

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ  
СУРГУУЛЬ

МЭДЭЭЛЭЛ, ХОЛВООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ



Гантөмөрийн Алтай

**Хувийн үүлэн хадгалалтын системийн  
зохиомж ба хэрэгжүүлэлт**

БАКАЛАВРЫН ТӨГСӨЛТИЙН АЖИЛ

Улаанбаатар хот

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ  
СУРГУУЛЬ

МЭДЭЭЛЭЛ, ХОЛВООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ

Компьютерын ухааны салбар

Хувийн үүлэн хадгалалтын системийн  
зохиомж ба хэрэгжүүлэлт

Мэргэжлийн индекс: 07140500000002306

Мэргэжил: Программ хангамжийн инженерчлэл

Үдирдагч: Магистр (M.Sc) Э.Батцэцэг

Зөвлөгч: Доктор (Ph.D), Дэд профессор Г.Ганбат

Гүйцэтгэгч: Г.Алтай

Улаанбаатар хот

2026 он 6 сар

Батлав. Компьютерын ухааны салбарын эрхлэгч:

..... /Доктор (Ph.D), дэд профессор А.Худэр/

Удирдагч:

..... /Магистр (M.Sc) Э.Батцэцэг/

## ТӨГСӨЛТИЙН АЖЛЫН ТӨЛӨВЛӨГӨӨ

Сэдэв: "Хувийн үүлэн хадгалалтын системийн зохиомж ба  
хэрэгжүүлэлт"

№	Ажлын бүлэг, хэсгийн нэр	Эзлэх хувь	Дуусах хугацаа
1	Удиртгал хэсэг	5%	2026-2-4
2	Судалгааны хэсэг	15%	2026-2-21
3	Шинжилгээний хэсэг	15%	2026-3-4
4	Зохиомжийн хэсэг	20%	2026-4-1
5	Хөгжүүлэлтийн хэсэг	40%	2026-4-29
6	Дүгнэлт	5%	2026-5-27

Төлөвлөгөөг боловсруулсан оюутан: ..... /Г.Алтай/

## ТӨГСӨЛТИЙН АЖЛЫН ЯВЦ

№	Хийж гүйцэтгэсэн ажил	Биелсэн хугацаа	Удирдагчийн гарын үсэг
1	Удиртгал	2026-2-4	
2	Судалгааны хэсэг	2026-2-21	
3	Шинжилгээний хэсэг	2026-3-4	
4	Зохиомжийн хэсэг	2026-4-1	
5	Хөгжүүлэлтийн хэсэг	2026-4-29	
6	Дүгнэлт	2026-5-27	

Ажлын товч дүгнэлт

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Удирдагч: ..... /Магистр (M.Sc) Э.Батцэцэг/

ЗӨВШӨӨРӨЛ

Оюутан Г.Алтай-н бичсэн төгсөлтийн ажлыг УШК-д хамгаалуулахаар тодорхойлов.

Салбарын эрхлэгч: ..... /Доктор (Ph.D), дэд профессор А.Хүдэр/

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
МЭДЭЭЛЭЛ, ХОЛБООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ

**ШҮҮМЖИЙН ХУУДАС**

Компьютерын ухааны салбарын төгсөх курсийн оюутан Г.Алтайгын "Хувийн үүлэн хадгалалтын системийн зохиомж ба хэрэгжүүлэлт" сэдэвт төгсөлтийн ажлын шүүмж:

- Төслөөр дэвшүүлсэн асуудал, үүнтэй холбоотой онолын материал уншиж судалсан байдал; Энэ талаар хүмүүсийн хийсэн судалгаа, түүний үр дүнг уншиж тусгасан эсэх:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

- Төслийн ерөнхий агуулга, шийдвэр зүйлс, хүрсэн үр дүн; Өөрийн санааг гарган, харьцуулалт хийн, дүгнэж байгаа чадвар:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

- Эмх цэгцтэй, стандарт хангасан өөрөөр хэлбэл диплом бичих шаардлагуудыг биелүүлсэн эсэх; Төсөлд анзарагдсан алдаанууд, зөв бичгийн болон өгүүлбэр зүйн гэх мэт /Хуудас дугаарлагдаагүй, зураг хүснэгтийн дугаар, тайлбар байхгүй, шрифт хольсон, хувилсан зүйл ихээр оруулсан/:

.....  
.....  
.....  
.....

4. Төслөөр орхигдуулсан болон дутуу болсон зүйлс; Цаашид анхаарах хэрэгтэй зүйлс:

.....  
.....  
.....  
.....

- ## 5. Төслийн талаар онцолж тэмдэглэх зүйлс:

.....  
.....  
.....  
.....

- #### 6. Ерөнхий оноо (5 оноо)

.....

Шүүмж бичсэн: ..... /Магистр (M.Sc) О.Нэр/

Ажлын газар: .....

## Хаяг (Үтас) .....

# Зохиогч эрхийн хамгаалал

Миний бие Г.Алтай, ”Хувийн үүлэн хадгалалтын системийн зохиомж ба хэрэгжүүлэлт” сэдэвт энэ ажил нь минийх бөгөөд дараахыг нотолж байна. Үүнд:

- Горилогч энэ ажлыг тус сургуулиас боловсролын зэрэг авахаар бүхэлд нь буюу голлон хийсэн болно.
- Энэ ажлын аль нэг хэсгийг тус сургуульд эсвэл өөр байгууллагад боловсролын зэрэг, мэргэшил авахаар өмнө нь илгээсэн бол түүнийгээ тодорхой заасан болно.
- Бусад хүмүүсийн хэвлүүлсэн ажлаас зөвлөгөө авсан бол түүнийгээ үндэслэсэн болно.
- Бусад хүмүүсийн ажлаас ишлэл хийхдээ гол эх үүсвэрийг нь заасан болно.
- Миний ажилд тусалсан голлох бүх эх үүсвэрт талархаж байна.
- Ажлыг бусадтай хамтарсан бол алийг нь бусад хүмүүс хийсэн болохыг тодорхой заасан болно.

Гарын үсэг: \_\_\_\_\_

Огноо: \_\_\_\_\_

*“Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.”*

Dave Barry

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
МЭДЭЭЛЭЛ, ХОЛБООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ

## Хураангуй

Хувийн үүлэн хадгалалтын системийн зохиомж ба

хэрэгжүүлэлт

Г.Алтай

bbyyydriver@gmail.com

*Түүхүүр үсэс: Үүлэн хадгалалт, end-to-end шифрлэлт, дамжуулагч серверийн архитектур*

Энэхүү төгсөлтийн ажлын хүрээнд хэрэглэгч өөрийн өгөгдлийг бүрэн хянах боломжтой, нөөц багатай төхөөрөмж (Raspberry Pi) дээр ажиллах хувийн үүлэн хадгалалтын системийг зохиомжлон хэрэгжүүлэв. Уг систем нь гуравдагч этгээдийн хадгалалтын үйлчилгээнээс хамааралгүйгээр өгөгдлийн өмчлөл, нууцлалыг хангах зорилготой. Системийн үндсэн архитектур нь дотоод сүлжээнд серверийг автоматаар илрүүлэх болон шууд холболт боломжгүй үед дамжуулагч сервер (Relay Server) ашиглан интернэт орчинд автомат холболт тогтоох шийдлүүдээс бүрдэнэ

Хөгжүүлэлтийн явцад олон хэрэглэгчийн зэрэг холболтыг дэмжих зэрэгцээ боловсруулалт бүхий дамжуулагч серверийн архитектурыг зохиож , хэрэглэгчийн гар утасны (IOS) программыг боловсруулсан. Аюулгүй байдлыг хангах үүднээс бүх өгөгдөл дамжуулалтад End-to-End шифрлэлт болон QR кодод суурилсан холболтын механизмыг ашигласан болно.

# Талархал

Энэхүү дипломын ажлыг хийж дуусгахад хамгийн гол нөлөө үзүүлсэн, сэдэв бүрийг нягталж, үнэтэй зөвлөгөө өгч, алдааг маань засаж, удирдан чиглүүлж байсан удирдагч багш Э.Батцэцэг багшдаа хамгийн түрүүнд чин сэтгэлээсээ баярлалаа гэж хэлмээр байна. Мөн дипломын ажлын зөвлөх багш болох Г.Ганбат багшдаа үнэт цаагаа зарцуулж, зааж, сургаж, орхигдуулсан зүйлсийг сануулж, хэрэгтэй зөвлөгөөнүүд өгч байсанд гүн талархлаа илэрхийлье.

Намайг бакалаврын дипломын ажил бичиж чадах хэмжээнд хүртэл зааж сургасан МХТС -н Компьютерын Ухааны салбарын багш нар, “Программ хангамж” мэргэжлийн хөтөлбөрийн хүрээнд хичээл зааж байсан ШУТИС -н багш нартаа баярлалаа. Суралцахад аюулгүй, цэвэрхэн орчныг бүрдүүлж өгч байсан МХТС -н ажилчдад баярлалаа.

Академик мэдлэгээ практик дээр туршин, дипломын ажлын хүрээнд шинэ технологиудыг судлан, мэргэжлийн хувьд туршлагажих боломж олгож байсан Грэйпсити Монгол компанийн “Технологийн Газар” -н захирал болон ахлах инженертээ талархлаа илэрхийлмээр байна.

Эцэст нь, би дипломын ажилдаа их цаг зарцуулах боломжийг олгож, олон талаар чухал дэмжлэг үзүүлж байсан гэр бүлийнхэн болон урам зориг, эрч хүчээр урамшуулж байсан найзууддаа ч бас чин сэтгэлийн талархлаа илэрхийлэх ёстой.

# Товчилсон ҮГС

<b>API</b>	Application Programming Interface
<b>CPU</b>	Central Processing Unit
<b>OS</b>	Operating System
<b>IDE</b>	Integrated Development Environment
<b>SQL</b>	Structured Query Language
<b>HTML</b>	Hypertext Markup Language
<b>XML</b>	Extensible Markup Language
<b>JSON</b>	Javascript Object Notation

# Гарчиг

<b>Зохиогч эрхийн хамгаалал</b>	<b>i</b>
<b>Хураангуй</b>	<b>iii</b>
<b>Талархал</b>	<b>iv</b>
<b>Товчилсон үгс</b>	<b>v</b>
<b>Удиртгал</b>	<b>1</b>
<b>1 ... системийн тухай онол, арга зүйн судалгаа</b>	<b>4</b>
1.1 ... тухай . . . . .	4
1.2 ... бий болгосон эхэн үеийн алгоритмууд . . . . .	4
1.2.1 Виола Жонес илрүүлэлт . . . . .	4
1.2.2 HOG /Histogram of Oriented Gradients/ илрүүлэлт . . . . .	5
1.2.3 DPM /Deformable Part-based Model/ илрүүлэлт . . . . .	8
1.3 Объект илрүүлэлтэд ашиглаж буй сүүлийн үеийн алгоритмууд . . . . .	8
1.3.1 RCNN, Fast RCNN, Faster RCNN илрүүлэлт . . . . .	8
1.3.2 SSD /Single Shot Detector/ илрүүлэлт . . . . .	11
1.3.3 YOLO /You Only Look Once/ илрүүлэлт . . . . .	12
1.4 Ижил төстэй системийн судалгаа . . . . .	13
1.5 Хөгжүүлэлтэд ашиглах алгоритм . . . . .	15
1.5.1 Объект илрүүлэлтэд ашиглах алгоритм . . . . .	15
1.6 Технологийн судалгаа . . . . .	20
1.6.1 Back-End хөгжүүлэлтэд ашиглах технологиуд . . . . .	20
OpenCV зургийн сан . . . . .	20
FAST API . . . . .	21
Python gTTS сан . . . . .	21
Chimege API . . . . .	21
MySQL өгөгдлийн сан удирдах систем . . . . .	22
PyCharm IDE . . . . .	22
1.6.2 Front-End хөгжүүлэлтэнд ашиглах технологиуд . . . . .	23
React фреймворк . . . . .	23
Ant design . . . . .	24
Visual Studio Code . . . . .	25
1.6.3 Кодын менежмент . . . . .	25
1.7 Бүлгийн дүгнэлт . . . . .	25

<b>2 ... системийн шинжилгээ</b>	<b>27</b>
2.1 Системийн үйл ажиллагааны тухай дэлгэрэнгүй . . . . .	27
2.2 Системийг ашиглах хэрэглэгчид . . . . .	27
2.3 Функцийн шаардлага . . . . .	27
2.4 Функцийн бус шаардлага . . . . .	28
2.5 Юзкейс диаграмм . . . . .	29
2.6 Юзкейсийн тодорхойлолт . . . . .	29
2.7 Шинжилгээний класс диаграмм . . . . .	32
2.8 Шинжилгээний дарааллын диаграмм . . . . .	32
2.9 Үйл ажиллагааны диаграмм . . . . .	34
2.10 Бүлгийн дүгнэлт . . . . .	36
<b>3 ... системийн зохиомж ба хөгжүүлэлт</b>	<b>37</b>
3.1 Системийн архитектур . . . . .	37
3.2 Зохиомжийн шатны класс диаграмм . . . . .	37
3.3 Өгөгдлийн ерөнхий схем . . . . .	38
3.4 Системийн прототип . . . . .	38
3.5 Хөгжүүлсэн системийн интерфейс . . . . .	41
3.6 Системд хийгдсэн тест . . . . .	42
3.7 Системийн нэвтрүүлэлт . . . . .	42
3.8 Бүлгийн дүгнэлт . . . . .	42
<b>Ерөнхий дүгнэлт</b>	<b>44</b>

# Зургийн жагсаалт

1.1	Зураг танилт болон объект илрүүлэлтийн ялгаа . . . . .	4
1.2	HAAR төст онцлогууд . . . . .	5
1.3	HAAR ашигласан хүний царай илрүүлэлт . . . . .	5
1.4	Саарал зургийн дурслэл . . . . .	5
1.5	HOG аргаар объект илрүүлэх алхмууд . . . . .	6
1.6	Урьдчилсан боловсруулалтын үе шат . . . . .	6
1.7	Градиент тооцоолох үе шат . . . . .	7
1.8	Магнитуд болон чиглэл тодорхойлох . . . . .	7
1.9	Вектор үүсгэх . . . . .	7
1.10	HOG -н үр дүн . . . . .	8
1.11	DPM илрүүлэлт . . . . .	8
1.12	RCNN -н процесс . . . . .	9
1.13	Сонгомол хайлтын алгоритмын дурслэл . . . . .	9
1.14	Хувиргах үйлдэл . . . . .	10
1.15	CNN -р объектуудыг өргөжүүлэх процесс . . . . .	10
1.16	Ангилах процесс . . . . .	10
1.17	Fast RCNN -н процесс . . . . .	10
1.18	Fastest RCNN -н процесс . . . . .	11
1.19	SSD -н процесс . . . . .	11
1.20	Зургийг блокуудад хуваах . . . . .	12
1.21	Объектыг хүрээлэх хүрээ . . . . .	12
1.22	Объектуудыг хүрээлсэн хүрээ огтлондох . . . . .	13
1.23	Iashin.ai веб сайтын нүүр хуудас . . . . .	13
1.24	Iashin.ai веб сайт зурганд объект илрүүлэлт хийсэн нь . . . . .	14
1.25	TapTapSee апликашны лого . . . . .	14
1.26	TapTapSee апликашн гар утсан дээр ажиллах байдал . . . . .	15
1.27	MobileNet - SSD V3 архитектур . . . . .	16
1.28	Filter - хийж буй процесс . . . . .	17
1.29	Filter - н төрлүүд . . . . .	17
1.30	Таамаглалыг сонгон авах . . . . .	18
1.31	Нэг цэг дээрх таамаглалууд . . . . .	18
1.32	Объектыг хүрээлсэн хүрээний урт, өргөнийг тооцоолох томьёо . . . . .	18
1.33	Таамаглалын multiBox . . . . .	18
1.34	MobileNet SSD архитектурын дундаж нарийвчлал . . . . .	19
1.35	OpenCV лого . . . . .	20
1.36	FAST API лого . . . . .	21

1.37 Google translate болон Python лого . . . . .	21
1.38 Чимэгэ Системс лого . . . . .	22
1.39 MySQL лого . . . . .	22
1.40 PyCharm IDE лого . . . . .	23
1.41 React лого . . . . .	24
1.42 Ant design лого . . . . .	24
1.43 Visual Studio Code лого . . . . .	25
 2.1 Юзкейс диаграмм . . . . .	29
2.2 Шинжилгээний класс диаграмм . . . . .	32
2.3 Хэрэглэгчийн бүртгэлийн мэдээлэл зохион байгуулах шинжилгээний дарааллын диаграмм . . . . .	32
2.4 Бүртгэл үүсгэх шинжилгээний дарааллын диаграмм . . . . .	33
2.5 Хадгалагдсан файлуудын мэдээллийг зохион байгуулах шинжилгээний дарааллын диаграмм . . . . .	33
2.6 Файл системд хуулах шинжилгээний дарааллын диаграмм . . . . .	33
2.7 Объект илрүүлэлт хийх шинжилгээний дарааллын диаграмм . . . . .	34
2.8 Хэрэглэгчийн бүртгэлийн мэдээлэл зохион байгуулах үйл ажиллагааны диаграмм . . . . .	34
2.9 Бүртгэл үүсгэх үйл ажиллагааны диаграмм . . . . .	35
2.10 Хадгалагдсан файлуудын мэдээллийг зохион байгуулах үйл ажиллагааны диаграмм . . . . .	35
2.11 Файл системд хуулах үйл ажиллагааны диаграмм . . . . .	35
2.12 Объект илрүүлэлт хийх үйл ажиллагааны диаграмм . . . . .	36
 3.1 Системийн архитектур . . . . .	37
3.2 Класс диаграмм . . . . .	38
3.3 Өгөгдлийн ерөнхий схем . . . . .	38
3.4 Зургаас объект илрүүлэлт хийх хуудасны прототип . . . . .	39
3.5 Бүртгэл үүсгэх хуудасны прототип . . . . .	39
3.6 Нэвтрэх хуудасны прототип . . . . .	39
3.7 Бүртгэлтэй файлын мэдээлэл харах хуудасны прототип . . . . .	39
3.8 Хэрэглэгчийн сонголт харуулах дэлгэцийн прототип . . . . .	39
3.9 Бүртгэлийн мэдээлэл шинэчлэх хуудасны прототип . . . . .	40
3.10 Бүртгэлийн мэдээлэл амжилттай шинэчлэгдсэнийг харуулах дэлгэцийн прототип . . . . .	40
3.11 OTP оруулах хуудасны прототип . . . . .	40
3.12 Бүртгэл амжилттай үүссэнийг харуулах дэлгэцийн прототип . . . . .	40
3.13 Код дахин илгээх эсэхийг лавлах дэлгэцийн прототип . . . . .	40
3.14 Объект илрүүлэлт хийх хуудасны интерфейс . . . . .	41
3.15 Нэвтрэх хуудасны интерфейс . . . . .	41
3.16 Бүртгэл үүсгэх хуудасны интерфейс . . . . .	41
3.17 OTP шалгах хуудасны интерфейс . . . . .	42
3.18 Хэрэглэгчийн бүртгэлээр орсон үеийн объект илрүүлэлт хийх хуудас . . . . .	42
3.19 Бүртгэлтэй зургуудын жагсаалт харуулах интерфейс . . . . .	42

# ХҮСНЭГТИЙН ЖАГСААЛТ

1.1	RCNN архитектуруудын харьцуулалт . . . . .	11
1.2	Судлагдсан системүүдийн харьцуулалт . . . . .	15
1.3	MobileNet SSD архитектурын үзүүлэлтүүд . . . . .	19
2.1	Системд файл хуулах юзкейсийн тодорхойлолт . . . . .	29
2.2	Объект илрүүлэлт хийх юзкейсийн тодорхойлолт . . . . .	30

## Үдиртгал

Өдгөө энгийн ухаалаг утас хэрэглэгч бүр үүлэн хадгалалт буюу Cloud Storage-ийг хэрэглэж байна. Хүн бүрт зураг, бичлэг, ажлын болон хувийн чухал файлууд гэх мэт өгөгдлүүдийг хадгалах хэрэгцээ гардаг бөгөөд түүнийгээ интернетийн тусламжтай хаанаас ч ашиглах боломж нь орчин үеийн мэдээллийн эрин үед амьдрахтай нь салшгүй нэг хэсэг болсон билээ. Гэвч энэхүү зах зээлийн ихэнх хувийг төвлөрсөн үүлэн хадгалалтын үйлчилгээ үзүүлдэг компаниуд удирдаж байгаа бөгөөд хэрэглэгчдийн өгөгдөл тухайн компанийн сервер дээр хадгалагддаг. Энэхүү хадгалалтын үйлчилгээ нь энгийнээр түрээсийн үйлчилгээ юм. Иймд хэрэглэгч өөрийн техник хангамжийг ашиглан өгөгдлөө хадгалах замаар түрээсийн төлбөр болон багтаамжийн хязгаарлалтаас ангид байх, үүний зэрэгцээ одоогийн үүлэн үйлчилгээнүүдтэй ижил түвшний хэрэглээний сэтгэл ханамжийг өгч чадах системийг хөгжүүлэх нь энэхүү ажлын гол зорилго юм.

## Зорилго

Энэхүү төслийн зорилго нь хэрэглэгч өөрийн эзэмшлийн техник хангамжид суурилсан, гуравдагч этгээдээс хамааралгүй, бие даасан хувийн үүлэн хадгалалтын системийг хөгжүүлэхэд оршино. Тус систем нь өгөгдлийн өмчлөл болон нууцлалыг бүрэн хангахын зэрэгцээ дотоод болон гадаад сүлжээний орчинд төхөөрөмжүүд хооронд өгөгдлийг саадгүй, аюулгүй дамжуулах нэгдсэн шийдлийг бүрдүүлнэ.

## Төслийн зорилтууд

Дэвшүүлсэн зорилгод хүрэхийн тулд дараах тодорхой зорилтуудыг хэрэгжүүлнэ.  
Үүнд:

- Хувийн үүлэн хадгалалтын ижил төстэй системүүдийн архитектур болон технологийн харьцуулсан судалгаа хийх;
- Нөөц багатай төхөөрөмж (Raspberry Pi г.м)-д зориулсан үүлэн хадгалалтын серверийн программ хангамжийг зохион бүтээх;
- Файл удирдах үндсэн үйлдлүүд (upload, download, organize)-ийг дэмжих iOS үйлдлийн системд зориулсан хэрэглэгчийн аппликашн боловсруулах;
- Дотоод сүлжээнд өгөгдлийг хамгийн бага хоногдолтой дамжуулах "LAN-first" механизмыг нэвтрүүлэх;

5. Интернэт орчинд олон хэрэглэгч зэрэг хандах боломжтой, зэрэгцээ боловсруулалт (parallel computing) бүхий дамжуулагч серверийн (relay server) архитектурыг шийдвэрлэх;
6. Өгөгдлийн нууцлал болон аюулгүй байдлыг хангах үүднээс төгсгөлийн цэг хоорондын (End-to-End) шифрлэлтийг хэрэгжүүлэх.

## Судлах үндэслэл

Өнөөгийн дижитал шилжилтийн эрин үед үүлэн хадгалалт (Cloud Storage) нь хэрэглэгчдийн өдөр тутмын хэрэгцээ болж, 2025 оны байдлаар дэлхий даяар ашиглагчдын тоо 2.3 тэрбумыг давсан үзүүлэлттэй байна. Гэвч энэхүү өсөлтийг дагаад дараах хоёр үндсэн асуудал тулгарч байна:

- 1. Өгөгдлийн нууцлал ба хараат байдал:** Зах зээлийн дийлэнх хувийг Google, Apple, Microsoft зэрэг томоохон компаниуд эзэлж байгаа нь хэрэглэгчдийн хувийн мэдээлэл гуравдагч этгээдийн серверт төвлөрөхөд хүргэж байна. Энэ нь өгөгдөл эзэмших эрх (data ownership) болон нууцлалын аюулгүй байдалд эрсдэл дагуулдаг бөгөөд хэрэглэгч өөрийн датаг хянах боломжийг хязгаарладаг.
- 2. Эдийн засгийн зардал ба хязгаарлагдмал нөөц:** Төвлөрсөн үүлэн үйлчилгээнүүд нь хадгалах багтаамжаас хамаарсан тогтмол (сарын эсвэл жилийн) төлбөр шаарддаг. Тухайлбал, Google One үйлчилгээний 100GB-aас 2TB хүртэлх багтаамж нь жилийн 20-100 долларын хооронд хэлбэлзэж байна. Их хэмжээний өгөгдөл хадгалах шаардлагатай хэрэглэгчдийн хувьд энэ нь урт хугацаандаа эдийн засгийн дарамт болдог.

Иймд хэрэглэгч өөрт байгаа техник хангамжийн нөөц боломжийг ашиглан, ямар нэгэн нэмэлт төлбөргүйгээр өөрийн өгөгдлийг бүрэн хянах, аюулгүй хадгалах боломжийг бурдүүлсэн хувийн үүлэн системийг хөгжүүлэх нь технологийн болон практик ач холбогдолтой юм.

## Судлагдсан байдал

Хувийн үүлэн хадгалалтын системийн чиглэлээр дэлхий дахинд Nextcloud, Seafile, Syncthing зэрэг нээлттэй эхийн төслүүд өргөнөөр судлагдаж, хөгжүүлэгдсээр ирсэн. Эдгээр системүүд нь хэрэглэгчийн өгөгдлийн нууцлалыг хангах үндсэн зорилготой боловч Raspberry Pi зэрэг нөөц багатай төхөөрөмж дээр ажиллахад системийн ачаалал ихсэх, тохиргоо хийхэд хэт нарийн мэргэжлийн мэдлэг шаарддаг зэрэг дутагдалтай талууд ажиглагддаг.

Сүлжээний хандалтын хувьд NAT traversal болон Reverse Proxy-д суурилсан шийдлүүдийг ашигладаг боловч олон хэрэглэгч зэрэг хандах үед дамжуулах хурд удаашрах асуудал тулгардаг. Иймд энэхүү төслөөр бага чадалтай төхөөрөмжид оновчтой (lightweight) ажиллах серверийн бүтэц болон зэрэгцээ боловсруулалт бүхий дамжуулагч серверийн архитектурыг ашиглан дээрх асуудлуудыг шийдвэрлэхээр зорьсон.

## Шинэлэг тал

Хувийн үүлэн хадгалалтын системийн шинэлэг тал:

- Өгөгдлийн бүрэн эзэмшил (Data Governance):** Хэрэглэгчийн өгөгдөл гуравдагч талын серверт бус, зөвхөн хэрэглэгчийн өөрийн техник хангамжид хадгалагдаж, нууцлал нь бүрэн хяналтад байна.
- Бага нөөцөд зориулсан шийдэл (Resource-Efficient):** Raspberry Pi болон хуучин компьютер зэрэг бага чадалтай төхөөрөмжүүдэд зориулсан хөнгөн (lightweight) архитектуртай.
- Хялбар тохиргоо (Zero-Config Networking):** Дамжуулагч серверийн тусламжтайгаар сүлжээний нарийн тохиргоо (Port forwarding, Static IP) шаардахгүйгээр шууд ашиглах боломжтой.

## Технологийн ач холбогдол

Энэхүү систем нь технологийн хувьд дараах ач холбогдолтой:

- Сүлжээний бие даасан байдал:** Дамжуулагч серверийн шийдэл нь интернэтийн дурын орчноос (Public IP-гүй байсан ч) саадгүй холбогдох боломжийг олгоно.
- Зардал хэмнэлт:** Үүлэн хадгалалтын тогтмол төлбөрийг халж, байгаа нөөцөө ашиглан эдийн засгийн хэмнэлт гаргана.
- Аюулгүй байдлын шинэ стандарт:** End-to-End шифрлэлтийг ашигласнаар дамжуулагч сервер хүртэл өгөгдлийг унших боломжгүй болж, нууцлалын өндөр түвшинд хүрнэ.

## Хамрах хүрээ

Тус төсөл нь дараах хүрээг хамарна:

- Өрийн өгөгдлийн нууцлалыг эрхэмлэдэг хувь хүн болон гэр бүлийн хэрэглэгчид;
- Өндөр өртөг бүхий үүлэн үйлчилгээнээс татгалзаж, өөрийн нөөцийг ашиглах хүсэлтэй технологи сонирхогчид;

---

---

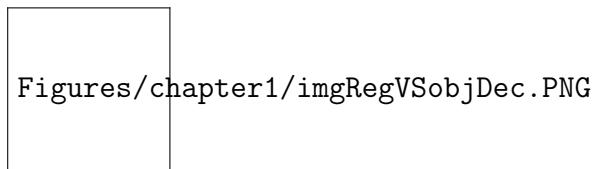
## БҮЛЭГ 1

---

... системийн тухай онол, арга зүйн  
судалгаа

## 1.1 ... тухай

Компьютерын хараа /Computer Vision/ бол визуал дата /Visual Data/ буюу зураг болон видео файлтай ажиллаж, тэдгээр дээр боловсруулалт, анализ хийн, компьютерыг зураг, видеог хүн шиг ойлгодог болгох тал дээр төвлөрдөг компьютерын шинжлэх ухааны салбар шинжлэх ухаан. Харин объект илрүүлэлт /Object Detection/ нь зураг эсвэл видео доторх обьектуудыг олох, танихад чиглэгддэг компьютерын харааны техник юм. Тодруулбал, объект илрүүлэлт нь эдгээр илрүүлсэн обьектуудаа хүрээлсэн хайрцаг зурдаг бөгөөд энэ нь тухайн обьектууд нь яг юу болохыг тодорхойлж, тухайн үзэгдэлд хаана байгааг (эсвэл тэд хэрхэн хөдөлж байгааг) олох боломжийг олгодог [**medeeStatistic**]. Объект илрүүлэлт нь ихэвчлэн зургийн танилт /Image Recognition/ -тай андуурагддаг. Тиймээс объект илрүүлэлтийг цааш судлахаасаа өмнө тэдгээрийн хоорондын ялгааг тодруулах нь чухал юм. Зураг 1.1 -д зураг танилт болон объект илрүүлэлтийн ялгааг харуулсан зургийг харуулсан.



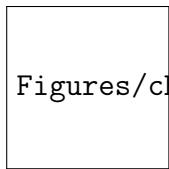
ЗУРАГ 1.1: Зураг танилт болон объект илрүүлэлтийн ялгаа

## 1.2 ... бий болгосон эхэн үеийн алгоритмууд

Объект илрүүлэлтийн эрин үеийг үндсэн хоёр үед хуваан үзэж болно. Эхний үе нь “Уламжлалт объект илрүүлэлтийн үе” буюу 2014 оноос өмнөх үе, дараагийн үе нь “Гүний сургалт дээр суурилсан объект илрүүлэлтийн үе” буюу 2014 оноос хойших онууд [**Reference1**].

### 1.2.1 Виола Жонес илрүүлэлт

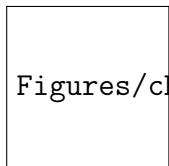
Паул Виола болон Михаел Жонес нарын 2001 онд хөгжүүлсэн хүний нүүр царайг илрүүлэх арга. Энэ нь HAAR -төст онцлогуудыг зураг дээрээс олсноор хүний нүүр царайг таньдаг. Зураг 1.2 -д Виола Жонес илрүүлэлтэд ашигладаг HAAR төст онцлогуудыг дүрсэлсэн.



Figures/chapter1/edge&lineFeatures.PNG

ЗУРАГ 1.2: HAAR төст онцлогууд

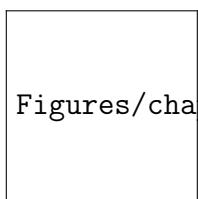
Зураг 1.3 -д хүний царайг дүрсэлсэн саарал зургаас HAAR -төст онцлогуудыг олсныг харуулж байна.



Figures/chapter1/featuresOnFace.PNG

ЗУРАГ 1.3: HAAR ашигласан хүний царай илрүүлэлт

HAAR -төст онцлогууд буюу дараах HAAR гэж нэрлэгддэг цонхнуудыг (Анх Алфред Хаар гэх хүн энэ санааг дэвшүүлсэн учраас түүний нэрээр нэрлэсэн) саарал зураг/Grayscale Image/ дээгүүр гүйлгэж тухайн зургийн HAAR -тай төстэй хэсгийг онцлон авч хүний нүд, хөмсөг, уруул, хамрын хэсэг гэж үздэг байсан. Саарал зургийн хувьд пикселүүд нь зөвхөн хар болон цагаанаар ялгагдаж харагддаггүй учраас HAAR -тай төстэй хэсэг байна гэдгийг батлахын тулд дараах тооцооллыг хийдэг. Зураг дээгүүр гүйлгэж буй цонх буюу кернелийн 0,5 с доош утгатай байгаа пикселүүдийг нэмнэ. Мөн адил 0,5 с дээш утгатай байгаа пикселүүдийг хооронд нь нэмээд дараа нь тэдгээрийн зөрүүг олно. Зөрүү утга 1-тэй дөхөх тусам HAAR төст хэсэг байна гэж үзэн 1-тэй хамгийн ойр HAAR төст хэсгийг авна. Зураг 1.4 -д саарал зургийн пикселийг 0-1 хооронд дүрсэлснийг харуулж байна.



Figures/chapter1/featureDetector.PNG

ЗУРАГ 1.4: Саарал зургийн дурслэл

### 1.2.2 HOG /Histogram of Oriented Gradients/ илрүүлэлт

Чиглэгдсэн градиентийг илрүүлэх гистограмм арга (HOG) нь бараг арван жилийн настай эртний алгоритмуудын нэг, гэсэн хэдий ч бусдаас ялгараах нэг зүйл бол өнөөг

хуртэл маш их ашиглагдаж байгаа бөгөөд үр дүнтэй [introHOG]. Чиглэгдсэн градиентийн гистограмм аргыг зургийг ангилахын тулд нүүр царай, дурс илрүүлэхэд ашигладаг. Навнеет Далал, Билл Тригс нар 2005 онд чиглэгдсэн градиентийн гистограмм (HOG) функцийг нэвтрүүлсэн. Чиглэгдсэн градиент тодорхойлогчийн гистограммын цаадах зарчим нь зураг доторх объектын дүр төрх, хэлбэрийг эрчмийн градиент эсвэл ирмэгийн чиглэлийн хуваарилалтаар тодорхойлж болно [Reference2]. Зургийн х ба у градиентууд нь эрчимжилтийн огцом өөрчлөлтийн улмаас ирмэг ба булангийн эргэн тойронд градиентийн хэмжээ их байдаг тул ирмэг ба булангууд нь хавтгай хэсгүүдээс илүү объектын хэлбэрийн талаар илүү их мэдээлэл агуулдаг. Тиймээс градиентийн чиглэлийн гистограммыг энэ тодорхойлогчийн шинж чанар болгон ашигладаг. Зураг 1.5 -д HOG аргаар объект илрүүлэх алхмуудын дарааллыг харуулж байна.

Figures/chapter1/stephog.PNG

ЗУРАГ 1.5: HOG аргаар объект илрүүлэх алхмууд

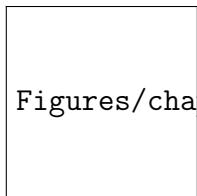
HOG аргаар объект илрүүлэх алхмууд:

- **Урьдчилан боловсруулах:** Зургийг урьдчилан боловсруулахын тулд нийтлэг харьцаа буюу (өргөн: өндөр) 1: 2 харьцаа руу зургийг хувиргах хэрэгтэй. 100x200, 500x1000 гэх мэт байж болно. Зураг 1.6 -д урьдчилсан боловсруулалтын үе шатанд явж буй хувиргагдсан харьцаатай зургуудыг дүрсэлж байна.

Figures/chapter1/preprocessinghog.PNG

ЗУРАГ 1.6: Урьдчилсан боловсруулалтын үе шат

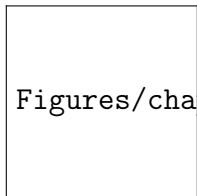
- **Градиент тооцоолох:** X болон Y чиглэлд пиксел тус бүрд зургийн градиентийг тооцоолно. Зураг 1.7 нь градиент тооцоолох томъёог харуулж байна.



Figures/chapter1/calculateGradient.PNG

ЗУРАГ 1.7: Градиент тооцоолох үе шат

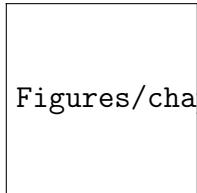
- **Градиентуудын гистограмм тооцоолох:** Энэ үйлдлийг ойлгомжтойгоор тайлбарлавал, зургийг хуваасан нүднүүдээ нэгтгэж  $8 \times 8$  бол нийт 64 магнитуди үүсгээд тэдгээрийгээ өнцгөөс нь хамааруулан 9 ширхэг саванд хувааж хийнэ гэсэн үг. Зураг 1.8 -д зургийн магнитудыг олсныг харуулж байна.



Figures/chapter1/gradientmagnitudeandirection.PNG

ЗУРАГ 1.8: Магнитуд болон чиглэл тодорхойлох

Зураг 1.9 -д градиент чиглэл болон магнитуудыг ашиглан градиентуудын гистограмм үүсгэж байна.

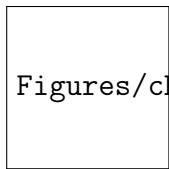


Figures/chapter1/creativector.PNG

ЗУРАГ 1.9: Вектор үүсгэх

- **Блок нормалчлал:** Гэрэлтүүлгийн өөрчлөлт гистограммын хэмжээнд нөлөөлдөг. Тиймээс гистограммын утгыг гэрэлтүүлгээс хамааралгүй байлгахын тулд өмнөх алхамд үүсгэсэн гистограмм векторыг нормалчилдаг.
- **Вектор үүсгэх:** Үүсгэсэн векторын тодорхойлогч векторыг олно.

Зураг 1.10 нь зургийн градиентуудыг дүрсэлсэн. Энэхүү дүрслэл нь градиентууд хаашаа шилжиж байгааг ойлгох, зургийн дотор объектууд хаана байгааг мэдэхэд хэрэгтэй байж болно.

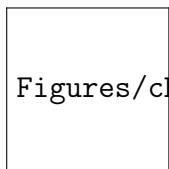


Figures/chapter1/visualhog.PNG

ЗУРАГ 1.10: HOG -н үр дүн

### 1.2.3 DPM /Deformable Part-based Model/ илрүүлэлт

Хэлбэрийг нь алдагдуулж болох хэсгүүдэд суурилсан загвар. Энэ нь 1973 оны Фишлер болон Элшлагер нарын "объектууд нь гажигтай хэсгүүдийн цуглуулга юм" гэсэн санаан дээрээс үүдэлтэй загвар [introdDPM]. Хөгжүүлэлт нь HOG загвараас санаа авснаар эхэлсэн. DPM нь зургийг пирамид загвар руу шилжүүлэн, давхарга бүрд нь HOG аргыг ашиглан онцлог шинжийг олж дараах томьёогоор хамгийн өндөр оноотой байршилыг тогтоон илрүүлдэг. Зураг 1.11 -д зургийг пирамид загвар руу шилжүүлж илрүүлэлт хийх процессыг харуулж байна.



Figures/chapter1/dpm.PNG

ЗУРАГ 1.11: DPM илрүүлэлт

## 1.3 Объект илрүүлэлтэд ашиглаж буй сүүлийн үеийн алгоритмууд

Объект илрүүлэлтийн сүүлийн үеийн гол архитектур болох CNN дээр тулгуурлан хөгжүүлэгдсэн RCNN -н архитектурууд, YOLO болон SSD илрүүлэлтийн тухай 1.3 -н дэд бүлгүүдэд авч үзнэ.

### 1.3.1 RCNN, Fast RCNN, Faster RCNN илрүүлэлт

**RCNN /Region-based Convolutional Neural Network/** нь бүсчлэлд суурилсан эргэлдсэн нейроны служээ буюу компьютерын хараа болон дүрс боловсруулахад ашигладаг машин сургалтын загваруудын нэг. R-CNN-ийн анхны зорилго нь объект илрүүлэхэд зориулагдсан бөгөөд тэдгээрийн эргэн тойрон дахь хил хязгаарыг тодорхойлсон оролтын дүрс дэх объектуудыг илрүүлэх явдал юм [introdRCNN]. R-CNN

загварт өгөгдсөн оролтын зураг нь сонгомол хайлт хэмээх механизмаар дамждаг бөгөөд тухайн бүсчлэлийн талаарх мэдээллийг гаргаж авдаг. Сонирхсон бүсийг тэгш өнцөгтийн хилээр төлөөлж болно. Хувилбараас хамааран 2000 гаруй сонирхлын бүс байж болно. Энэхүү ашиг сонирхлын бүс нь CNN-ээр дамжуулж гаралтын шинж чанарыг бий болгодог. Дараа нь эдгээр гаралтын функцууд нь SVM (Support Vector Machine/дэмжих вектор машин) ангилагчаар дамждаг бөгөөд тухайн объектыг сонирхж буй бүсэд ангилдаг. Зураг 1.12 -д RCNN илрүүлэлтийн процессыг харуулж байна.

Figures/chapter1/wholeprocessRCNN.PNG

ЗУРАГ 1.12: RCNN -н процесс

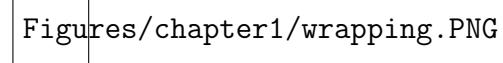
Зурган дээр өөр өөр хэмжээтэй гүйдэг шүүлтүүрийг ашиглан зургаас обьектыг гаргаж авах нь бүрэн хайлтын арга гэж нэрлэгддэг. Сонгомол хайлтын алгоритм нь бүрэн хайлтыг дангаар нь ашиглахын оронд зураг дээр үзүүлсэн өнгөний сегментчилэлтэй ажилладаг. Сонгомол хайлт нь обьектод өөр өөр өнгө өгөх замаар объектыг дурсээс салгах арга гэж хэлж болно. Энэ алгоритм нь олон жижиг цонх эсвэл шүүлтүүр хийхээс эхэлж, сонирхлын бүсчлэлийг тодорхойлохын тулд greedy алгоритмыг ашигладаг. Дараа нь бүсчлэлд ижил төстэй өнгийг олж, тэдгээрийг нэгтгэдэг. Бүсчлэлийн ижил төстэй байдлыг дараах байдлаар тооцоолж болно.  $S(a,b)=\text{Бүтэц}(a,b)+\text{Хэмжээ}(a,b)$  Бүтэц( $a,b$ ) нь бүс нутгийн хоорондох харагдах байдал ба Хэмжээ( $a,b$ ) ижил төстэй байдал юм. Зураг 1.13 -д сонгомол хайлтын арга болох өнгөний сегментчилэлийг ашигласан зургийг харуулж байна.

Figures/chapter1/selectivealgorithm.PNG

ЗУРАГ 1.13: Сонгомол хайлтын алгоритмын дүрслэл

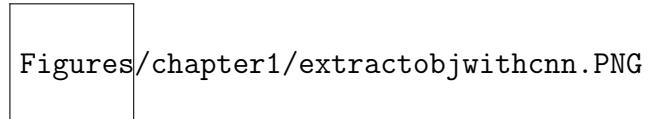
Энэ үйлдлийн дараа эдгээр үйлдлүүд хийгдэнэ.

**1.Хувиргах** Бүсчлэлийг сонгосны дараа бүсчлэлүүдийг агуулсан зураг нь CNN-ээр дамждаг бөгөөд CNN загвар нь тухайн бүсээс обьектуудыг гаргаж авдаг. Зургийн хэмжээг CNN-ийн багтаамжийн дагуу тохируулах ёстой тул дүрсийг өөрчлөхөд хэсэг хугацаа эсвэл ихэнх цаг хугацаа шаардагдана. Үндсэн R-CNN-д бүсчлэлийг 227 x 227 x 3 хэмжээтэй зураг болгон хуваадаг. Зураг 1.14 -д хувиргах үйлдлийг харуулж байна.



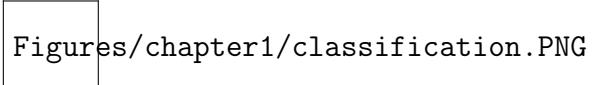
ЗУРАГ 1.14: Хувиргах үйлдэл

**2.CNN -р объектуудыг өргөжүүлэх** Объектыг задлахын тулд CNN-ийн эргэлдсэн оролтыг боловсруулна. Зураг 1.15 -с CNN -р объектуудыг өргөжүүлэх процессыг харж болно.



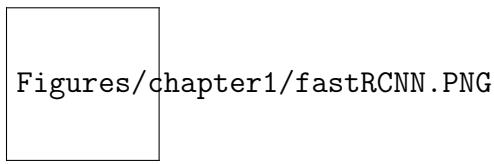
ЗУРАГ 1.15: CNN -р объектуудыг өргөжүүлэх процесс

**3.Ангилах** Үндсэн R-CNN нь өөр өөр объектуудыг ангилаалдаа тусгаарлах SVM ангилагчаас бүрддэг. Зураг 1.16 -д ангилах процесс хэрхэн явагддагийг харуулж байна.



ЗУРАГ 1.16: Ангилах процесс

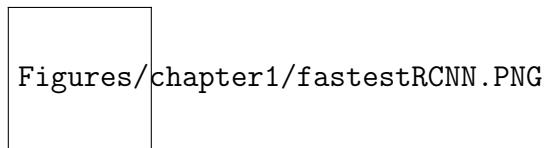
**Fast RCNN /Region-based Convolutional Neural Network/** нь RCNN -ээс илүү хурдан ажилладаг архитектур. Учир нь RCNN шиг бүсчлэл болгонд CNN -г хэрэгжүүлэхийн оронд эхлээд оролтын зурагтаа CNN -г бүхэлд нь хэрэгжүүлж дараа нь "сонирхлын бүсчлэл" нэртэй давхаргыг үүсгэн тухайн давхаргаасаа бүсчлэлүүдээ гаргаж авна, дараа нь бүрэн холбогдсон давхаргаар дамжуулж ажиллана. Объектыг хүрээлсэн хүрээг тодорхойлохдоо шугаман регресс хийдэг болсон нь өмнөх RCNN илүү боловсронгуй болгосон. Зураг 1.17 -д Fast RCNN -н процессыг дурсэлсэн.



ЗУРАГ 1.17: Fast RCNN -н процесс

**Fastest RCNN /Region-based Convolutional Neural Network/** нь хамгийн хурдан ажилладаг RCNN архитектур. Илүү хурдан R-CNN нь бүсчлэлийн багцыг үүсгэхийн тулд бүсчлэлийн саналын аргыг ашигладаг [introdRCNN]. Илүү хурдан

R-CNN нь бүс нутгийн саналыг авах нэмэлт CNN-тэй бөгөөд үүнийг бүсчлэлийн саналын сүлжээ гэж нэрлэдэг. Сургалтын үед бүсчлэлийн саналын сүлжээ нь feature map -г оролт, гаралт болгон бүсчлэлийн саналыг авдаг. Эдгээр саналууд нь цаашдын процедурын хувьд сонирхлын бүсчлэлийг нэгтгэх давхаргад очдог. Зураг 1.18 -д Fastest RCNN -н процессыг дүрсэлсэн.



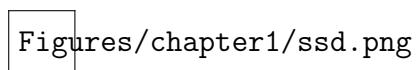
ЗУРАГ 1.18: Fastest RCNN -н процесс

#### ХҮСНЭГТ 1.1: RCNN архитектуруудын харьцуулалт

	RCNN	Fast RCNN	Fastest RCNN
<b>Ашиглагддаг арга</b>	Сонгомол хайлт OS	Сонгомол хайлт	Бүсчлэлийн саналын арга
<b>Таамаглах хугацаа</b>	40-50 сек	2 сек	0.2 сек
<b>Тооцоолол</b>	Тооцооллын хугацаа өндөр	Тооцооллын хугацаа өндөр	Тооцооллын хугацаа бага

### 1.3.2 SSD /Single Shot Detector/ илрүүлэлт

SSD нь үндсэн загвар ба SSD толгой гэсэн хоёр бүрэлдэхүүн хэсэгтэй. Үндсэн загвар нь ихэвчлэн урьдчилан бэлтгэгдсэн дурс ангилах сүлжээ юм [introdSSD]. Эцсийн бүрэн холбогдсон ангиллын давхаргыг устгасан ResNET, MobileNet шиг. Үндсэн загварын сүлжээнүүдээс feature map -үүд гарч ирдэг. SSD толгой нь зөвхөн нэг буюу хэд хэдэн эргэлтийн давхарга бөгөөд гаралт нь эцсийн давхаргын үр дүнгийн орон зайн байршил дахь объектуудын хязгаарлагдмал хайрцаг, ангиллаар тайлбарлагддаг. SSD бол зөвхөн нэг алхмаар зургаас feature map, таамаглалыг гарган авч, итгэлийн оноог тооцоолон, match хийгээд хамгийн өндөр оноотой ангийг сонгон тухайн объектод харгалзах хэмжээгээр хүрээ үүсгэн буцаадаг архитектур юм. Зураг 1.19 -д дүрслэгдсэн эхний хэдэн давхарга (цагаан хайрцаг) нь үндсэн давхарга, суулийн хэдэн давхарга (цэнхэр хайрцаг) нь SSD толгойг илэрхийлнэ.



ЗУРАГ 1.19: SSD -н процесс

### 1.3.3 YOLO /You Only Look Once/ илрүүлэлт

YOLO гэдэг нь зураг дээрх янз бүрийн обьектыг (бодит цаг хугацаанд) илрүүлж, таних алгоритм юм. YOLO дахь обьект илрүүлэлт нь регрессийн бодлого хэлбэрээр хийгдэг бөгөөд илрүүлсэн зургийн ангиллын магадлалыг өгдөг. YOLO алгоритм нь обьектуудыг бодит цаг хугацаанд нь илрүүлэхийн тулд convolutional neural network (CNN) ашигладаг [**introdYOLO**]. Нэрээс нь харахад алгоритм нь обьектыг илрүүлэхийн тулд нейроны сүлжээг зөвхөн нэг удаа л ашигладаг. Энэ нь бүхэл бүтэн зураг дээрх таамаглалыг нэг алгоритмын дагуу хийдэг гэсэн үг юм. YOLO алгоритм нь дараах гурван техникийг ашиглан ажилладаг.

- Үлдэгдэл блокууд
- Объектыг хүрээлэх хүрээний регресс
- Intersection Over Union (IOU)

Нэгдүгээрт, зургийг янз бүрийн сүлжээнд хуваана. Сүлжээ бүр  $S \times S$  хэмжээтэй байна. Зураг 1.20 дээр ижил хэмжээтэй олон тор нүд байна. Зураг дээрх сүлжээний нүд бүр тэдгээрийн дотор гарч буй обьектуудыг илрүүлэх болно. Жишээлбэл, хэрэв тодорхой сүлжээний нүдэнд обьектын төв гарч ирвэл энэ нүд үүнийг илрүүлэх үүрэгтэй.

Figures/chapter1/residual.png

ЗУРАГ 1.20: Зургийг блокуудад хуваах

Объектыг хүрээлэх хүрээний регресс : Объектыг хүрээлэх хүрээ нь зураг дээрх обьектыг тодотгох тойм юм. Зурган дээрх обьектыг хүрээлэх хүрээ бүр дараах шинж чанаруудаас бүрдэнэ.Өргөн (bw), өндөр (bh), анги (жишээлбэл, хүн, машин, гэрлэн дохио гэх мэт)- Үүнийг с үсгээр илэрхийлнэ. Хязгаарлах хайрцгийн төв (bx, by). Зураг 1.21 -д обьектыг хүрээлэх хүрээ болон хүрээний регрессийг харуулж байна.

Figures/chapter1/boundingbox.PNG

ЗУРАГ 1.21: Объектыг хүрээлэх хүрээ

YOLO нь обьектын өндөр, өргөн, төв, ангиллыг таамаглахын тулд нэг хязгаарлах хайрцгийн регрессийг ашигладаг. Дээрх зурган дээрх обьектын хязгаарлах хайрцагт гарч ирэх магадлалыг илэрхийлнэ.

Intersection over union (IOU) нь объект илрүүлэх узэгдэл бөгөөд хайрцгууд хэрхэн давхцаж байгааг тодорхойлдог. YOLO нь объектуудыг төгс хүрээлэх гаралтын хүрээг хангахын тулд IOU-г ашигладаг [[introdYOLO](#)]. Сүлжээний нүд бүр нь объектыг хүрээлэх хүрээ болон тэдгээрийн итгэлийн оноог урьдчилан таамаглах үүрэгтэй. Урьдчилан таамагласан хязгаарын хайрцаг нь бодит хайрцагтай ижил байвал IOU нь 1-тэй тэнцүү байна. Энэ механизм нь жинхэнэ хайрцагтай тэнцүү биш хязгаарлах хайрцгийг арилгадаг. Зураг 1.22 нь IOU хэрхэн ажилладаг тухай энгийн жишээг харуулж байна.

Figures/chapter1/intersection.jpeg

ЗУРАГ 1.22: Объектуудыг хүрээлсэн хүрээ огтлондоо

## 1.4 Ижил төстэй системийн судалгаа

“Upload” үйлдэл хийж оруулсан зургаас зөвхөн объект илрүүлэлт хийж чаддаг “Iashin” вэбсайт болон хараагүй хүмүүсийн суралцдаг Перкинс сургууль болон Техасын харааны бэрхшээлтэй хүмүүсийн сургууль (TSBVI) хоёрын хамтын вэбсайт болох Paths to Literacy -н судалгаагаар харааны бэрхшээлтэй иргэдэд тусалж чадах хамгийн шилдэг аппликеийн гэж зарлагдсан “TapTapSee” гар утасны аппликеийныг ижил төстэй системээр сонгон судална.

**Iashin.ai вебсайт :** 2019 онд анх интернэт сүлжээнд тавигдсан Владимир Иашин гэх судлаач, видео контентыг ойлгох нейроны сүлжээг зохион бүтээх чиглэлээр мэргэшсэн профессорын судалгааны ажилдаа зориулан хөгжүүлсэн үнэ төлбөргүй веб сайт юм [[iashin](#)]. Вэбсайтын хувьд back-end болон front-end гэсэн 2 хэсгээс бүрдэнэ. Front-end нь javascript ашиглан веб -д хуулагдсан зургийг уншиж авч дараа нь илгээх үүрэгтэй. Харин back-end хэсэг нь веб аппликеийн бий болгоход зориулагдсан Flask фреймворкыг ашигладаг бол өгөгдсөн зургаас объект илрүүлэлтийг хийхдээ YOLOv3 архитектур, COCO dataset, PyTorch машин сургалтын фреймворкыг ашигладаг. Өгөгдсөн зургаас 80 төрлийн зүйлийг олж чадна [[iashin](#)]. Зураг 1.23 -д Iashin.ai вэбсайтын нүүр хуудсыг харуулж байна.

Figures/chapter1/iashinweb1.PNG

ЗУРАГ 1.23: Iashin.ai веб сайтын нүүр хуудас

Зураг 1.24 -д iashin.ai сайтын “upload” хийж оруулсан зурагт хийсэн объект илрүүлэлтийн үр дүнг харуулж байна. Тус сайт нь объектыг хүрээлсэн хүрээ болон түүний ямар ангилалд хамгийн ихдээ хэд хувиар ижил байгааг зурагт дурслэн харуулдаг.

Figures/chapter1/iashinweb.PNG

ЗУРАГ 1.24: Iashin.ai веб сайт зурганд объект илрүүлэлт хийсэн нь

**TapTapSee мобайл аппликації** : нь хараагүй болон харааны бэрхшээлтэй хэрэглэгчдэд тусгайлан зориулсан гар утасны үнэгүй программ юм. Тус аппликаціяныг 2012 онд CloudSight компани анхны мобайл аппликаціянаа болгон гаргаж байсан [**cloudsightinc**]. Гар утасны Android болон IOS үйлдлийн системтэй хэрэглэгчид ашиглах боломжтой. TapTapSee нь төхөөрөмжийн камераар аливаа зүйлийн зураг, видеог авч, VoiceOver функцийг ашиглан илэрсэн объектуудыг хэрэглэгчид чанга дуугаар уншиж өгдөг. Объект илрүүлэлтдээ CloudSight -aac хөгжүүлсэн CloudSight API -г ашигладаг бөгөөд энэ нь хүний тархийг дуурайдаг, цаг хугацааны явцад алдаанаасаа "суралцдаг" гүн гүнзгий суралцах технологи, одоогоор энэ API нэг тэрбум гаруй зургийг зөв таньсан [**taptapsee**]. CloudSight API нь Large Language Model буюу их хэмжээний шошгогүй өгөгдөл дээр сургалцдаг self-supervised сургалтыг ашигласан нэг төрлийн нейроны сүлжээний загвар [**cloudsightapi**]. Энэ загвар нь ерөнхийдөө Transformer neural network буюу хувиргагч нейроны сүлжээг авч ашигласан байдаг.

Figures/chapter1/taptapsee.PNG

ЗУРАГ 1.25: TapTapSee аппликаційны лого

Дэлгэцийн баруун талд хоёр товшиж зураг авах эсвэл дэлгэцийн зүүн талд давхар товшоод видео хийх боломжтой. TapTapSee нь секундийн дотор аль ч өнцгөөс хоёр буюу гурван хэмжээст объектыг үнэн зөв тодорхойлох боломжтой. Төхөөрөмжийн доод хэсэгт илэрсэн объектууд текст хэлбэрээр гарч ирэх бол дараа нь төхөөрөмжийн VoiceOver функц тус текст мэдээллийг хэрэглэгчид чангаар хэлж өгдөг. Зураг 1.26 -д TapTapSee аппликація гар утсан дээр ажиллаж байгааг харуулж байна.

Figures/chapter1/taptapsee2.PNG

ЗУРАГ 1.26: TapTapSee аппликашн гар утсан дээр ажиллах байдал

ХҮСНЭГТ 1.2: Судлагдсан системүүдийн харьцуулалт

Системүүд	Vladimir Iashin веб	TapTapSee мобайл аппликашн
Платформын төрөл	Веб аппликашн	Мобайл аппликашн
Объектын байршил харуулах	Тийм	Үгүй
Илэрсэн объектуудын мэдээл- лийг текст руу хөрвүүлэх	Үгүй	Тийм
Илэрсэн объектуудын мэдээл- лийг аудио руу хөрвүүлэх	Үгүй	Тийм
Видеонаос объект илрүүлэлт хийх	Үгүй	Тийм
Илрүүлж чадах объектын нэр төрлийн тоо	80	80<
Төлбөртэй эсэх	Үгүй	Үгүй

## 1.5 Хөгжүүлэлтэд ашиглах алгоритм

Энэ бүлэг нь системийн хөгжүүлэлтийн гол хэсэг болох объект илрүүлэлтэд ашиглах алгоритмыг тайлбарлана. Дэлгэрэнгүй тайлбар болон судалгааг 2.2.1 -р дэд бүлэгт авч үзнэ.

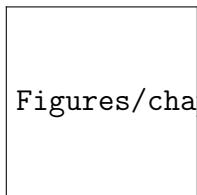
### 1.5.1 Объект илрүүлэлтэд ашиглах алгоритм

**MobileNet – SSD V3** буюу Microsoft -н COCO dataset дээр сургагдсан моделдийг системийн back-end хөгжүүлэлтийн объект илрүүлэлтэд ашиглана. **MobileNet** бол бага хэмжээний CPU ашиглах боломжийг олгодог нэг төрлийн жижиг хэмжээний, бага чадалтай CNN/Convolutional Neural Network/ архитектур. Анх 2017 онд Google компанийн инженерүүд “Мобайл вишн аппликашн -д илүү үр нөлөөтэй эргэлдсэн нейроны сүлжээ” /Efficient Convolutional Neural Network for Mobile Vision Application/ гэх судалгааны ажил дээрээ танилцуулж байсан [**mobilenet**]. **SSD /Single**

**Shot Detector**/нь нэг алхамт объект илрүүлэгч архитектур. 2016 онд Google компанийн судалгааны багийнхан бодит цагийн горимд хоцрогдолгуйгээр, цөөн алхмаар ажиллаж чадах объект илрүүлэх архитектур гаргасан нь SSD байсан. SSD -н процесс нь

1. “Feature map” – уудыг өргөтгөх
2. Объектыг илрүүлэх гэсэн үндсэн хоёр хэсэгт хуваагддаг.

SSD нь “base network” буюу “Feature map” – уудыг өргөтгөх процессоо тусдаа ажиллуулах боломжтой байдаг учраас эхний процессыг MobileNet архитектурыг ашиглан ажиллуулж, дараагийн процессыг SSD гүйцэтгэнэ. MobileNet –г Convolution үйлдэл хийлгэж, feature map -уудыг өргөтгөхийн тулд ашигладаг [ssd]. Зураг 1.27 нь MobileNet - SSD V3 -н архитектурыг харуулж байна.

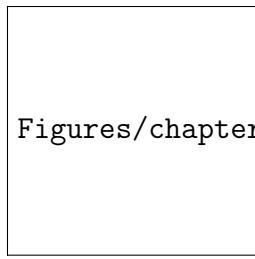


Figures/chapter1/mobilenetSSD-architecture.PNG

ЗУРАГ 1.27: MobileNet - SSD V3 архитектур

**Convolution үйлдэл :** Тус давхаргад зургийг матриц хэлбэрээр хүлээн авна. Ямар ч зургийг цэгэн утгуудын матрицаар илэрхийлж болдог. Өнгөт зургийг дэлгэцэд харуулахад улаан, ногоон, цэнхэр (Red, Green, Blue = RGB) өнгөний хослолоор шийддэг. Үүнийг, өнгө бүр нь нэг давхарга болж, дээр дээрээс нь давхарласан гурван давхарга гэж төсөөлж болох бөгөөд давхарга бүр дээрх цэгийн утга 0–255 хооронд тоон утга авна. Харин хар цагаан зураг бол нэг сувагтай. Нэг сувагтай гэдэг нь өнгөний нэг л давхарга байна гэсэн үг. Давхарга дээрх цэг бүрийн утга мөн л 0–255 хооронд байна. 0 (тэг) бол хар, 255 бол цагаан өнгө болно. /Энгийнээр, өнгөт зураг бол 3 давхаргаар дурслэгдэнэ, хар зураг бол нэг давхаргаар илэрхийлэгдэнэ. Давхарга(матриц) бүрийн утгууд 0-255 хооронд байна./

Цонх матриц буюу Filter/мөн kernel эсвэл feature detector ч гэж нэрлэдэг / - ийг үүсгэн түүнийгээ оролтын зурган матриц дээгүүр тодорхой алхмаар гүйлгэж тодорхой feature - үүдийг агуулсан шинэ матриц үүсгэдэг. Оролтын матриц болон цонх матрицын үржвэр, нийлбэрээс үүссэн гаралтын энэ матрицыг ‘Convolved Feature’ эсвэл ‘Activation Map’ эсвэл ‘Feature Map’ гэж нэрлэнэ. Зураг 1.28 -д зураг дээгүүр 3x3 хэмжээтэй Filter гүйлгэснээр ‘Feature Map’ хэрхэн үүсэж байгааг харуулж байна.



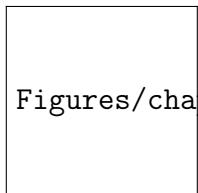
Figures/chapter1/kernelling.PNG

ЗУРАГ 1.28: Filter - хийж буй процесс

Шүүлтүүрийн тоо, түүний хэмжээ, сүлжээ (CNN) -ний архитектур бүтэц зэргийг сургалтаас өмнө бид тодорхойлж өгнө. Хэдий чинээ олон фильтртэй байна төдийн чинээ олон шинж тодорхойлогдоно. Ингэснээр тухайн зургаас суралцах зүйл ихсэж, өөр шинэ зургийг танихад илүү хэрэгтэй болно. Feature Map (Convolved Feature) -ын хэмжээ дараах гурван параметрээр тодорхойлогдоно.

1. **Depth буюу Гүн:** Энэ параметр нь convolution үйлдэлд хэдэн фильтер байхыг заана
2. **Stride буюу Алхам:** Энэ бол оролтын матриц дээгүүр шүүлтүүр матрицыг нэг удаад шилжүүлэх цэгийн тоо юм. Хэрэв алхам 1 гэвэл шүүлтүүр 1 цэгээр шилжинэ. Хэрэв алхам 2 гэвэл шүүлтүүр 2 цэгээр шилжинэ. Алхмын тоо том байвал feature map -ын хэмжээ жижиг болно.
3. **Zero-padding буюу Хоосон зай:** Оролтын матрицын гадна хүрээгээр хоосон зай ( $0=$ тэг утга) оруулах нь зарим тохиолдолд тохиromжтой байдаг. Ингэсэн үед шүүлтүүр нь оролтын матрицын ирмэгийн цэгт ч хурч чадна. Мөн feature map -ын хэмжээг удирдах боломж олгоно. Хоосон зай оруулсан бол wide convolution, хоосон зай оруулаагүй бол narrow convolution гэж нэрлэдэг.

Зураг 1.29 -с нэг л зураг өгөгдсөн байхад ялгаатай фильтерүүд ялгаатай шинжийг (ирмэг, зураас, тойрог, өнгө гэх мэт) тодруулж байгааг харж болно.

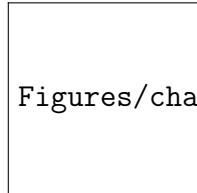


Figures/chapter1/cnn-filters.PNG

ЗУРАГ 1.29: Filter - н төрлүүд

Convolution -ы layer бүр дээр SSD нь объект илрүүлэлт хийнэ. Ингэхдээ convolution үйлдлээс гарч ирсэн layer -н channel болгоныг  $38 \times 38$ ,  $19 \times 19$ ,  $10 \times 10$ ,  $6 \times 6$  гэх мэт grid болгож grid -н нэг нүднээс 4-6 таамаглал гарган авдаг. Том хэмжээний объект нь

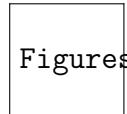
жижиг хэмжээтэй grid -г ашигласан үед, жижиг хэмжээний объект нь том хэмжээтэй grid -г ашигласан үед олддог. Зураг 1.30 -д өөр өөр хэмжээтэй grid -с таамаглал хэрхэн гарч байгааг харуулж байна.



Figures/chapter1/featureMaps.PNG

ЗУРАГ 1.30: Таамаглалыг сонгон авах

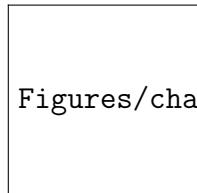
SSD нь таамагласан объектыг хүрээлж буй хүрээний өндөр, өргөнөө (1,2,3,0.5,0.3) гэсэн харьцаагаар бодон таамаглал бүртээ default хэмжээтэй хүрээ өгдөг. SSD -н талаарх энэхүү ойлголтын илүү ойлгомжтой дурслэлийг Зураг 1.31 -с харж болно.



Figures/chapter1/predictions.PNG

ЗУРАГ 1.31: Нэг цэг дээрх таамаглалууд

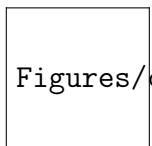
Таамаглал бурд өгч буй объектыг хүрээлсэн хүрээний урт, өргөнийг тооцоолохдоо Зураг 1.32 -д үзүүлсэн томьёогоор олно.



Figures/chapter1/f1.PNG

ЗУРАГ 1.32: Объектыг хүрээлсэн хүрээний урт, өргөнийг тооцоолох томьёо

Зураг 1.33 -д үзүүлсний дагуу таамаглал бурд multibox байна. Multibox нь дотроо тухайн таамаглалыг хүрээлж байгаа хүрээний x,y байршил, өндөр, өргөний хэмжээ болон таамагласан класс болгонд харгалзах итгэлийн оноо/confidence score/ -г агуулсан байна. SSD нь таамаглал бүрээ авч match хийнэ. Нэгтэй тэнцүү эсвэл нэгтэй хамгийн ойр итгэлийн оноотой классыг сонгон түүнийгээ объектын илрүүлэлт болгон авна.



Figures/chapter1/ssd-prediction.PNG

ЗУРАГ 1.33: Таамаглалын multiBox

**MobileNet - SSD V3 моделийн илрүүлэлтийн нарийвчлал :** Малайз улсын “MARA Технологийн Их сургууль” -ийн “Компьютер Математикийн тэнхим”-ийн MobileNet SSD болон YOLO V3 илрүүлэлтийн архитектуруудыг харьцуулсан судалгаа нь MobileNet SSD -н илрүүлэлтийн нарийвчлалын талаар дараах үр дүнг харуулсан.

- MobileNet SSD архитектурыг ашигласан сургалтыг явуулсан техник хангамжийн үзүүлэлт : Intel Core i5-7300HQ 4 цөмт, хамгийн дээд давтамж нь 3.5 GHz хүрч чадах CPU, 8GB RAM, 4GB VRAM -тай NVIDIA -н GTX 1050 GPU.
- Сургалтад ашигласан өгөгдөл : MS COCO dataset -н 2.5 сая шошгыг агуулсан 328000 зураг [accuracy-mobilenet]

Сургалт нь нийт 5 цаг орчим үргэлжилж 128000 алхамт ажлыг гүйцэтгэсэн. Объект илрүүлэлтийн архитектуруудын илрүүлэлтийн нарийвчлалыг хэмждэг нэгж mAP буюу дундаж нарийвчлалаар үр дүнг хэмжсэн. [accuracy-mobilenet] mAP нь нарийвчлалыг 0-1 гэсэн утгын хооронд хэмждэг. Зураг 1.34 -д үзүүлсний дагуу сургалтын эхэн үед илрүүлэлтийн нарийвчлал 0.4mAP байсан бол 4 дэх цагаас эхлэн 0.53mAP орчим болж тогтворжиж байна.

Figures/chapter1/map-mobilenet.PNG

ЗУРАГ 1.34: MobileNet SSD архитектурын дундаж нарийвчлал

Хүснэгт 1.3 -д судалгааны үр дүнд гарсан MobileNet SSD архитектурын илрүүлэлтийн нарийвчлалыг харуулж байна.

ХҮСНЭГТ 1.3: MobileNet SSD архитектурын үзүүлэлтүүд

Үр дүн	MobileNet SSD V3
mAP	53.29%
mAP /Дундаж хэмжээтэй Объект/	58.33%
mAP /Жижиг хэмжээтэй Объект/	53.34%
mAP(0.5IOU)	97.8%
mAP(0.75IOU)	51.28%
Илрүүлэлтийн хурд	135.5мс

Объект илрүүлэлтийн модел болгоны объект илрүүлэлтийн нарийвчлал нь ашиглаж байгаа архитектур, өгөгдлийн тоо, сургалтын хугацаа болон сургалт явуулж байгаа техник хангамжаас шалтгаалан бага зэргийн зөрүүтэй байх боломжтой.

## 1.6 Технологийн судалгаа

Системийн back-end болон front-end хөгжүүлэлтэд ашиглах технологиуд болон хөгжүүлэлтийг хэрэгжүүлэхэд шаардлагатай хөгжүүлэлтийн орчны тайлбар судалгаа зэргийг агуулсан.

### 1.6.1 Back-End хөгжүүлэлтэд ашиглах технологиуд

#### OpenCV зургийн сан

OpenCV/Open Computer Vision/ нь 1999 онд Intel -н CPU ихээр ашигладаг программуудад зориулсан, C++ хэл дээр хөгжүүлэгдсэн сан. Одоо OpenCV нь Apache 2.0 лиценцийн дагуу нээлттэй эхийн компьютер вишн, машин сургалтад ашиглагддаг 2500+ алгоритмыг агуулсан сан юм. OpenCV нь Python, MATLAB, Java хэлийг дэмждэг, Linux, Windows, macOS, OpenBSD, NetBSD, FreeBSD үйлдлийн систем дээр ажиллаж чадна. Хэрвээ OpenCV нь системээс Intel -н Integrated Performance Primitivies/Нэгдсэн гүйцэтгэлийн командууд/-г олж чадвал оновчтой горимуудыг ашиглан өөрийгөө хурдастгаж чадна [OpenCV].

Figures/chapter1/opencv.png

ЗУРАГ 1.35: OpenCV лого

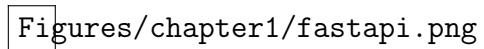
Царай таних, дохио зангаа таних, видео дээрх хөдөлгөөнийг бүртгэн авах, AR/Augmented Reality/ зэрэг компьютер вишн -н олон төрөлд ашиглах боломжтой сангуйдыг агуулсан. Мөн машин сургалтын "boosting" алгоритм, Бейсийн тодорхойлогч алгоритм, шийдвэрийн модны сургалтын алгоритм, "k-nearest neighbour" алгоритм гэх мэт олон алгоритмуудыг хэрэгжүүлдэг. Зургаас объект илрүүлэлтийг хийхийн тулд OpenCV -гийн DNN пакежийн DetectionModel классыг ашиглана. Энэ класс нь объект илрүүлэлтэнд зориулсан өндөр түвшний API -г илэрхийлдэг, сургагдсан моделийн weight болон configuration файлыг ашиглан зургаас объект илрүүлэлт хийх сүлжээг бий болгодог. Ингэснээр аль хэдийн сургагдсан моделийн weight болон давхаргуудаар сүлжээ үүсгэн, тэр сүлжээг ашиглан зургаас объект илрүүлэлтийг хийнэ. Энэ класс нь байгуулагч функцийн 2 аргумент авдаг. Эхнийх нь сургагдсан моделийн weight -г агуулсан файл, сүүлийнх нь configuration файл байна. DetectionModel класс нь SSD, Faster R-CNN, YOLO архитектураар хөгжүүлэгдсэн моделийг дэмждэг.

## FAST API

FAST API нь 2018 онд анхны хувилбараа гаргаж байсан, MIT License -тэй Python3.7+ дээр API/Application Programming Interface/ үүсгэхэд зориулагдсан орчин үеийн хурдан, өндөр гүйцэтгэлтэй веб фреймворк юм.

Гол онцлогууд нь :

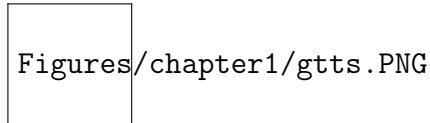
- **Хурдан:** NodeJS болон GO -тэй ижил өндөр гүйцэтгэлтэй, хамгийн хурдан Python веб фреймворкуудын нэг.
- **Хурдан кодчлол:** Фреймворкуудын онцлогуудыг ашигласнаар хөгжүүлэлтийн хурдыг 200%-300% -р нэмэгдүүлнэ.
- **Цөөн алдаа:** Хөгжүүлэгчээс үүдэлтэй алдааны 40 орчим хувийг бууруулна.
- **Хялбар:** Ашиглах болон сурахад хялбар байхаар бүтээгдсэн. Документ унших цагийг багасгасан.
- **Найдвартай:** Автомат интерактив баримт бичгийг бий болгодог.



ЗУРАГ 1.36: FAST API лого

## Python gTTS сан

Python хэлний gTTS сан нь Google Translate -н текстийг яриа руу хөрвүүлдэг функцийг ашигладаг бөгөөд энэ нь Google Cloud Text-to-Speech -ээс ялгаатай. Энэ функцийг CLI түүл хэлбэрээр ч ашиглаж болно. Нийт 17 хэлийг таньж оролтын текстийг нь mp3 форматтай аудио болгож чадна.



ЗУРАГ 1.37: Google translate болон Python лого

## Chimege API

Chimege API нь Чимэгэ системс ХХК -ын гаргасан монгол хэл дээрх текстийг аудио руу, аудиог текст хэлбэр рүү хөрвүүлэх боломжийг олгодог. “Чимэгэ уншигч” endpoint бол хүссэн бичвэрийг хиймэл оюунаар боловсруулж, амьд мэт хоолойгоор уншдаг. Үнийн хувьд Chimege API -н нэг endpoint -г дуудахад 8 үнэтэй байдаг. Яриа руу хөрвүүлэх бичвэрийг мөн засаж хөрвүүлдэг. Жишээлбэл:

- Шаардлагагүй тэмдэгтүүдийг хасна.
- Түгээмэл товчилсон үгсийг задална. (МУ - Монгол Улс)
- Тоог задална. (123 - Зуун хорин гурав)

[Figures/chapter1/chimege.png]

ЗУРАГ 1.38: Чимэгэ Системс лого

### MySQL өгөгдлийн сан удирдах систем

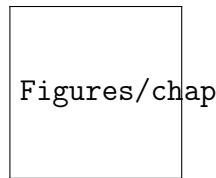
MySQL нь бүтэцлэгдсэн өгөгдлийн сангийн менежментийн систем. Ахь Майкл Видениус зохион бүтээж байсан бөгөөд өөрийн охин My -ын нэр болон SQL хэлний нэрийг нийлүүлж нэрлэсэн бөгөөд анхны хувилбараа 1995 онд гаргасан. 2010 онд тус өгөгдлийн сангийн менежментийн системийг Oracle компани худалдаж авсан, одоо MySQL нь GNU General Public License-ийн нөхцөлийн дагуу үнэгүй, нээлттэй эхийн программ хангамж. MySQL нь C болон C++ хэл дээр бичигдсэн, AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, IBM i, IRIX, Linux, macOS, Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, зэрэг олон системийн платформ дээр ажилладаг.

[Figures/chapter1/mysql.png]

ЗУРАГ 1.39: MySQL лого

### PyCharm IDE

PyCharm нь Python, веб, өгөгдлийн шинжлэх ухааныг үр бүтээлтэй хөгжүүлэхэд тохиромжтой орчныг бүрдүүлэхийн тулд нягт нэгтгэсэн, Python хөгжүүлэгчдэд зориулсан өргөн хүрээний чухал хэрэгслээр хангагдсан Python-ийн нэгдсэн хөгжүүлэлтийн орчин (IDE) юм. Java болон Python хэл дээр JetBrains компаниар хөгжүүлэгдсэн, 2010 онд анхны хувилбараа гаргасан, Linux, Windows, macOS дээр ажиллах боломжтой. Нээлттэй эхийн Community болон төлбөртэй Professional хувилбаруудтай. Anaconda, IPython, Kite, Pylint, Pytest зэрэг машин сургалт, шинжлэх ухааны тооцоололд ашиглагддаг сангийн, плаг-ин, фреймворкуудыг агуулсан.



ЗУРАГ 1.40: PyCharm IDE лого

PyCharm IDE -н давуу талууд :

- Pandas, Scikit-learn зэрэг шинжлэх ухааны янз бүрийн номын санд хандах боломжтой.
- Линзний функцээр нарийвчилсан дүн шинжилгээ хийх боломжтой.
- Синтаксын алдааг олоход хялбар болгогч кодын алдаа засагчтай.
- Автоматаар бөглөх синтакс функц нь цаг хэмнэдэг

PyCharm IDE -н сул талууд :

- Сурахад их цаг зарцуулдаг.
- Үндсэн скриптэд нэг их үнэ цэнэ нэмдэггүй.
- Олон нийтийн хувилбар нь бусад кодчиллын хэл рүү нэвтрэх эрхийг хязгаарладаг.

### 1.6.2 Front-End хөгжүүлэлтэнд ашиглах технологиуд

#### React фреймворк

React/ReactJS, React.js гэж бас хэлэгддэг / нь веб апликашны User Interface/хэрэглэгчийн интерфейс/- г хөгжүүлэхэд зориулагдсан JavaScript фреймворк юм. 2011 онд Мета компанийн программ хангамжийн инженер Жордан Уолке JavaScript хөгжүүлэлтийн нарийн төвөгтэй байдлыг арилгахын тулд components/бүрэлдэхүүн хэсгүүд/-д суурисан, дахин ашиглах боломжтой React фреймворкыг хөгжүүлж, 2013 онд анхны хувилбараа гаргаж байсан.

ReactJS -н давуу талууд :

- Суралцах болон хэрэглэхэд хялбар. Ойлгоход хялбар документ болон зааварчилгаатай. Хамгийн олон практик хичээлтэй фреймворк.
- JavaScript Extension -г ашиглаж динамик веб хөгжүүлэхэд хялбар болгосон.
- Virtual DOM буюу HTML, XHTML, XML -тай ажилладаг API -г ашигласнаар гүйцэтгэлийг нэмэгдүүлсэн.
- Дахин ашиглах боломжтой component/бүрэлдэхүүн хэсэг/ -г бий болгодог.

ReactJS -н сул талууд :

- Байнгын шинэчлэл. Энэ нь хөгжүүлэгчдийн хувьд шинэчлэлтийг байнга су-далж, хөгжүүлэлтдээ нэвтрүүлэх шаардлагатай байdag учраас хөгжүүлэлтийг удаашруулах хандлагатай байдаг.
- Системийн хувьд зөвхөн хэрэглэгчийн интерфейсийн хэсгийг хамардаг учраас системийг бүхэлд нь хөгжүүлэхийн тулд бусад технологийг ч бас нэвтрүүлэх шаардлага гардаг.

▪ Figures/chapter1/react.png

ЗУРАГ 1.41: React лого

### Ant design

Ant Design нь Alibaba, Alipay, Huabei, MYbank-н толгой компани болох Ant Group-ийн боловсруулсан нээлттэй эхийн дизайны систем юм. 2017 онд анхны хувилбараа гаргаж байсан. Ant Design -н бүрэлдэхүүн хэсгүүд нь React, Vue болон Angular гэх мэт front-end хөгжүүлэлтийн фреймворкуудыг дэмждэг. Тус системийн онцлог давуу талууд нь системийг хэрэглэх гол хүчин зүйл нь болж байдаг.

Ant Design -н онцлогууд:

- **Арчилгаа сайтай:** Ant Design-ийн баг байнгын шинэчлэлтуудээр дизайны системийг сайжруулахаар тасралтгүй ажилладаг.
- **Цогц дизайны сан:** Дизайны ямар ч асуудлыг шийдэж чадах бүрэлдэхүүн хэсэг, загвар болон тэмдэгт дурсуудийг агуулсан байдаг. Нэмж дурдахад элемент бүр нь ямар ч хувилбарт тохирох олон харагдацтай байдаг.
- **Интернационал:** Дэлхийн өнцөг булан бүрийн хэлийг дэмждэг бөгөөд хөгжүүлэгчид илүү ихийг нэмэх боломжтой.
- **Маягт:** Ямар ч төрлийн маягтыг бий болгож чадна.
- **Бэлэн загвар:** Өгөгдлийн үзүүлэлт, тайлан, график, админы интерфейс, чат, нэвтрэх хэсэг зэрэг олон төрлийн интерфейсүүдэд зориулсан 100+ бэлэн загваруудтай.

▪ Figures/chapter1/antdesign.jpg

ЗУРАГ 1.42: Ant design лого

## Visual Studio Code

Visual Studio Code /VS Code гэж нэрлэдэг/ нь Microsoft, Electron Framework-тай хамтран хийсэн Windows, Linux болон macOS-д зориулсан кодын засварлагч юм. 2015 оны 11-р сарын 18-нд Visual Studio Code-ийн эх код MIT лицензийн дагуу гарч, GitHub дээр тавигдсан. Онцлогуудад нь дигитал хийх, синтакс онцлох, ухаалаг код гүйцээлт, хэсэгчилсэн хэсгүүд, кодын дахин боловсруулалт, суулгагдсан Git зэрэг орно. Хэрэглэгчид загвар, гарын товчлол, тохиргоог өөрчлөх, нэмэлт функц бүхий өргөтгөлүүдийг суулгах боломжтой. Visual Studio Code нь C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust зэрэг олон төрлийн программчлалын хэлээр ашиглах боломжтой. Энэ нь Blink layout хөдөлгүүр дээр ажилладаг Node.js вэб программуудыг боловсруулахад хэрэглэгддэг Electron framework дээр суурилдаг. Visual Studio Code нь хамгийн түгээмэл программчлалын хэлнүүдийн үндсэн дэмжлэгийг агуулдаг. Энэхүү үндсэн дэмжлэг нь синтакс тодруулах, хаалт тааруулах, код тохицуулах боломжтой хэсгүүдийг агуулдаг. Visual Studio Code нь JavaScript, TypeScript, JSON, CSS, HTML-д зориулсан IntelliSense-тэй, мөн Node.js-ийн дигитал хийх дэмжлэгтэй. Нэмэлт хэлний дэмжлэгийг VS Code Marketplace дээр чөлөөтэй ашиглах боломжтой.

Figures/chapter1/vscode.png

ЗУРАГ 1.43: Visual Studio Code лого

### 1.6.3 Кодын менежмент

## 1.7 Бүлгийн дүгнэлт

Тус бүлэгт объект илрүүлэлтийн гэж юу болох талаар судалж, зургаас объект илрүүлэлт хийх болон зураг таних гэсэн ойлголтуудыг ялгааг судалж тайлбарласан. Объект илрүүлэлтийн уламжлалт аргад тооцогдох Виола Жонес, HOG болон DPM илрүүлэлтүүд, орчин үеийн архитектуруудад тооцогдох RCNN, FastRCNN, FasterRCNN, SSD, YOLO тэдгээрийг хэрэгжүүлэх процессын талаар судалсан. Уламжлалт аргууд нь ихэвчлэн тодорхой биетийг зураг дээр дурслэгдэхдээ яаж дурслэгддэгийг тодорхойлоод тэр тодорхойлолтоо зургаас хайдаг. Жишээ нь хүний царай зурагт дурслэгдэхдээ дүх хэсэг нь цайвар, нүд хэсэг нь бараан, хамар хэсэг нь бараан, хамрын

хоёр талаар цайвар харагдана гэж үзээд тийм дурслэлийг зургаас хайдаг. Харин орчин үеийн аргуудын гол архитектур болох CNN нь зургийн пиксел бүртэй ажиллаж олон төрлийн филтерээр филтердэж, олон төрлийн feature map -уудыг гарган авч тэдгээрээс хамгийн тохиromжтой объектоо тооцоолон гаргадаг.

Тус бүлэгт ижил төстэй системүүд, системийг хөгжүүлэхэд шаардлагатай технологии болон алгоритмуудыг тайлбарласан. Iashin.ai гэх зөвхөн зурагнаас объект илрүүлэлт хийдэг веб сайт болон TapTapSee гэх зураг болон видеоноос объект илрүүлэлт хийж зураг болон видеоны агуулгыг хэрэглэгчид текст болон аудио хэлбэрээр хүргэдэг мобайл аппликашныг ижил төстэй системээр сонгон судалж харьцуулалт хийсэн. Объект илрүүлэлт хийх MobileNet V3 - SSD архитектур, сургагдсан моделийг шууд ашиглах боломжийг олгодог OpenCV сан, хөгжүүлэлтийн back-end болон front-end хэсгүүдийг холбоход туслах Python-д зориулсан FastAPI фреймворк. Текстэн мэдээллийг аудио руу хөрвүүлэхэд хэрэглэгдэх Text-to-Speech API, Chimege API. Front-end хөгжүүлэлтэнд ашиглах React фреймворк, AntDesign CSS сан болон PyCharm, VS Code хөгжүүлэлтийн орчнуудыг судалж тайлбарласан.

---

---

## БҮЛЭГ 2

---

... системийн шинжилгээ

## 2.1 Системийн үйл ажиллагааны тухай дэлгэрэнгүй

Энэхүү систем нь хэрэглэгчээс зурган файлыг авч тухайн зурагт дүрслэгдсэн объектуудын нэрийг олж, тэдгээрийн нэрийг байршилтай нь хамт тухайн зураг дээр нь хүрээлсэн хайрцагт хийж тодотгож харуулдаг объект илрүүлэлт хийдэг веб юм. Объектын илрүүлэлтийг хийж, зургийг хэрэглэгчид буцаан харуулахаас гадна хэрэглэгчид дүрслэгдсэн объектуудыг, илэрсэн тоотой нь хамт текст болон монгол, англий хэл дээрх аудио файл болгон хөрвүүлж өгнө [Reference3]. Хэрэглэгчид утасны дугаар эсвэл өөрийн имейл хаягаар бүртгэл үүсгэж системд нэвтэрвэл тухайн хэрэглэгчийн хэрэглэгчийн бүртгэлээрээ нэвтэрсэн үедээ хийсэн объект илрүүлэлтүүд хадгалагдсан байна. Үүнд объект илрүүлэлт хийсний дараах шинэчлэгдсэн зураг, зурагт хамаарах объект илрүүлэлтийн мэдээллийг агуулсан текст болон аудио файлууд огноотойгоо хамт хадгалагдсан байна. Системийн объект илрүүлэлт нь илрүүлсэн объектоо дараах 80 төрлийн зүйлд хуваан танина. Хүн, унадаг дугуй, машин, мотоцикл, онгоц, автобус, галт тэрэг, ачааны машин, завь, гэрлэн дохио, галын цорго, зогсох тэмдэг, зогсоолын тоолуур, вандан, шувуу, муур, нохой, морь, хонь, үхэр, заан, баавгай, тахь, анааш, үүргэвч, шүхэр, гар цүнх, зангиа, чөмодан, фрисби, цана, снөубоард, спортын бөмбөг, цаасан шувуу, бейсболын цохиур, бейсболын бээлий, скейтборд, серфингийн самбар, тенисний цохиур, лонх, дарсны шил, аяга, эрээ, хутга, халбага, аяга, банана, алым, сэндвич, жүрж, брокколи, лууван, хот дог, пицца, донат, бялуу, сандал, бүйдан, савтай ургамал, ор, хоолны ширээ, бие засах газар, ТВ монитор, зөөврийн компьютер, хулгана, удирдлага, гар, гар утас, богино долгионы зуух, зуух, талх шарагч, угаалтуур, хөргөгч, ном, цаг, ваар, хайч, бамбарууш, үс хатаагч, шүдний сойз.

## 2.2 Системийг ашиглах хэрэглэгчид

Тус системийг нас, хүйс хамаарахгүйгээр объект илрүүлэлтийг ашиглаж үзэхийг хүссэн, интернэтэд холбогдсон хэн бүр ашиглаж болно. Системийг ашиглаж буй хэрэглэгчдийг бүртгэлтэй болон бүртгэлгүй гэж 2 ангилана.

## 2.3 Функцийн шаардлага

Хэрэглэгчийн функцийн шаардлага

1. Хэрэглэгч объект илрүүлэлтэд ашиглахыг хүссэн зургаа веб -д байршуулна.
2. Аудио файлын хэлийг монгол эсвэл англи хэлний аль нэгээр нь сонгоно.

### Бүртгэлтэй хэрэглэгчийн функцийн шаардлага

1. Хэрэглэгч системд овог нэр, утасны дугаар, имейл хаяг, нууц үг гэсэн талбаруудыг бөглөж бүртгүүлнэ
2. Имейл хаягаар илгээгдэх ОТР кодоор бүртгэлээ баталгаажуулна
3. Нэр, утас, имейл хаягийн мэдээллийг агуулсан хэрэглэгчийн бүртгэлийн хэсэгт бүртгэлээ засдаг байна.
4. Бүртгэлтэй хэрэглэгч өөрийн бүртгэлээ устгаж болно, энэ тохиолдолд бүртгэлтэй хэрэглэгчид хамааралтай бүх файл устана.
5. Хэрэглэгчийн бүртгэлээр нэвтэрсэн үедээ ашигласан зургуудын объект илрүүлэлтийг агуулсан шинэчилсэн зургууд, тэдгээрт хамаарах дурслэгдсэн объектуудын мэдээллийг агуулсан текст болон аудио файлууд огноотойгоо хадгалагдсан байна.
6. Хэрэглэгч өөрт бүртгэлтэй байгаа объект илрүүлэлтийг агуулсан зургууд, тэдгээрт хамаарах текст болон аудио файлтай нь хамт устгаж болно.

### Системийн функцийн шаардлага

1. Хэрэглэгчийн байршуулсан зураг дээр объект илрүүлэлт хийж, объект илрүүлэлтийг агуулсан зургийг хэрэглэгчид буцаан харуулдаг байна.
2. Объект илрүүлэлтэд илэрсэн объектуудын нэр болон тоо хэмжээг агуулсан текстэн мэдээллийг хэрэглэгчид харуулна.
3. Объект илрүүлэлтэд илэрсэн объектуудын нэр болон тоо хэмжээг илэрхийлсэн монгол эсвэл англи/хэрэглэгчийн сонголтоор/ хэл дээрх mp4 өргөтгөлтэй аудио файлыг буцаана.
4. Бүртгэл үүсгэж буй хэрэглэгчийн бүртгэлийг баталгаажуулахын тулд имейл хаягаар ОТР код илгээнэ.

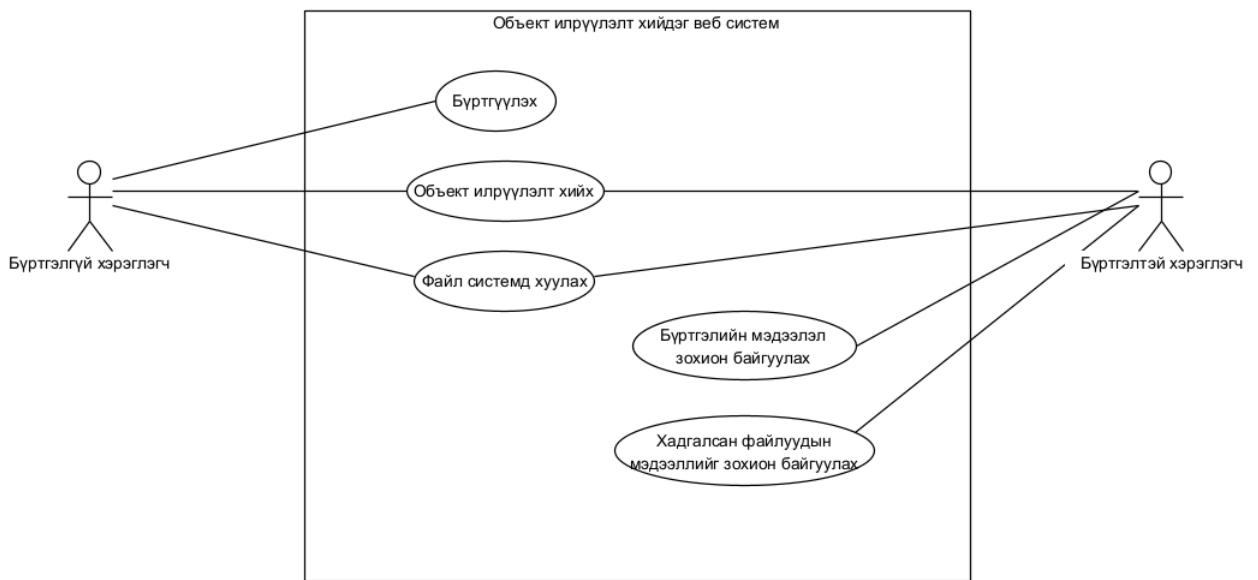
## 2.4 Функцийн бус шаардлага

1. Хэрэглэгч болон бүртгэлтэй хэрэглэгчийн интерфейс ойлгомжтой, минимал загвартай байна.
2. Бүртгэлтэй хэрэглэгчийн нууц үг хамгийн багадаа 6 ширхэг том, жижиг үсэг, тоо, тэмдэгтийг агуулсан байна.
3. 200MB хүртэлх хэмжээтэй файлыг веб -д хуулна.
4. Веб -д хуулах файл нь jpg, jpeg өргөтгөлтэй байна.
5. Front-end болон back-end системүүдийн хооронд мэдээлэл солилцох API -н хүсэлт нь multipart-form-data хэлбэртэй байх бол хариу нь Application JSON форматтай байна.

6. Package ба Import statements - Пакеж доторх классуудтай холбоотой нэр үгээр пакежийг нэрлэнэ.
7. Class Header ба Declaration - Классын нэрийг тухайн класс дотор хийгдэж байгаа үйл ажиллагаатай холбоотой нэр үгээр нэрлэнэ
8. Method Headers ба Declarations - Метод буюу дэд функцийг тухайн функцийн гүйцэтгэх үйл үгээр нэрлэнэ
9. Функцийн дээд мөрөнд функцийн тайлбарыг коммент хэлбэрээр бичсэн байна.
10. Хөгжүүлэлтийн туршид хөгжүүлэлтийн нэг phase бүр дээр нэгжийн тестийг хийдэг байна.

## 2.5 Юзкейс диаграмм

Зураг 2.1 -д дүрслэгдсэн объект илрүүлэлт хийдэг веб системийн юзкейс диаграмм нь бүртгэлтэй болон бүртгэлгүй хэрэглэгч гэсэн тоглогчид болон тэдгээрийн ашиглаж болон нийт 5 юзкейсыг агуулсан.



ЗУРАГ 2.1: Юзкейс диаграмм

## 2.6 Юзкейсийн тодорхойлолт

ХҮСНЭГТ 2.1: Системд файл хуулах юзкейсийн тодорхойлолт

Юзкейс:	Системд файл хуулах
---------	---------------------

ID:	1
Үүсгэсэн:	Хүслэн
Үүсгэсэн огноо:	2023.03.15
Үндсэн тоглогч:	Бүртгэлтэй хэрэглэгч эсвэл бүртгэлгүй хэрэглэгч
Нэмэлт тоглогч:	Байхгүй
Товч тайлбар:	Хэрэглэгч объект илрүүлэлтэд ашиглах файлыг веб хуудас руу хуулна.
Өмнөх нөхцөл:	<ol style="list-style-type: none"> <li>Хэрэглэгч объект илрүүлэлт хийдэг вебийг броузер дээрээ нээсэн байна.</li> <li>Веб хуудас руу хуулах гэж файл нь веб хуудас руу хандалт хийж байгаа төхөөрөмж дээр байна.</li> </ol>
Үндсэн урсгал:	<ol style="list-style-type: none"> <li>Хэрэглэгч веб хуудасны "Байршуулах"button -г дарна.</li> <li>Төхөөрөмж дээр байх файлуудыг харуулсан "Open" цонх хэрэглэгчид харагдана.</li> <li>Хэрэглэгч веб хуудас руу хуулах файлаа сонгоод "Open" цонхны "Open"button -г дарна.</li> <li>"Open" цонх хаагдана.</li> <li>if Файл хуулалт амжилттай болсон бол             <ol style="list-style-type: none"> <li>Вебийн файл байршуулах хэсэгт файл байршсан байна.</li> </ol> </li> <li>if Файл хуулалт амжилтгүй болсон бол             <ol style="list-style-type: none"> <li>"Амжилтгүй, дахин оролдоно уу + алдааны мессеж" гэсэн контексттэй pop-up -г хэрэглэгчид харуулна.</li> <li>Файл байршуулах хэсэгт файл байршаагүй байна.</li> </ol> </li> </ol>
Дараах нөхцөл:	<ol style="list-style-type: none"> <li>Системд файл хуулагдсан байна.</li> </ol>
Альтернатив урсгал:	Байхгүй.

ХҮСНЭГТ 2.2: Объект илрүүлэлт хийх юзкейсийн тодорхойлолт

Юзкейс:	Объект илрүүлэлт хийх
ID:	2
Үүсгэсэн:	Хүслэн

Үүсгэсэн огноо:	2023.03.15
Үндсэн тоглогч:	Бүртгэлтэй хэрэглэгч эсвэл бүртгэлгүй хэрэглэгч
Нэмэлт тоглогч:	Байхгүй
Товч тайлбар:	Веб хуудсанд байршуулсан файлд объект илрүүлэлт хийх хүсэлт явуулна.
Өмнөх нөхцөл:	<ol style="list-style-type: none"> <li>Веб хуудсанд объект илрүүлэлтэд ашиглах файлыг хуулсан байна..</li> </ol>
Үндсэн урсгал:	<ol style="list-style-type: none"> <li>Хэрэглэгч объект илрүүлэлтийн мэдээллийг хүлээн авах хэлээ сонгосон байна.</li> <li>Хэрэглэгч "Объект илрүүлэх"button дарна.</li> <li>if Объект илрүүлэлт амжилттай бол             <ol style="list-style-type: none"> <li>Объект илрүүлэлт хийсэн файлыг хэрэглэгчид файл байршуулах хэсэгт буцаан харуулна.</li> <li>Файлд илэрсэн обьектуудын нэрийг текст мэдээлэл болгон сонгосон хэл руу хөрвүүлэн "Текст мэдээлэл"хэсэгт харуулна.</li> <li>Текст мэдээллийг аудио руу хөрвүүлэн "Аудио мэдээлэл"хэсэгт харуулна.</li> <li>Объект илрүүлэлтэд ашигласан файл түүний үр дүнг агуулсан текст болон аудио файлуудыг систем бүртгэлтэй хэрэглэгчийн "Хадгалагдсан файлууд"руу нэмнэ.</li> </ol> </li> <li>if Объект илрүүлэлт амжилтгүй болсон бол             <ol style="list-style-type: none"> <li>"Амжилтгүй, дахин оролдоно уу + алдааны мессеж"гэсэн контексттэй рор-ир -г хэрэглэгчид харуулна.</li> <li>Файл байршуулах хэсэгт файл харагдахгүй, хэлний сонголт арилсан байна.</li> </ol> </li> </ol>
Дараах нөхцөл:	<ol style="list-style-type: none"> <li>Объект илрүүлэлтийн үр дүнг хэрэглэгч харсан байна.</li> </ol>
Алтернатив урсгал:	Байхгүй.

## 2.7 Шинжилгээний класс диаграмм

Зураг 2.2 дээрх системийн шинжилгээний шатны класс диаграмм нь системд үндсэн үүрэг гүйцэтгэх "control" төрөлтэй байж болох 6 классыг тодорхойлж тэдгээрийн хоорондын холбоо хамаарлыг дурсэлсэн. Үр дунгийн боловсруулалт класс нь объект илрүүлэлт хийх хүсэлтийг хүлээн авч, үр дунг нэгтгэн боловсруулаад буцаах үүрэгтэй класс, Хэлний төрөл класс нь хөрвүүлэлт хийж болох хэлний төрлүүдийг агуулсан класс, Объект илрүүлэлт класс нь хэрэглэгчийн хуулсан файлд объект илрүүлэлт хийж илэрсэн объектуудын төрөл болон объект илрүүлэлт хийсэн шинэ файлыг Үр дүнгийн боловсруулалт класс руу илгээдэг. Хэрэглэгчийн файл болон Хэрэглэгч классууд нь хэрэглэгчийн мэдээлэл болон хэрэглэгчийн файлын мэдээлэл зохион байгуулдаг классууд, Бүртгэл класс нь бүртгэлтэй холбоотой ОТР үүсгэж, шалгаж, бүртгэл үүсгэдэг класс юм.

Figures/chapter2/sh-class.pdf

ЗУРАГ 2.2: Шинжилгээний класс диаграмм

## 2.8 Шинжилгээний дарааллын диаграмм

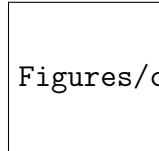
Шинжилгээний класс диаграммд тодорхойлсон control төрлийн 5 классын операторуудыг ашиглан файл системд хуулах, объект илрүүлэлт хийх, бүртгүүлэх, хэрэглэгчийн файлын мэдээлэл зохион байгуулах, бүртгэлийн мэдээлэл зохион байгуулах гэсэн юзкейсүүдэд шинжилгээний шатны дарааллын диаграммыг гаргасан.

Бүртгэлтэй хэрэглэгч бүртгэлийн мэдээллээ шинэчлэх эсвэл устгах үед дараах дарааллаар процесс явагдана. Бүртгэлтэй хэрэглэгчийн хүсэлт хэрэглэгчийн мэдээллийг удирдах хэрэглэгч классаар гүйцэтгэгдэнэ. Зураг 2.3 -д дээрх дарааллыг агуулсан хэрэглэгчийн бүртгэлийн мэдээлэл зохион байгуулах шинжилгээний дарааллын диаграммыг харуулж байна.

Figures/chapter2/burtgel-zh-1.pdf

ЗУРАГ 2.3: Хэрэглэгчийн бүртгэлийн мэдээлэл зохион байгуулах шинжилгээний дарааллын диаграмм

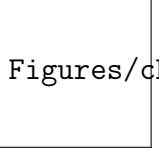
Хэрэглэгч системд шинээр бүртгэл үүсгэх хүсэлтийг бүртгэл класс хүлээн авч системд бүртгэлтэй эсэхийг шалгаад хэрэв бүртгэлгүй бол бүртгэлийн процессыг дуусгаад хэрэглэгч классаар дамжуулан шинэ хэрэглэгчийг системд бүртгэх хүсэлтийг явуулна. Дээрх процессыг Зураг 2.4 дээрх шинжилгээний дарааллын диаграммд дүрсэлсэн.



Figures/chapter2/burtguuleh-1.pdf

ЗУРАГ 2.4: Бүртгэл үүсгэх шинжилгээний дарааллын диаграмм

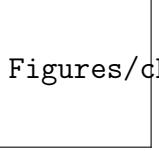
Зураг 2.5 -н хадгалагдсан файлуудын мэдээллийг зохион байгуулах шинжилгээний дарааллын диаграммд үзүүлсний дагуу системд бүртгэлтэй хэрэглэгч дээр бүртгэгдсэн байгаа объект илрүүлэлт хийсэн файлын мэдээллийг хэрэглэгчид харуулах болон устгах процессыг хэрэглэгчийн файл класс гүйцэтгэнэ.



Figures/chapter2/file-zh-1.pdf

ЗУРАГ 2.5: Хадгалагдсан файлуудын мэдээллийг зохион байгуулах шинжилгээний дарааллын диаграмм

Зураг 2.6 -д үзүүлсний дагуу хэрэглэгчийн хуулсан файлыг үр дүнгийн боловсруулалт класс локал сервер дээрээ хүлээн авч, объект илрүүлэлт хийхэд бэлддэг.



Figures/chapter2/fileHuulah-1.pdf

ЗУРАГ 2.6: Файл системд хуулах шинжилгээний дарааллын диаграмм

Объект илрүүлэлт хийхийн тулд хэрэглэгч үр дүнгийн боловсруулалт класст файлын мэдээллийг явуулна, дараа нь тус файлд объект илрүүлэлт класс объект илрүүлэлт хийгээд үр дүнг буцаана. Үр дүнгийн боловсруулалт класс текст болон аудио мэдээллийг боловсруулаад, бүртгэлтэй хэрэглэгчийн хүсэлтийг эсэхийг хэрэглэгч классаар дамжуулан шалгаад, хэрэв тийм бол хэрэглэгчийн файл класс руу объект илрүүлэлт хийсэн файлын мэдээллийг системд нэмэх хүсэлтийг илгээгээд үр дүнг хэрэглэгчид буцаана, бүртгэлгүй хэрэглэгчийн хүсэлтийг илгээгээд үр дүнг шууд буцаана. Зураг 2.7 нь

дээрх процессыг агуулсан объект илрүүлэлт хийх шинжилгээний дарааллын диаграммыг харуулж байна.

Figures/chapter2/objectDetection-1.pdf

ЗУРАГ 2.7: Объект илрүүлэлт хийх шинжилгээний дарааллын диаграмм

## 2.9 Үйл ажиллагааны диаграмм

Объект илрүүлэлт хийх веб системийн юзкейс диаграммд тодорхойлогдсон юзкейсүүдийн эхлэлээс төгсгөл хүртэлх үйлдлүүдийг дүрсэлсэн үйл ажиллагааны диаграммуудыг доор харуулав.

Хэрэглэгч бүртгэлийн мэдээллээ зохион байгуулах үйл ажиллагааг эхлэхийн тулд эхлээд системд нэвтэрсэн байна. Хэрэглэгч мэдээлэл шинэчлэх хэсгийг сонгосноор өөрийн бүртгэлийн мэдээллээ харна. Хэрэглэгч шинэчлэхийг хүссэн мэдээллээ шинэчилж оруулаад шинэчлэх хүсэлт илгээнэ. Систем хүсэлтийг хүлээн аваад мэдээллийг шинэчлээд хэрэглэгчид бүртгэгдсэн эсэх мэдээллийг хүргэнэ, хэрэв бүртгэгдээгүй бол хэрэглэгч хуучин бүртгэлийн мэдээллээ дахин харснаар дээрх процесс давтагдаж болно. Дээрх процессыг Зураг 2.8 -н хэрэглэгчийн бүртгэлийн мэдээлэл зохион байгуулах үйл ажиллагааны диаграммаас харж болно.

Figures/chapter2/ua-burtgel-zh.pdf

ЗУРАГ 2.8: Хэрэглэгчийн бүртгэлийн мэдээлэл зохион байгуулах үйл ажиллагааны диаграмм

Зураг 2.9 -н үйл ажиллагааны диаграмм нь дараах процессыг дүрсэлнэ. Хэрэглэгч бүртгэлийн мэдээллээ бөглөж оруулаад системд бүртгэлтэй эсэхээ шалгуулна, бүртгэлтэй бол хэрэглэгчид бүртгэлтэй гэсэн мэдээллийг хүргээд үйл ажиллагаа дуусна, бүртгэлгүй бол систем хэрэглэгчийн имейл руу ОТР -г илгээж, баталгаажуулалтыг хүлээнэ. Баталгаажуулалт амжилттай бол бүртгэл үүсэж үйл ажиллагаа дуусна харин амжилтгүй бол хэрэглэгчээс код дахин илгээх эсэхийг лавлаж үйл ажиллагаа ОТР илгээх үйлдэл руу буцна.

Figures/chapter2/ua-burtguuleh.pdf

ЗУРАГ 2.9: Бүртгэл үүсгэх үйл ажиллагааны диаграмм

Хадгалагдсан файлуудын мэдээллийг зохион байгуулахын тулд хэрэглэгч нэвтэрсэн байна. Дараа нь мэдээлэл харах хэсгээс өөрт байгаа файлуудын мэдээллийг харна. Устгахыг хүссэн файлынхаа ард байрлах устгах товчийг дарна, систем хэрэглэгчийг устгах эсэхийг лавлаж асууна. Хэрэглэгч файлыг устгана гэдгийг баталгаажуулснаар файл устаж үйл ажиллагаа дуусна. Харин хэрэглэгч лавлах асуултад цуцлах хэсгийг сонгож файлыг устгахгүйгээр үйл ажиллагааг дуусгаж болно. Зураг 2.10 дээрх хадгалагдсан файлуудын мэдээллийг зохион байгуулах үйл ажиллагааны диаграмм дээрх процессыг дурсэлж байна.

Figures/chapter2/ua-file-zh.pdf

ЗУРАГ 2.10: Хадгалагдсан файлуудын мэдээллийг зохион байгуулах үйл ажиллагааны диаграмм

Зураг 2.11 -д үзүүлсэн файл системд хуулах үйл ажиллагааны диаграммын дагуу хэрэглэгчийн сонгосон файл эхлээд веб хуудсанд хуулагдана, дараа нь объект илрүүлэх хүсэлттэй хамт илгээгдэж ирснээр back-end сервер дээр хуулагдана.

Figures/chapter2/ua-fileHuulah.pdf

ЗУРАГ 2.11: Файл системд хуулах үйл ажиллагааны диаграмм

Объект илрүүлэлт хийх үйл ажиллагаа эхлэхийн өмнө файл системд хуулагдсан байна. Файлд эхлээд объект илрүүлэлт хийгээд сонгогдсон хэл дээр текст болон аудио мэдээллийг боловсруулна. Хэрэглэгч системд нэвтэрсэн эсэхийг шалгаад үнэн бол үр дүнг хэрэглэгчийн файлын санд хадгалаад дараа нь үр дүнг хэрэглэгчид буцааснаар үйл ажиллагаа дуусна. Хэрэв хэрэглэгч нэвтэрч ороогүй байвал үр дүнг шууд буцааснаар үйл ажиллагаа дуусгавар болно. Зураг 2.12 -д дараах процессыг үйл ажиллагааны диаграммаар дурсэлсэн.

Figures/chapter2/ua-objectDetection.pdf

ЗУРАГ 2.12: Объект илрүүлэлт хийх үйл ажиллагааны диаграмм

## 2.10 БҮЛГИЙН ДҮГНЭЛТ

Энэ бүлэгт системийн үйл ажиллагааг дэлгэрэнгүй тайлбарлаж, системийг ашиглах хэрэглэгчид болон шаардлагуудыг тодорхойлж, юзкейс, юзкейсийн тодорхойлолт болон шинжилгээний шатны класс, дараалал, үйл ажиллагааны диаграммуудыг гаргасан. Эхлээд системийг хэрэглэх хэрэглэгчдийг тодорхойлоод системд оролцогч тал бүрийн функцийн шаардлагыг тодорхойлж мөн системийн функцийн бус шаардлагыг ч тодорхойлсон. Системд оролцогчдын функцийн шаардлага дээр үндэслэн 2 тоглогчтой 5 юзкейс бүхий юзкейс диаграммыг байгуулж, юзкейс бүрд тодорхойлолт гаргасан. Юзкейсийг хэрэгжүүлж болох шинжилгээний шатны 6 классуудыг гаргаж тэдгээрийн холбоо хамаарлыг тодорхойлсон. Тодорхойлогдсон классууд болон юзкейсийн тодорхойлолтын дагуу дарааллын диаграмм болон үйл ажиллагааны диаграммуудыг гаргаж системийн үйл ажиллагаа, оролцогч талууд, процессын дарааллыг шинжилгээний түвшинд ойлгох боломжийг олгосон.

---

---

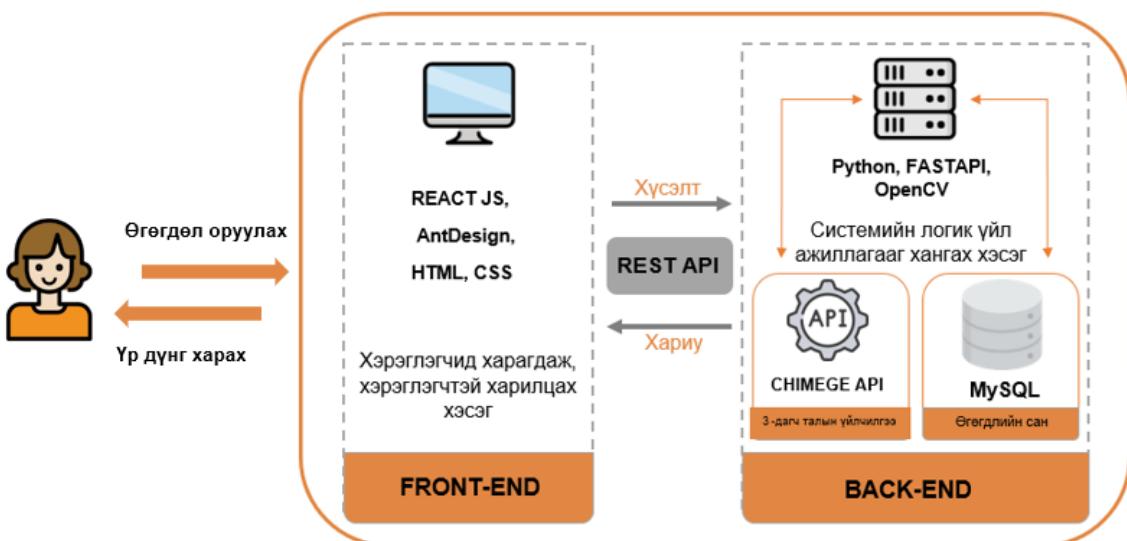
## БҮЛЭГ 3

---

... системийн зохиомж ба хөгжүүлэлт

### 3.1 Системийн архитектур

Зургаас объект илрүүлэлт хийдэг веб систем нь Back-end болон Front-end гэсэн үндсэн хоёр хэсгээс бүрдэнэ. Зураг 3.1 -д үзүүлсний дагуу Front-end хэсэг нь хэрэглэгчид харагдаж, хэрэглэгчтэй харилцах хэсэг бөгөөд хэрэглэгчийн оролтын мэдээллийг авч Back-end хэсгээс гаргасан сервисийг ашиглан Back-end хэсэг рүү хүсэлт явуулж, хариуг аван хэрэглэгчийг гаралтын мэдээллээр хангана. Back-end хэсгийн хувьд системийн логик үйл ажиллагааг хангаж ажилладаг. Хэрэглэгчийн нэвтрэлт, бүртгэл, объект илрүүлэлттэй холбоотой сервисүүдийг гаргаж, монгол хэл дээрх текстийг аудио руу хөрвүүлж өгөх 3-дагч талын сервис болох ChimegeAPI, хэрэглэгчийн мэдээлэл болон тэдгээрт хамааралтай объект илрүүлэлтэд ашигласан файлын мэдээллийг хадгалах MySQL өгөгдлийн сантай ажиллана.



ЗУРАГ 3.1: Системийн архитектур

### 3.2 Зохиомжийн шатны класс диаграмм

Объект илрүүлэлт хийдэг веб системийн зохиомжийн шатны класс диаграммд control төрлийн Хэрэглэгч, Үр дүнгийн боловсруулалт, Хэлний төрөл, Хэрэглэгчийн файл, Объект илрүүлэлт, Бүртгэл гэсэн 6-н класс ,entity төрлийн Хэрэглэгчийн файлын мэдээлэл, Хэрэглэгчийн мэдээлэл гэх 2 класс, boundary төрлийн Объект илрүүлэлтийн дэлгэц, Бүртгэлийн дэлгэц, ОТР шалгах дэлгэц, Хэрэглэгчийн файлын мэдээллийн дэлгэц, Хэрэглэгчийн мэдээллийн дэлгэц гэсэн 5-н класс буюу нийт 13

классын атрибут болон операторуудыг тодорхойлсон. Зураг 3.2 -д тэдгээр классуудын стерео төрлийг тодорхойлж, класс хоорондын холбоо хамаарлыг ассоцийши болов бүрдмэл холбоо хамаарлаар дүрсэлж, гарим класс хоорондын холбоо хамааралд ашиглалтын хараат байдлыг дүрсэлсэн.

Figures/chapter3/class\_diagram

ЗУРАГ 3.2: Класс диаграмм

### 3.3 Өгөгдлийн ерөнхий схем

Объект илрүүлэлт хийдэг веб системийн өгөгдлийн сан нь хэрэглэгч болон хэрэглэгчийн файл гэсэн хоёр хүснэгтийг агуулна. Хэрэглэгч хүснэгт нь системд бүртгүүлсэн хэрэглэгчийн овог нэр, имейл, нууц үг болон хэрэглэгчийн кодыг анхдагч түлхүүрээр хадгална. Хэрэглэгчийн файл хүснэгт нь файлын нэр, локал сервер дээр хадгалагдсан байршил, текст, аудио мэдээлэл, огноо, хэрэглэгчийн кодыг гадаад түлхүүрээр, файлын кодыг анхдагч түлхүүрээр хадгална. Зураг 3.3 -д системийн хөгжүүлэлийн хүрээнд үүсгэсэн өгөгдлийн сангийн "Хэрэглэгч", "Хэрэглэгчийн файл" гэсэн хүснэгтүүд, тэдгээрийн атрибутууд болон тэдгээр хүснэгтүүд "1 : олон" харьцаагаар дурслэгдсэнийг харуулж байна.

Figures/chapter3/erd.PNG

ЗУРАГ 3.3: Өгөгдлийн ерөнхий схем

### 3.4 Системийн прототип

Системийн үндсэн хуудас, бүртгүүлэх болон нэвтрэх хуудас, бүртгэлтэй хэрэглэгчийн мэдээллээ шинэчлэх болон файлын мэдээллийдээ зохион байгуулах хуудаснуудын прототип загварыг дурслэн харуулна.

Зураг 3.4 -д зургаас объект илрүүлэх вебийн хэрэглэгчийн бүртгэлээр ороогүй байх үеийн объект илрүүлэлт хийх нүүр хуудас харагдаж байна.

 Figures/chapter3/Frame 1.pdf

ЗУРАГ 3.4: Зургаас объект илрүүлэлт хийх хуудасны прототип

Зураг 3.4 -д үзүүлсэн хэрэглэгчийн бүртгэлээр нэвтрээгүй байх үеийн нүүр хуудасны толгой хэсэгт байрлах "Бүртгүүлэх" лабел -г дарснаар тус хуудас гарч ирнэ. Зураг 3.5 -д хэрэглэгчийн бүртгэлийн мэдээллийг авч бүртгэл үүсгэх хуудсыг харуулж байна.

 Figures/chapter3/Frame 2.pdf

ЗУРАГ 3.5: Бүртгэл үүсгэх хуудасны прототип

Нүүр хуудасны толгой хэсэгт байрлах "Нэвтрэх" лабел -г дарснаар тус хуудас гарч ирнэ. Зураг 3.6 -д хэрэглэгчийн имейл хаяг, нууц үгээр вебд нэвтрэх хуудсыг харуулж байна. Тус хуудас дээрх "Нууц үг сэргээх" лабел нь нууц үг сэргээх хуудас руу чиглүүлнэ.

 Figures/chapter3/Frame 3.pdf

ЗУРАГ 3.6: Нэвтрэх хуудасны прототип

Хэрэглэгчийн бүртгэлээр нэвтрээд хуудасны толгой хэсэгт байрлах жагсаалтыг илэрхийлсэн icon дээр дарснаар Зураг 3.7 -д үзүүлсэн хуудас харагдана. Тус зурагт хэрэглэгчид бүртгэлтэй объект илрүүлэлтэд ашигласан зураг, түүний үр дүнгийн текст болон аудио файлууд харагдана. Жагсаалт дахь файл бүрийн ард байрлах устгах icon нь харгалзах зургийн мэдээллээ устгах бол "Бүгдийг устгах" лабел -тэй icon нь бүх бүртгэлтэй файлыг устгана.

 Figures/chapter3/Frame 4.pdf

ЗУРАГ 3.7: Бүртгэлтэй файлын мэдээлэл харах хуудасны прототип

Хэрэглэгчийн бүртгэлээр нэвтрээд хуудасны толгой хэсэгт байрлах хэрэглэгчийг илэрхийлсэн icon дээр дарснаар Зураг 3.8 -д үзүүлсэн дэлгэц харагдана. Дэлгэцэд "Бүртгэлийн мэдээлэл шинэчлэх", "Гарах" гэсэн сонголтуудыг идэвхжүүлж харуулна.

 Figures/chapter3/Frame 5.pdf

ЗУРАГ 3.8: Хэрэглэгчийн сонголт харуулах дэлгэцийн прототип

Зураг 3.8 -д үзүүлсэн дэлгэц дээрх сонголтуудаас "үртгэлийн мэдээлэл шинэчлэх" онголтыг сонгосноор Зураг 3.9 -д үзүүлсэн хуудас гарч ирнэ. Тус зурагт үзүүлсэн хуудас нь

хэрэглэгчийн бүртгэлийн мэдээллийг харуулаад, шинэчилсэн мэдээллийг аваад хадгална.

 Figures/chapter3/Frame 6.pdf

ЗУРАГ 3.9: Бүртгэлийн мэдээлэл шинэчлэх хуудасны прототип

Зураг 3.10 -д бүртгэлийн мэдээлэл амжилттай шинэчлэгдсэний дараах pop-up хэлбэрийн мэдээлэл хэрэглэгчид хэрхэн харагдахыг харуулж байна.

 Figures/chapter3/Frame 7.pdf

ЗУРАГ 3.10: Бүртгэлийн мэдээлэл амжилттай шинэчлэгдсэнийг харуулах дэлгэцийн прототип

Зураг 3.11 -д бүртгэлийн мэдээлэл оруулсны дараа хэрэглэгчийн бүртгэлдээ ашигласан имейл хаяг руу илгээгдэх бүртгэл баталгаажуулах ОТР оруулах хуудсыг харуулж байна.

 Figures/chapter3/Frame 8.pdf

ЗУРАГ 3.11: ОТР оруулах хуудасны прототип

Хэрэглэгч ОТР -г зөв оруулснаар бүртгэл баталгаажна. Зураг 3.12 -д бүртгэл амжилттай үүссэнийг харуулах pop-up мэдээлэл хэрэглэгчид хэрхэн харагдахыг харуулж байна.

 Figures/chapter3/Frame 9.pdf

ЗУРАГ 3.12: Бүртгэл амжилттай үүссэнийг харуулах дэлгэцийн прототип

Зураг 3.13 -д хэрэглэгч буруу ОТР оруулсны дараах дахин ОТР авах эсвэл бүртгэлийг цуцлах сонголтуудыг агуулсан pop-up мэдээлэл хэрхэн харагдахыг харуулсан.

 Figures/chapter3/Frame 10.pdf

ЗУРАГ 3.13: Код дахин илгээх эсэхийг лавлах дэлгэцийн прототип

### 3.5 Хөгжүүлсэн системийн интерфейс

Хөгжүүлэгдсэн системийн объект илрүүлэлт хийх, нэвтрэх, бүртгүүлэх, ОТР шалгах хуудаснууд болон хэрэглэгчийн бүртгэлээр нэвтэрч орсон үеийн объект илрүүлэлт хийх хуудас, өөрт бүртгэлтэй зургуудын жагсаалт харах хуудасны интерфейсийг энэ хэсэгт тайлбарлана.

Зураг 3.14 -д харуулсан интерфейстэй хуудас нь хэрэглэгчийг веб хуудас руу хэрэглэгчийн бүртгэлгүйгээр хандах үед харагдана. Хуудасны зураг оруулах хэсэгт зургийг оруулснаар зургийн нэр доод хэсэгт харагдана. Англи болон монгол гэсэн сонголтуудаас хэлний сонголтоо хийгээд "Илрүүлэлт хийх" товчийг дарснаар текст болон аудио мэдээлэл хэсгүүдэд үр дүн харагдана.

 Figures/chapter3/home.pdf

ЗУРАГ 3.14: Объект илрүүлэлт хийх хуудасны интерфейс

Зураг 3.15 -д системийн нэвтрэх хуудасны интерфейсийг харуулсан. Бүртгэлтэй хэрэглэгч имейл хаяг болон нууц үгээ оруулаад "Нэвтрэх" товчийг дарж системд нэвтэрнэ. Хэрэглэгчийн оруулсан имейл хаяг бүртгэлгүй эсвэл нууц үт буруу тохиолдолд алдааны мессежийг хэрэглэгчид харуулна. Хэрэв бүртгэлгүй бол "Одоо бүртгүүлэх" гэсэн лабелийг дарж бүртгүүлэх хуудас руу очиж болно.

 Figures/chapter3/login.pdf

ЗУРАГ 3.15: Нэвтрэх хуудасны интерфейс

Зураг 3.16 -д хэрэглэгчийн бүртгэл үүсгэх хуудасны интерфейс харагдаж байна. Хэрэглэгч бүртгэлийн мэдээллээ оруулаад "Бүртгүүлэх" товч дарна. Хэрэв хэрэглэгчийн бүртгүүлсэн имейл хаяг бүртгэлтэй байвал "Бүртгэлтэй имейл байна" гэсэн алдааны мессежийг хэрэглэгчид харуулна, имейл хаяг бүртгэлгүй байвал хэрэглэгчийн имейл хаяг руу ОТР явуулаад ОТР шалгах хуудас руу чиглүүлнэ.

 Figures/chapter3/signup.pdf

ЗУРАГ 3.16: Бүртгэл үүсгэх хуудасны интерфейс

Зураг 3.17 -д хэрэглэгч рүү илгээсэн ОТР -г шалгах хуудасны интерфейсийг харуулж байна. Хэрэглэгч ОТР -г оруулаад "ОТР шалгах" товчийг дарна. Хэрэв ОТР зөв байвал хэрэглэгчийн бүртгэл амжилттай үүссэнийг хэрэглэгчид мессежээр мэдэгдээд

нэвтрэх хуудас руу чиглүүлнэ. Буруу бол "Таны оруулсан OTP буруу байна" гэсэн мессежийг хэрэглэгчид хүргээд хуудас солигдохгүй. Хуудасны "Арилгах" товч оруулсан OTP -г устгадаг, "Бүртгэл цуцлах" товч таны оруулсан мэдээллүүдийг устгаж бүртгэл үүсгэх процессыг цуцлаад бүртгүүлэх хуудас руу чиглүүлнэ.

Figures/chapter3/otp.pdf

ЗУРАГ 3.17: OTP шалгах хуудасны интерфейс

Зураг 3.18 -д системд хэрэглэгчийн бүртгэлээр нэвтэрсэн байх үеийн объект илрүүлэлт хийх хуудасны интерфейсийг харуулж байна. Хуудасы толгой хэсэгт жагсаалт болон хэрэглэгчийг дүрсэлсэн icon -ууд харагдана. Тус хуудаснаас жагсаалтыг дүрсэлсэн icon дээр дарвал хэрэглэгчид бүртгэлтэй, объект илрүүлэлт илрүүлэлт хийсэн зургуудын жагсаалт харагдана. Хэрэглэгчийг дүрсэлсэн icon дээр дарвал "Бүртгэл шинэчлэх" болон "Гарах" гэсэн сонголтууд харагдана.

Figures/chapter3/home-with-user.pdf

ЗУРАГ 3.18: Хэрэглэгчийн бүртгэлээр орсон үеийн объект илрүүлэлт хийх хуудас

Зураг 3.19 -д хэрэглэгчид бүртгэлтэй зургуудын жагсаалтыг харуулдаг хуудасны интерфейсийг дүрсэлсэн. Жагсаалтын нэг item бүрд объект илрүүлэлт хийсэн зураг, текст болон аудио үр дүн, объект илрүүлэлт хийсэн огноо болон "Устгах" icon байна. Хэрэглэгч "Устгах" icon -г дарснаар тус icon -д харгалзах зургийг устгах боломжтой.

Figures/chapter3/list-items.pdf

ЗУРАГ 3.19: Бүртгэлтэй зургуудын жагсаалт харуулах интерфейс

## 3.6 Системд хийгдсэн тест

## 3.7 Системийн нэвтрүүлэлт

## 3.8 Бүлгийн дүгнэлт

Энэ бүлэгт ерөнхийдөө системийн зохиомжийг гаргасан. Системийн архитектур, класс диаграмм, дарааллын диаграмм, өгөгдлийн ерөнхий схем, системийн прототип

загвар болон хөгжүүлэгдсэн системийн интерфейсийг багтаасан. Системийн архитектурыг front-end болон back-end гэсэн үндсэн 2 хэсэгт хувааж, тэдгээр хэсгүүд дээр хийгдэх үйл ажиллагаануудыг бүдүүвч байдлаар дүрсэлсэн. Ингэхдээ front-end хэсгийг хэрэглэгчид харагдаж, хэрэглэгчтэй харилцах хэсэг, back-end хэсгийг системийн логик үйл ажиллагааг хангах хэсэг гэж тодорхойлоод REST API ашиглан тэдгээрийн дунд холболт үүсгэхээр дүрсэлсэн. Класс диаграмм дээр шинжилгээний шатны control төрлийн 6-н класс дээр нэмээд boundary төрлийн 5-н класс, entity төрлийн 2 классыг гаргаж тэдгээрийн харьцаа хамаарлыг тодорхойлсон. Шинээр тодорхойлсон классуудыг оролцуулан илүү дэлгэрэнгүй, ойлгомжтой дарааллын диаграммуудыг системийн юзкейс болгон дээр үүсгэсэн. Θгөгдлийн ерөнхий схемийг зохиомжлоходоо Хэрэглэгч, Хэрэглэгчийн файл гэсэн 2 хүснэгтүүдийг үүсгэн аттрибутуудыг тодорхойлоод, тэдгээрийг (1 : олон) харьцаагаар тодорхойлсон. Хөгжүүлэлт эхлэхээс өмнөх системийн прототип загварыг boundary төрлийн классуудын хувьд гаргасан. Мөн хөгжүүлэлтийн дараах системийн интерфейсийг хөгжүүлэгдсэн вебийн хуудас бүрээр оруулсан.

## Ерөнхий дүгнэлт

“Зургаас объект илрүүлэлт хийх веб хөгжүүлэх нь” сэдэвтэй дипломын ажлын зорилго нь зурагт дурслэгдэж буй объектуудыг таних, тэдгээр объектын нэрсийг тодорхойлох, байршилыг дурслэн харуулах, дурслэгдсэн объектуудын нэрийг текст болон монгол, англи хэл дээрх аудио хэлбэр рүү хөрвүүлэх, мөн хэрэглэгчид өмнө нь хөрвүүлсэн зураг, тухайн зургийн текст, аудио мэдээллийг хадгалах чадвартай веб сайтыг хөгжүүлэх юм. Тус веб сайтыг хөгжүүлснээр хэрэглэгчид зурагт агуулагдаж буй объектуудыг ялгаж, таньж мэдэх, илэрсэн объектуудыг нэрээр нь ангилан тоо-лох, зурагт дурслэгдсэн жижиг объектуудыг олж харах, харааны бэрхшээлтэй иргэд илэрсэн объектуудын мэдээллийг аудио хэлбэрээр сонссоноор зургийн агуулгыг тө-сөөлөх гэх мэт боломжийг олгоно.

Системийг хөгжүүлэхийн тулд объект илрүүлэлт гэж юу болох талаар судалж, хэ-рэглэгч болон системийн шаардлагуудыг тодорхойлж, системийг хөгжүүлэхэд шаард-лагатай ЮМЛ -н диаграммуудыг гаргасан. Системийн архитектур нь хэрэглэгчтэй харилцах хэсэг болох front-end болон системийн логик үйл ажиллагааг хангах хэ-сэг болох back-end гэсэн 2 хэсгээс бүрдэж хоорондоо REST API ашиглан холбогддог байхаар зохиомжилсон. MobileNet SSD V3 архитектурын хөлдөөсөн моделийг авч OpenCV -н DetectionModel классаар объект илрүүлэлт хийх сүлжээ үүсгэж, түүгээ-рээ дамжуулан зургаас объект илрүүлэлтийг гүйцэтгэж байгаа юм. Текст мэдээллийг аудио руу хөрвүүлэхдээ Python gTTS сан, Chimege API -г ашигласан. Системийн хэ-рэглэгчтэй харилцах хэсгийг ReactJS фреймворк ашиглан хөгжүүлсэн.

Энэхүү дипломын ажил нь зургаас объект илрүүлэлт хийх веб хөгжүүлэхэд ашиг-ласан технологиуд, хэд хэдэн объект илрүүлэлтийн архитектурын талаарх мэдээл-лийг агуулсан бичиг баримт ба зургаас объект илрүүлэлт хийх веб сайтыг багтаасан болно.

---

Чухал кодын хэсгүүд

## Хэрэглэгчийн бүртгэл үүсгэх кодын хэсэг

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22            if M is None:
23                M = np.copy(VT)
24            else:
25                M = np.concatenate((M, VT), 1)
26
27            VT = np.zeros((n*m,1), int)
28
29    return M
```