

**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: \_\_\_\_\_ Др. Б.Хуягбаатар

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.

## ГАРЧИГ

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 .....	5
2.3 Технологийн судалгаа .....	5
3. СИСТЕМИЙН ШИНЖИЛГЭЭ .....	6
3.1 Бизнесийн үйл ажиллагааны шинжилгээ .....	6
3.2 Хэрэглэгч .....	6
3.3 Функционал шаардлага .....	6
3.4 Функционал бус шаардлага .....	6
3.5 Use case диаграм .....	6
4. СИСТЕМИЙН ЗОХИОМЖ .....	7
4.1 Өгөгдлийн сангийн диаграм .....	7
4.2 Өгөгдлийн элемент .....	7
5. ХЭРЭГЖҮҮЛЭЛТ, ҮР ДҮН .....	8
5.1 Хөгжүүлсэн байдал .....	8
НОМ ЗҮЙ .....	9
ХАВСРАЛТ .....	10
А. ҮЕЧИЛСЭН ТӨЛӨВЛӨГӨӨ .....	10

В. КОДЫН ХЭРЭГЖҮҮЛЭЛТ .....	11
В.1 Өгөгдөл цуглуулалт .....	11

## ЗУРГИЙН ЖАГСААЛТ

A.1	Бакалаврын судалгааны ажлын үечилсэн төлөвлөгөө .....	10
B.1	Фолдерийн бүтэц .....	11

## ХҮСНЭГТИЙН ЖАГСААЛТ

## Кодын жагсаалт

B.1	Бүх өгөгдлийг цуглуулах - dataScrapping.py . . . . .	11
B.2	Нэг зарын өгөгдлийг цуглуулах - adScrape.py . . . . .	13
B.3	Өгөгдлийн төрөл - classTypes.py . . . . .	16
B.4	Scrape хийх функц - classTypes.py . . . . .	16



## **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 хийх

# 1. СЭДВИЙН ТАНИЛЦУУЛГА

## 1.1 Оршил

Энэхүү бакалаврын судалгааны ажлын хүрээнд "Ажил олгогчдын өгөгдлийн анализ систем дээр суурилсан чатбот" сэдвийн дагуу 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 горьлогчдын 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 горьлогчдын 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 эрэлттэй ажлын байр, 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 машин сургалт, 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]

<sup>1</sup> 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.

---

<sup>1</sup><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.

**Системийг сонгох****2.1.2****2.2 Ижил төстэй системүүд****2.3 Технологийн судалгаа**

## **3. СИСТЕМИЙН ШИНЖИЛГЭЭ**

**3.1 Бизнесийн үйл ажиллагааны шинжилгээ**

**3.2 Хэрэглэгч**

**3.3 Функционал шаардлага**

**3.4 Функционал бус шаардлага**

**3.5 Use case диаграм**

## **4. СИСТЕМИЙН ЗОХИОМЖ**

### **4.1 Өгөгдлийн сангийн диаграм**

### **4.2 Өгөгдлийн элемент**

## **5. ХЭРЭГЖҮҮЛЭЛТ, ҮР ДҮН**

### **5.1 Хөгжүүлсэн байдал**



# Bibliography

- [1] Чатбот системийн тухай  
<https://www.engati.com/blog/types-of-chatbots-and-their-applications>

# 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	1	2	3	4	5	6	7	8	9	10	11	12 Урьдчилсан хамгаалалт	13	14 Жинхэнэ хамгаалалт	Тайлбар
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 сэнгийн зохиомж															
		Чат бот хөгжүүлэлт															
6	Хэрэгжүүлэлт	Өгөгдөлд анализ хийх															
		Хайгуулах															
7	Бичиг баримт	Тайлан боловсруулах															

Тайлбар: Төслийг гэрээжүүлэх төлөвлөгөөг 7 хоногийн дотоод хугацаагаар хийж төв гэрээг бүрдэж гүйцэтгэнэ. Хийх ажил дэд хэсгийн байдал үз ажилд зарцуулах хугацааг хувиар тусгасан байна. Ажлын эхлэх өдөрхөө хугацааг хөрөндөө дэлгэцжээ байна. Ажлын гүйцэтгэлийг дүгнэж үзвэл хэсгээр хийх боломжтой байдал "7 хоног" баганыг үүсгэнэ.

Зөвшөөрсөн: Удирдагч багш ...../Б. Хужаагтар/

Боловсруулсан: Оюутин ...../Мэдээллийн технологи А. Сайнболбоо/

Оюутны ID: 1851num1762

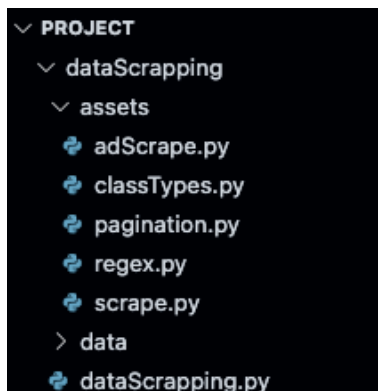
Холбогдох утас: 91990388

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

```

```

78         'Address' + '\t' +
79         'Phone' + '\t' +
80         'Fax' + '\t' +
81         'Ad Added Date' + '\n')
82 print(adUrlDict)
83 for adUrl in adUrlDict:
84     print(adUrl)
85     try:
86         tempAdItem = useAdScrape(adUrl)
87         tempAdItem.setCategory(adUrlDict[adUrl])
88         file.write(
89             tempAdItem.category.parentId+'\t' +
90             tempAdItem.category.name+'\t' +
91             tempAdItem.url+'\t' +
92             tempAdItem.company+'\t' +
93             tempAdItem.title+'\t' +
94             tempAdItem.roles+'\t' +
95             tempAdItem.requirements+'\t' +
96             tempAdItem.additionalInfo+'\t' +
97             tempAdItem.city+'\t' +
98             tempAdItem.district+'\t' +
99             tempAdItem.level+'\t' +
100            tempAdItem.type+'\t' +
101            tempAdItem.minSalary+'\t' +
102            tempAdItem.maxSalary+'\t' +
103            tempAdItem.isDealable+'\t' +
104            tempAdItem.address+'\t' +
105            tempAdItem.phoneNumber+'\t' +
106            tempAdItem.fax+'\t' +
107            tempAdItem.adAddedDate+'\n')
108         del tempAdItem
109     except:
110         print('Ad writing error')
111 file.close()
112 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 listScrapper(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, ' ')
89     advertisement.requirements = listScrapper(
90         sections, ' ')
91     advertisement.additionalInfo = listScrapper(
92         sections, ' ')
93     advertisement.level = singleItemScrapper(sections, ' ', ' ')
94     advertisement.type = singleItemScrapper(sections, ' ', ' ')
95     minSalary, maxSalary, isDealable = salaryScrapper(
96         singleItemScrapper(sections, ' ', ' '))
97     city, district = locationScrapper(
98         singleItemScrapper(sections, ' ', ' '))
99     advertisement.minSalary = minSalary
100    advertisement.maxSalary = maxSalary
101    advertisement.isDealable = isDealable
102    advertisement.city = city
103    advertisement.district = district
104    advertisement.address = singleItemScrapper(sections, ' ', ' ')
105    advertisement.phoneNumber = singleItemScrapper(
106        sections, ' ', ' ')
107    advertisement.fax = singleItemScrapper(
108        sections, ' ', ' ')
109    advertisement.adAddedDate = singleItemScrapper(
110        sections, ' ', ' ')
111    print('SINGLE AD SCRAPING DONE!!!', url)
112
113    return advertisement

```

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

**В.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
15 class Advertisement:
16     category = Category
17     url = ''
18     company = ''
19     title = ''
20     # ListInfo
21     roles = ''
22     requirements = ''
23     additionalInfo = ''
24     # OtherInfo
25     city = ''
26     district = ''
27     level = ''
28     type = ''
29     minSalary = ''
30     maxSalary = ''
31     isDealable = ''
32     # ContactInfo
33     address = ''
34     phoneNumber = ''
35     fax = ''
36     adAddedDate = ''
37
38     def __init__(self, url, title) -> None:
39         self.url = url
40         self.title = title
41
42     def setCategory(self, category) -> None:
43         self.category = category

```

Код В.3: Өгөгдлийн төрөл - classTypes.py

**В.1.4 Url хаягийн html-ийг авах функц**

```

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
15 class Advertisement:
16     category = Category
17     url = ''
18     company = ''
19     title = ''
20     # ListInfo
21     roles = ''
22     requirements = ''
23     additionalInfo = ''
24     # OtherInfo
25     city = ''
26     district = ''
27     level = ''
28     type = ''
29     minSalary = ''
30     maxSalary = ''
31     isDealable = ''
32     # ContactInfo
33     address = ''
34     phoneNumber = ''
35     fax = ''
36     adAddedDate = ''
37
38     def __init__(self, url, title) -> None:
39         self.url = url
40         self.title = title
41
42     def setCategory(self, category) -> None:
43         self.category = category
```

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