлэл
9	location_id	FK	varchar	not null	Ажлын байрны зард хамаарах ангиллын дугаар
10	level		level_types	null	Ажлын түвшин
11	type		workTime_type	null	Ажиллах цагийн төрөл

№	Баганын нэр	Түлхүүр өгөгдөл	Өгөгдлийн төрөл	Хоосон утга	Тайлбар
12	minSalary		float	null	Доод цалин
13	maxSalary		float	null	Дээд цалин
14	isDeable		boolean	null	Тохиролцох эсэх
15	contact_id	FK	varchar	not null	Ажлын байрны зард хамаарах холбоо барих хаягийн дугаар
16	adAddedDate		datetime	not null	Зар нийтэлсэн огноо

Энд *level* буюу ажлын түвшин, *type* буюу ажлын цагийн өгөгдлийн төрлийг тодорхойлохдоо дараах байдлаар зааж өгсөн.

Enum level_types буюу ажлын түвшний шаардлага нь дараах үндсэн 4 өгөгдлийн төрлөөс хамаарна:

- student - Оюутан / дадлагажигч
- professional - Мэргэжлийн
- occupasionDoesntRequire - Мэргэжил шаардахгүй
- intermediateManagemet - Дунд шатны удирдлага

workTime_type буюу ажиллах цагийн нөхцөл нь дараах үндсэн 4 өгөгдлийн төрлөөс хамаарна:

- shift - Ээлжийн
- fullTime - Бүтэн цагийн
- halfTime - Хагас цагийн
- contract - Гэрээт / зөвлөх

4.2.2 category - Ангилал

Ажлын байрны зарын бүх ангиллуудын хаяг болон нэрийн мэдээллийг хадгалах хүснэгт юм. Ангиллууд нь дэд ангилал байж болох учир түүнийг эцэг ангиллын дугаарыг хадгалах байдлаар зохиомжлов.

Table 4.2: category хүснэгт

№	Баганын нэр	Түлхүүр өгөгдөл	Өгөгдлийн төрөл	Хоосон Утга	Тайлбар
1	id	PK	varchar	not null	Ажлын байрны зарын ангиллын дугаар
2	url		varchar	not null	Ангиллын хаяг
3	name		varchar	not null	Ангиллын нэр
4	parent_id	FK	varchar	null	Эцэг ангиллын дугаар

4.2.3 location - Байршил

Ажлын байрны байршил болон хот, аймаг, дүүргийн дэлгэрэнгүй өгөгдлийг хадгална.

Table 4.3: location хүснэгт

№	Баганын нэр	Түлхүүр өгөгдөл	Өгөгдлийн төрөл	Хоосон Утга	Тайлбар
1	id	PK	varchar	not null	Ажлын байрны зарын хаягийн дугаар
2	city		varchar	null	Ажлын байрны зарын байрших хот, аймгийн нэр

№	Баганын нэр	Түлхүүр өгөгдөл	Өгөгдлийн төрөл	Хоосон Утга	Тайлбар
3	district		varchar	null	Ажлын байрны зарын байрших дүүрэг, сумын нэр
4	exactAddress		varchar	null	Дэлгэрэнгүй хаяг

4.2.4 contactInfo - Холбоо барих

Ажлын байр олгогчийн хаягийн дэлгэрэнгүй өгөгдлийг хадгалана.

Table 4.4: contactInfo хүснэгт

№	Баганын нэр	Түлхүүр өгөгдөл	Өгөгдлийн төрөл	Хоосон Утга	Тайлбар
1	id	PK	varchar	not null	Ажил олгогчтой холбоо барих хаягийн дугаар
2	phoneNumber		int	null	Ажил олгогчийн утасны дугаар
3	fax		int	null	Ажил олгогчийн факс дугаар

4.3 Өгөгдлийн сангийн холбоосын тайлбар

- Нэг ангилал буюу категорид олон ажлын байрны зар байж болно.
- Нэг ангилал буюу категорид олон категори байж болно.
- Нэг ажлын байрны зард нэг байршлын мэдээлэл байна.
- Нэг ажлын байрны зард нэг холбоо барих хаягийн мэдээлэл байна.

5. ХЭРЭГЖҮҮЛЭЛТ, ҮР ДҮН

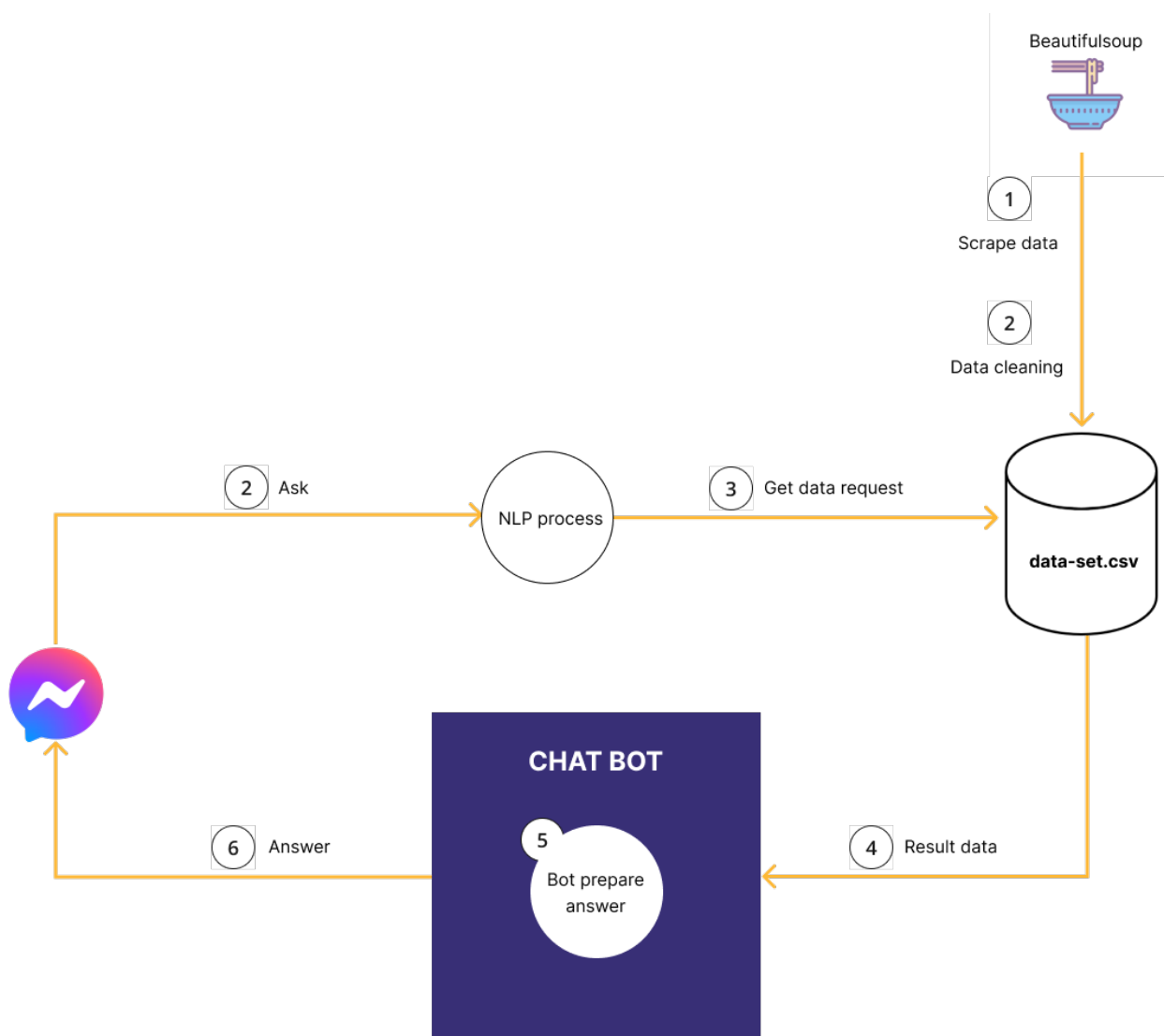
5.1 Хөгжүүлсэн байдал

Чатбот системийн хөгжүүлэлтийг хийхдээ шаардлагууд дээр үндэслэн, үечилсэн төлөвлөгөө болон шаардлагатай хөгжүүлэлтийг дэс дараалалтайгаар хийж гүйцэтгэсэн.

- Өгөгдөл цуглуулах
- Өгөгдлийг нэгтгэх, цэвэрлэх
- Системийн шаардлага, үйл ажиллагааг тодорхойлох
- Өгөгдөлд анализ хийх
- Эх хэлний боловсруулалт хийх
- Чатбот хөгжүүлэх

гэсэн дарааллын дагуу хөгжүүлэлтийг хийсэн болно.

Доорх зурагт чатбот системийн үндсэн процессийн зураглал харагдаж байна.



Зураг 5.1: Үндсэн процесс зураглал

5.1.1 Өгөгдөл цуглуулах

Үндсэн ашиглагдах өгөгдөл болох ажил олгогчид, ажлын байрны өгөгдлийг **zangia.mn**-ээс BeautifulSoup ашиглан авсан. Эхлээд вебсайтынхаа HTML бүтцийг нь судалж, авах өгөгдлийнхөө класс утгуудыг (className) олж авах нь зөв юм. Вебсайтаас өгөгдөл цуглуулах 2 үндсэн арга байдгаас өгөгдлийг олж илрүүлж, хаягийг цуглуулах (data crawling) аргаар бүх ангиллуудын хаяг (url)-уудын түүж авна. Харин data scraping нь тэр хооронд олсон бүх хаягуудаараа явж

хэрэгтэй агуулгыг цуглуулна.¹

```
1  initialUrl = 'https://www.zangia.mn/'
2  today = str(date.today())
3  # all categories set
4  categorySet = set()
5  # all advertisement's link set
6  adUrlDict = {}
7  # all ads object set
8  adsSet = set()
9
10 # scrape initial links
11 soup = useScrape(initialUrl)
12 navigatorList = soup.find_all('div', class_='filter')
13 for navigator in navigatorList:
14     if navigator.find('h3').text.strip() != 'Salbar, mergejil':
15         continue
16     # ALL CATEGORY LINKS
17     categoryList = navigator.find_all('div')
18
19 for categoryItem in categoryList:
20     categories = categoryItem.find('a')
21     url = initialUrl + categories['href']
22     tempCategory = Category(url, categories.text, '')
23     soup = useScrape(url)
24     subCategory = soup.find('div', class_='pros')
25     # ALL SUBCATEGORY LINKS
26     subCategoryList = subCategory.find_all('a')
```

¹Кодын жишээг оруулахад utf-8 формат танихгүй байсан тул монголоос галиглаж бичсэн болно.

```

27     for subCategoryItem in subCategoryList:
28         subCategoryUrl = initialUrl + subCategoryItem['href']
29         tempSubCategory = Category(
30             subCategoryUrl, subCategoryItem.text, tempCategory.name)
31         categorySet.add(tempSubCategory)

```

Код 5.1: Data Link crawling

Дээрх код нь эхлээд вебсайтруу орж “filter” класс доторх “Салбар, мэргэжил” гэсэн хэсгээс бүх эцэг категориудыг data crawling хийж авч байна. Үүний дараа хүүхэд категориудыг олж categorySet дотор бүх хаягуудыг хийж хадгалж байна.² Энд categorySet set-ийн элемент нь category төрлийн объект бөгөөд өгөгдлийн сангийн диаграм дээр тодорхойлж өгсөн байгаа. Ингэснээр data crawling-ийг зогсоож, цуглуулсан хаягаасаа өгөгдлөө цуглуулъя.

```

1  for categoryItem in categorySet:
2      if categoryItem.parentId == '':
3          continue
4      soup = useScrape(categoryItem.url)
5      hasPagination = soup.find('div', class_='page-link')
6      pagesUrl = []
7      if hasPagination != None:
8          pagesUrl = createLinkList(hasPagination, categoryItem.url)
9      else:
10         pagesUrl.append(categoryItem.url)
11     for pageUrl in pagesUrl:
12         soup = useScrape(pageUrl)
13         ads = soup.find_all('div', class_='ad')
14         # CREATE UNIQUE AD DICTIONARY
15         for ad in ads:

```

²Python хэлний set өгөгдлийн төрөл нь давхацахгүй утгуудын хүснэгт гэж хэлж болно.

```

16         adUrl = initialUrl+ad.find('a', class_=None)['href']
17         adUrlDict[adUrl] = categoryItem
18     pagesUrl.clear()

```

Код 5.2: Өгөгдөл цуглуулах

Дээрх кодонд бүх хүүхэд категориудын дотор агуулагдаж буй зарын мэдээллийг цуглуулж байна. Ингэхдээ эхлээд категори доторх өгөгдлүүд нь хуудаслагдсан (pagination) байх боломжтой бөгөөд хэрэв олон хуудастай байвал хаягуудыг нь угсарч тэдгээрээс ч мөн өгөгдлийг нь цуглуулах ёстой юм.

```

1  from .scrape import UseBeautifulSoup as useScrape
2
3  def createLinkList(pagination, url) -> array:
4      linkList = []
5      total = int(useRegex(pagination.find_all('a')[-1]['href']))
6
7      for i in range(total + 1):
8          if i == 0:
9              continue
10         link = url + '/pg.' + str(i)
11         linkList.append(link)
12     return linkList

```

Код 5.3: Хуудаслалтыг задлах

Энэ хэсэгт хуудаслан дугаарласан хэсгийн хамгийн сүүлийн тоог авч *createLinkList* функцруу дамжуулснаар тухайн категорийн бүх өгөгдлийг цуглуулах боломж үүсч байгаа юм. Ингээд дахин data crawling хийж бүх хаягуудыг цуглуулж энэ удаад dictionary үүсгэж зарын хаягуудыг хадгалсан. Энд dictionary үүсгэхдээ хаягийг нь түлхүүр (key) болгож категори объектыг нь утга(value) болгож хадгалсан. Мэдээж хэрэг dictionary нь түлхүүр давхцахаас сэргийлдэг тул

бид ямар нэгэн байдлаар нэг зарын өгөгдлийг 2 удаа цуглуулах эрсдэлгүй болж байна.³

Харин одоо үүсгэсэн dictionary-оо ашиглан өгөгдлөө CSV файлуугаа бичихэд ашиглаж болно.

```
1 file = open(today+'adScrape.csv', 'w', encoding='utf-8')
2 file.write('Parent Category Name' + '\t' +
3           'Category Name ' + '\t' +
4           'Link' + '\t' +
5           'Employee Company' + '\t' +
6           'Title' + '\t' +
7           'Roles' + '\t' +
8           'Requirements' + '\t' +
9           'Additional Info' + '\t' +
10          'City/Province' + '\t' +
11          'District' + '\t' +
12          'Level' + '\t' +
13          'Type' + '\t' +
14          'Min Salary' + '\t' +
15          'Max Salary' + '\t' +
16          'Is Dealable' + '\t' +
17          'Address' + '\t' +
18          'Phone' + '\t' +
19          'Fax' + '\t' +
20          'Ad Added Date' + '\n')
21
22 for adUrl in adUrlDict:
23     print(adUrl)
24     try:
```

³Нэг зарын өгөгдлийг цуглуулахад интернетийн хурдаас хамааран 0.2-оос 0.5 секунд хугацаа зарцуулдаг

```

25     tempAdItem = useAdScrape(adUrl)
26     tempAdItem.setCategory(adUrlDict[adUrl])
27     file.write(
28         tempAdItem.category.parentId+'\t' +
29         tempAdItem.category.name+'\t' +
30         tempAdItem.url+'\t' +
31         tempAdItem.company+'\t' +
32         tempAdItem.title+'\t' +
33         tempAdItem.roles+'\t' +
34         tempAdItem.requirements+'\t' +
35         tempAdItem.additionalInfo+'\t' +
36         tempAdItem.city+'\t' +
37         tempAdItem.district+'\t' +
38         tempAdItem.level+'\t' +
39         tempAdItem.type+'\t' +
40         tempAdItem.minSalary+'\t' +
41         tempAdItem.maxSalary+'\t' +
42         tempAdItem.isDealable+'\t' +
43         tempAdItem.address+'\t' +
44         tempAdItem.phoneNumber+'\t' +
45         tempAdItem.fax+'\t' +
46         tempAdItem.adAddedDate+'\n')
47     del tempAdItem
48 except:
49     print('Ad writing error')
50 file.close()

```

Код 5.4: CSV файлуу хадгалах

Дээрх код нь энгийн python програм файльтай харьцаж өөрт цуглуулсан өгөгдлөө хадгалж

байна. Нийт өгөгдлийн хүснэгтийг энд ⁴ оруулав.

A		B	C	D		E		F		G		H		I		J		K		L		M		N		O		P		Q		R		S		T		U		V		W		X		Y		Z		AA		AB		AC		AD		AE		AF		AG		AH		AI		AJ		AK		AL		AM		AN		AO		AP		AQ		AR		AS		AT		AU		AV		AW		AX		AY		AZ		BA		BB		BC		BD		BE		BF		BG		BH		BI		BJ		BK		BL		BM		BN		BO		BP		BQ		BR		BS		BT		BU		BV		BW		BX		BY		BZ		CA		CB		CC		CD		CE		CF		CG		CH		CI		CJ		CK		CL		CM		CN		CO		CP		CQ		CR		CS		CT		CU		CV		CW		CX		CY		CZ		DA		DB		DC		DD		DE		DF		DG		DH		DI		DJ		DK		DL		DM		DN		DO		DP		DQ		DR		DS		DT		DU		DV		DW		DX		DY		DZ		EA		EB		EC		ED		EE		EF		EG		EH		EI		EJ		EK		EL		EM		EN		EO		EP		EQ		ER		ES		ET		EU		EV		EW		EX		EY		EZ		FA		FB		FC		FD		FE		FF		FG		FH		FI		FJ		FK		FL		FM		FN		FO		FP		FQ		FR		FS		FT		FU		FV		FW		FX		FY		FZ		GA		GB		GC		GD		GE		GF		GG		GH		GI		GJ		GK		GL		GM		GN		GO		GP		GQ		GR		GS		GT		GU		GV		GW		GX		GY		GZ		HA		HB		HC		HD		HE		HF		HG		HH		HI		HJ		HK		HL		HM		HN		HO		HP		HQ		HR		HS		HT		HU		HV		HW		HX		HY		HZ		IA		IB		IC		ID		IE		IF		IG		IH		II		IJ		IK		IL		IM		IN		IO		IP		IQ		IR		IS		IT		IU		IV		IW		IX		IY		IZ		JA		JB		JC		JD		JE		JF		JG		JH		JI		JJ		JK		JL		JM		JN		JO		JP		JQ		JR		JS		JT		JU		JV		JW		JX		JY		JZ		KA		KB		KC		KD		KE		KF		KG		KH		KI		KJ		KL		KM		KN		KO		KP		KQ		KR		KS		KT		KU		KV		KW		KX		KY		KZ		LA		LB		LC		LD		LE		LF		LG		LH		LI		LJ		LK		LL		LM		LN		LO		LP		LQ		LR		LS		LT		LU		LV		LW		LX		LY		LZ		MA		MB		MC		MD		ME		MF		MG		MH		MI		MJ		MK		ML		MN		MO		MP		MQ		MR		MS		MT		MU		MV		MW		MX		MY		MZ		NA		NB		NC		ND		NE		NF		NG		NH		NI		NJ		NK		NL		NM		NN		NO		NP		NQ		NR		NS		NT		NU		NV		NW		NX		NY		NZ		OA		OB		OC		OD		OE		OF		OG		OH		OI		OJ		OK		OL		OM		ON		OO		OP		OQ		OR		OS		OT		OU		OV		OW		OX		OY		OZ		PA		PB		PC		PD		PE		PF		PG		PH		PI		PJ		PK		PL		PM		PN		PO		PP		PQ		PR		PS		PT		PU		PV		PW		PX		PY		PZ		QA		QB		QC		QD		QE		QF		QG		QH		QI		QJ		QK		QL		QM		QN		QO		QP		QQ		QR		QS		QT		QU		QV		QW		QX		QY		QZ		RA		RB		RC		RD		RE		RF		RG		RH		RI		RJ		RK		RL		RM		RN		RO		RP		RQ		RR		RS		RT		RU		RV		RW		RX		RY		RZ		SA		SB		SC		SD		SE		SF		SG		SH		SI		SJ		SK		SL		SM		SN		SO		SP		SQ		SR		SS		ST		SU		SV		SW		SX		SY		SZ		TA		TB		TC		TD		TE		TF		TG		TH		TI		TJ		TK		TL		TM		TN		TO		TP		TQ		TR		TS		TT		TU		TV		TW		TX		TY		TZ		UA		UB		UC		UD		UE		UF		UG		UH		UI		UJ		UK		UL		UM		UN		UO		UP		UQ		UR		US		UT		UU		UV		UW		UX		UY		UZ		VA		VB		VC		VD		VE		VF		VG		VH		VI		VJ		VK		VL		VM		VN		VO		VP		VQ		VR		VS		VT		VU		VV		VW		VX		VY		VZ		WA		WB		WC		WD		WE		WF		WG		WH		WI		WJ		WK		WL		WM		WN		WO		WP		WQ		WR		WS		WT		WU		WV		WW		WX		WY		WZ		XA		XB		XC		XD		XE		XF		XG		XH		XI		XJ		XK		XL		XM		XN		XO		XP		XQ		XR		XS		XT		XU		XV		XW		XX		XY		XZ		YA		YB		YC		YD		YE		YF		YG		YH		YI		YJ		YK		YL		YM		YN		YO		YP		YQ		YR		YS		YT		YU		YV		YW		YX		YY		YZ		ZA		ZB		ZC		ZD		ZE		ZF		ZG		ZH		ZI		ZJ		ZK		ZL		ZM		ZN		ZO		ZP		ZQ		ZR		ZS		ZT		ZU		ZV		ZW		ZX		ZY		ZZ	
1	Parent Category Name	Category Name	Link	Employees Company	Title	Level	Type	Min Salary	Max Salary	Is Available	City/Province	District	Notes	Requirements	Additional Info																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
1	Маркетинг, PR, Менеджмент	PR, опции нейтинге карьерного роста	https://www.linkedin.com/company/100																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

Зупаг 5.2: Data set

Энд хамгийн сүүлд буюу 3 сарын 31нд өгөгдлийн цуглуулга хийж 9175 өгөгдлийн excel хэлбэрт оруулсныг харж болж байна.

5.1.2 Өгөгдлийг нэгтгэх, цэвэрлэх

Ихэнх цуглуулсан өгөгдөл нь өгөгдлийн сангийн диаграмын дагуу амжилттай цуглуулсан бөгөөд дүн шинжилгээ хийх боломжтой өгөгдлүүдийг тусад нь хадгалж ашигласан болно.

⁴<https://docs.google.com/spreadsheets/d/1rtATUKhUlleIKaWgFGvqiUWMipsrv-aCWZk-tYmzezU/>

edit?usp=sharing

Bibliography

- [1] Чатбот системийн тухай
<https://www.engati.com/blog/types-of-chatbots-and-their-applications>
- [2] Өгүүлбэр хувиргалтын арга зүй
<https://www.sbert.net/docs/quickstart.html>
- [3] Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks
<https://arxiv.org/abs/1908.10084>
- [4] Use case diagram
https://app.diagrams.net/#G1jhom3sc_holt-X9XLALtQja_Gl_Eykhj
- [5] Business Process Model Notation 2.0 диаграм
<https://cawemo.com/diagrams/ea037ec0-c1c5-4ab6-8262-521657472803--bpmn-2-0?v=960,418,1>
- [6] Өгөгдлийн сангийн диаграм
<https://dbdiagram.io/d/6249fb7cd043196e39e87451>

А. ҮЕЧИЛСЭН ТӨЛӨВЛӨГӨӨ

Батлав.

МКУТ-ийн эрхлэгч:...../док. проф. Н.Оюун-Эрдэнэ/

2022 оны 02 сарын 11

Монгол нэр

Ажил олгогчдын өгөгдлийн анализ систем дээр суурилсан чат бот

Англи нэр

Chat bot based on system analysis of employers' data

Сэдэвт бакалаврын судалгааны ажлын 7 хоногийн үечилсэн төлөвлөгөө

Хугацаа: 2022.02.07-оос 2022.05.06 хүртэл 13 долоо хоног

№	Хийх ажил	Долоо хоног										12 Урьдчилсан хамгаалалт	13	14 Жинхэнэ хамгаалалт	Тайлбар
	Онолын судалгаа														
1	Онолын судалгаа	Scrapper tool													
		Bot tool													
2	Өгөгдөл цуглуулалт	Цуглуулах код бичих													
		Өгөгдлийг бааруулах													
3	Системийн шинжилгээ	Системийн шардлага тодорхойлох													
		Хэрэглэгчийн шардлага тодорхойлох													
5	Системийн зохиомж	Өгөгдлийн сэлтийн зохиомж													
		Чат бот хөгжүүлэлт													
6	Хэрэгжүүлэлт	Өгөгдөлд анализ хийх хайгуурлах													
7	Бичиг баримт	Тайлан боловсруулах													

Тайлбар: Төслийг гэрээжүүлэх төлөвлөгөөг 7 хоногийн дотоод хугацаагаар хийж төв газраар бүрдэж гүйцэтгэнэ. Хийх ажил дэд хэсгийн байдал үз ажилд зориулах хугацааг хувиар тэмдэглэж байна. Ажлын эхлэх өдөр хугацааг хувиар тэмдэглэж байна. Ажлын эхлэх өдөр хугацааг хувиар тэмдэглэж байна.

Зөвшөөрсөн: Удирдагч багш/Б. Хуягбаатар/

Боловсруулсан: Оюутин/Мэдээллийн технологи А. Сайнболбоо/

Оюутны ID: 1851num1762

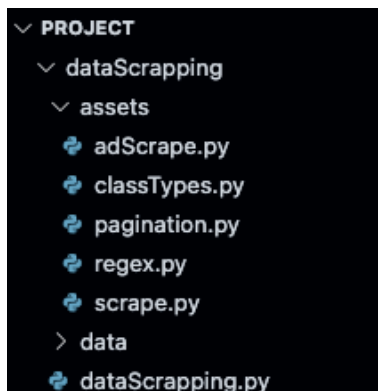
Холбогдох утас: 91990388

Зураг А.1: Бакалаврын судалгааны ажлын үечилсэн төлөвлөгөө

В. КОДЫН ХЭРЭГЖҮҮЛЭЛТ

В.1 Өгөгдөл цуглуулалт

Өгөгдөл цуглуулах програм нь дараах бүтэцтэй байх бөгөөд assets доторх кодууд нь үндсэн кодыг ажлуулахад туслах функцууд байна.



Зураг В.1: Фолдерийн бүтэц

В.1.1 Үндсэн өгөгдлийг цуглуулах эх код

```
1 from datetime import date
2 import time
3 from assets.classTypes import Category
4 from assets.scrape import UseBeautifulSoup as useScrape
5 from assets.adScrape import advertisementScrape as useAdScrape
6 from assets.pagination import createLinkList as createLinkList
7
8 start_time = time.time()
9 initialUrl = 'https://www.zangia.mn/'
10 today = str(date.today())
11 # all categories set
12 categorySet = set()
13 # all advertisement's link set
14 adUrlDict = {}
15 # all ads object set
16 adsSet = set()
17
18 # scrape initial links
19 soup = useScrape(initialUrl)
20 navigatorList = soup.find_all('div', class_='filter')
21 for navigator in navigatorList:
22     if navigator.find('h3').text.strip() != 'Salbar, mergejl':
23         continue
24     # ALL CATEGORY LINKS
25     categoryList = navigator.find_all('div')
```

```

26
27 for categoryItem in categoryList:
28     categories = categoryItem.find('a')
29     url = initialUrl + categories['href']
30     tempCategory = Category(url, categories.text, '')
31     soup = useScape(url)
32     subCategory = soup.find('div', class_='pros')
33     # ALL SUBCATEGORY LINKS
34     subCategoryList = subCategory.find_all('a')
35     for subCategoryItem in subCategoryList:
36         subCategoryUrl = initialUrl + subCategoryItem['href']
37         tempSubCategory = Category(
38             subCategoryUrl, subCategoryItem.text, tempCategory.name)
39         categorySet.add(tempSubCategory)
40
41 for categoryItem in categorySet:
42     if categoryItem.parentId == '':
43         continue
44     soup = useScape(categoryItem.url)
45     hasPagination = soup.find('div', class_='page-link')
46     pagesUrl = []
47     if hasPagination != None:
48         pagesUrl = createLinkList(hasPagination, categoryItem.url)
49     else:
50         pagesUrl.append(categoryItem.url)
51     for pageUrl in pagesUrl:
52         soup = useScape(pageUrl)
53         ads = soup.find_all('div', class_='ad')
54         # CREATE UNIQUE AD DICTIONARY
55         for ad in ads:
56             adUrl = initialUrl+ad.find('a', class_=None)['href']
57             adUrlDict[adUrl] = categoryItem
58     pagesUrl.clear()
59
60 file = open(today+'adScrape.csv', 'w', encoding='utf-8')
61 file.write('Parent Category Name' + '\t' +
62           'Category Name ' + '\t' +
63           'Link' + '\t' +
64           'Employee Company' + '\t' +
65           'Title' + '\t' +
66           'Roles' + '\t' +
67           'Requirements' + '\t' +
68           'Additional Info' + '\t' +
69           'City/Province' + '\t' +
70           'District' + '\t' +
71           'Level' + '\t' +
72           'Type' + '\t' +
73           'Min Salary' + '\t' +
74           'Max Salary' + '\t' +
75           'Is Dealable' + '\t' +
76           'Address' + '\t' +
77           'Phone' + '\t' +

```

```

78         'Fax' + '\t' +
79         'Ad Added Date' + '\n')
80
81 for adUrl in adUrlDict:
82     print(adUrl)
83     try:
84         tempAdItem = useAdScrape(adUrl)
85         tempAdItem.setCategory(adUrlDict[adUrl])
86         file.write(
87             tempAdItem.category.parentId+'\t' +
88             tempAdItem.category.name+'\t' +
89             tempAdItem.url+'\t' +
90             tempAdItem.company+'\t' +
91             tempAdItem.title+'\t' +
92             tempAdItem.roles+'\t' +
93             tempAdItem.requirements+'\t' +
94             tempAdItem.additionalInfo+'\t' +
95             tempAdItem.city+'\t' +
96             tempAdItem.district+'\t' +
97             tempAdItem.level+'\t' +
98             tempAdItem.type+'\t' +
99             tempAdItem.minSalary+'\t' +
100            tempAdItem.maxSalary+'\t' +
101            tempAdItem.isDealable+'\t' +
102            tempAdItem.address+'\t' +
103            tempAdItem.phoneNumber+'\t' +
104            tempAdItem.fax+'\t' +
105            tempAdItem.adAddedDate+'\n')
106         del tempAdItem
107     except:
108         print('Ad writing error')
109 file.close()
110 print("--- %s seconds ---" % (time.time() - start_time))

```

Код В.1: Бүх өгөгдлийг цуглуулах - dataScrapping.py

В.1.2 Нэг зарын шаардлагатай бүх мэдээллийг цуглуулах код

```

1 import re
2 from .classTypes import Advertisement
3 from .scrape import UseBeautifulSoup as useScrape
4
5
6 def listScraper(sections, key) -> str:
7     content = []
8     for section in sections:
9         subTitle = section.find('h2', class_=None).text
10        if key != subTitle:
11            continue
12        div = section.find('div', class_=None)
13        children = div.next_element
14

```

```

15         while(children != None):
16             try:
17                 content.append(textStrip(children.text))
18                 children = children.next_sibling
19                 continue
20             except:
21                 print('An error occurred')
22                 children = children.next_sibling
23             content = [s for s in filter(listFunc, content)]
24         if not content:
25             return ''
26         return ' '.join(content)
27
28
29 def textStrip(text) -> str:
30     pattern = re.compile('[\r\n\xa0\t ]+', re.MULTILINE | re.IGNORECASE)
31     return pattern.sub(' ', text.strip())
32
33
34 def listFunc(e):
35     return len(e) != 0
36
37
38 def singleItemScrapper(sections, key, subKey) -> str:
39     for section in sections:
40         subTitle = section.find('h2', class_=None).text
41         if key != subTitle:
42             continue
43         div = section.find_all('div', class_=None)
44         for item in div:
45             if item.next_element.text == subKey:
46                 return textStrip(item.find('span').text)
47     return 'None'
48
49
50 def salaryScrapper(salary):
51     isDealable = ''
52     k = re.split(r'[^d,]+', salary, 2, re.IGNORECASE)
53     if len(k) < 2:
54         [a] = k[0:1]
55         return a, a
56     [a, b] = k[0:2]
57     if len(k) > 2:
58         isDealable = ' '
59     return a, b, isDealable
60
61
62 def locationScrapper(location):
63     city = ''
64     district = ''
65     k = location.split(',')

```

```

66     if len(k) < 2:
67         city = k[0]
68         return city, district
69     [city, district] = k[0:2]
70     return city, district
71
72
73 def advertisementScrape(url) -> Advertisement:
74     soup = useScrape(url)
75     advertisement = Advertisement(url, soup.find('h3').text.strip())
76     companyTitle = soup.find('div', class_='nlp').find('td')
77     for item in companyTitle:
78         try:
79             if item.name == None:
80                 advertisement.company = textStrip(item.text)
81         except:
82             print('Company name scrape error')
83     # advertisement.company = textStrip(company)
84
85     # all items
86     sections = soup.find_all('div', class_='section')
87     advertisement.roles = listScrapper(
88         sections, 'Guitsetgeh undsen uurg'')
89     advertisement.requirements = listScrapper(
90         sections, 'Ajliin bairnii shaardlaga')
91     advertisement.additionalInfo = listScrapper(
92         sections, 'Nemelt medeeler')
93     advertisement.level = singleItemScrapper(sections, 'Busad', 'Tuvshin
94         ')
95     advertisement.type = singleItemScrapper(sections, 'Busad', 'Turul')
96     minSalary, maxSalary, isDeable = salaryScrapper(
97         singleItemScrapper(sections, 'Busad', 'Tsalin'))
98     city, district = locationScrapper(
99         singleItemScrapper(sections, 'Busad', 'Bairshil'))
100     advertisement.minSalary = minSalary
101     advertisement.maxSalary = maxSalary
102     advertisement.isDeable = isDeable
103     advertisement.city = city
104     advertisement.district = district
105     advertisement.address = singleItemScrapper(sections, '
106         ', ' ')
107     advertisement.phoneNumber = singleItemScrapper(
108         sections, 'Holboo barih', 'Utas')
109     advertisement.fax = singleItemScrapper(
110         sections, 'Holboo barih', 'Fax')
111     advertisement.adAddedDate = singleItemScrapper(
112         sections, 'Zariin hugatsaa', 'Zar niitelsen ognoo')
113     print(advertisement.additionalInfo)
114     print('SINGLE AD SCRAPING DONE!!!', url)
115
116     return advertisement

```

Код В.2: Нэг зарын өгөгдлийг цуглуулах - adScrape.py

B.1.3 Цуглуулах өгөгдлийн төрөл

```
1 class Category:
2     url = ''
3     name = ''
4     parentId = ''
5
6     def __init__(self, url, name, parentId='None') -> None:
7         self.url = url
8         self.name = name
9         self.parentId = parentId
10
11     def getUrl(self) -> str:
12         return self.url
13
14 class Advertisement:
15     category = Category
16     url = ''
17     company = ''
18     title = ''
19     # ListInfo
20     roles = ''
21     requirements = ''
22     additionalInfo = ''
23     # OtherInfo
24     city = ''
25     district = ''
26     level = ''
27     type = ''
28     minSalary = ''
29     maxSalary = ''
30     isDealable = ''
31     # ContactInfo
32     address = ''
33     phoneNumber = ''
34     fax = ''
35     adAddedDate = ''
36
37     def __init__(self, url, title) -> None:
38         self.url = url
39         self.title = title
40
41     def setCategory(self, category) -> None:
42         self.category = category
```

Код B.3: Өгөгдлийн төрөл - classTypes.py

B.1.4 BeautifulSoup scraper

```
1 from bs4 import BeautifulSoup
2 import requests
3 from urllib.error import HTTPError
```



```
4
5
6 def UseBeautifulSoup(url):
7     try:
8         response = requests.get(url)
9         response.raise_for_status()
10    except HTTPError as error:
11        print(error)
12    soup = BeautifulSoup(response.text, 'html.parser')
13    return soup
```

Код В.4: Scrape хийх функц - scrape.py