

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

Хүрэлбаатар Лүндэгцэрэн

Чимэгэ гар
(Chimege keyboard)

Мэдээллийн технологи (D061303)
Үйлдвэрлэлийн дадлагын тайлан

Улаанбаатар

2020 оны 03 сар

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

Чимэгэ гар
(Chimege keyboard)

Мэдээллийн технологи (D061303)
Үйлдвэрлэлийн дадлагын тайлан

Удирдагч:	_____	Ц.Бат-оргил
Зөвлөх багш:	_____	Др. Б.Хуягбаатар
Гүйцэтгэсэн:	_____	Х.Лүндэгцэрэн (16B1SEAS0389)

Улаанбаатар

2020 оны 03 сар

ГАРЧИГ

1. СИСТЕМИЙН ТАНИЛЦУУЛГА	1
1.1 Компани танилцуулга	1
1.2 Ямар үйлчилгээ үзүүлдэг вэ?	1
1.3 Ямар систем дээр голчлон төвлөрдөг вэ?	2
1.4 Системийн ямар хэсэг дээр төвлөрч байгаа вэ?	2
1.5 Системээс хамаарсан ямар сайжруулалт хийх вэ?	2
2. ИЖИЛ ТӨСТЭЙ СИСТЕМИЙН СУДАЛГАА	4
2.1 Виртуал гаруудын үйл ажиллагааны судалгаа	4
2.2 Үйл ажиллагааны харьцуулалт	6
3. СИСТЕМИЙН ШААДЛАГА	8
3.1 Хэрэглэгчийн шаардлага	8
3.2 Use case diagram (Ажлын явцын диаграм)	10
4. СИСТЕМИЙН ШИНЖИЛГЭЭ ЗОХИОМЖ	11
4.1 Өгөгдлийн сангийн диаграмм	11
4.2 Өгөгдлийн сангийн холбоост диаграмм	12
4.3 Өгөгдлийн толь бичиг	12
4.4 Дарааллын диаграмм (Sequence diagram)	16
5. ТЕХНОЛОГИЙН СУДАЛГАА	18
5.1 Үндсэн үгнүүдийн орчуулга	18
5.2 Дэлгэцэн дээрх оролтын арга	18
5.3 Оролтын арга гэж юу вэ?	18
5.4 Оролтын арга эдиторыг удирдах үндсэн XML шинж чанарууд	21
5.5 EditText удирдлага бүрийн оролтын төрлийг тодорхойлох	22

5.6	Хэмжээг өөрчлөх горим болон бусад цонхны онцлог чанаруудыг идэвхжүүлэх	23
5.7	Оролтын аргын эдитор -г хянах API	24
5.8	Оролтын арга үүсгэх	24
5.9	Manifest доторх IME-ийн бүрэлдэхүүнийг зарлах	26
6.	ХЭРЭГЖҮҮЛЭЛТ	27
6.1	Системийг бичихэд ашигласан технологи	27
6.2	Ажлын явцын диаграммын тохиолдол бүрийн хэрэглэгчийн интерфейс	40
7.	ДҮГНЭЛТ	43
7.1	Үр дүн.....	43
7.2	Үр дүнгийн тайлан	43
	ХАВСРАЛТ	45
A.	ШИНЖИЛГЭЭ ЗОХИОМЖ	46
A.1	Дадлагын ажлын удирдагчийн тодорхойлолт	46
B.	КОДЫН ХЭРЭГЖҮҮЛЭЛТ	47

ЗУРГИЙН ЖАГСААЛТ

3.1	Чимэгэ виртуал гарны ажлын явцын диаграмыг үүсгэв.	10
4.1	Чимэгэ системийн бүтээгдэхүүн хэрэглэгчийн өгөгдлийн сан.	11
4.2	ER диаграмаас гарч ирсэн холбоост диаграмм.	12
4.3	Яриаг бичвэр болгох үйлчилгээний дарааллын диаграмм.	17
5.1	Ухаалаг утасны үндсэн зураг.	19
5.2	Виртуал гар дэлгэцэн дээр идэвхжсэн байдал.	19
5.3	Зөвхөн өөрчлөх талбар дээр төвлөрсөн байдал.	20
5.4	Утас хөндлөн үед виртуал гар дэлгэцэнд идэвхжих.	21
5.5	Уян хатан гар нь хэрэглэгчийн оруулсан текстэд тааруулан хамгийн сайн гарын байрлалыг ашиглахын тулд боломжтой оролтын зөв төрлийг ашиглах нь чухал юм.	23
5.6	Автомат тохиргоо.	24
5.7	IME-ийн амьдралын мөчлөг.	25
5.8	IME-ийн амьдралын мөчлөг.	26
6.1	Монгол кирилл болон монгол бичгийн виртуал гарыг тодорхойлсон файлын зураг	29
6.2	Монгол кирилл бичгийн виртуал гар дээрх үсэг болон тэмдэгтүүдийг зааж өгсөн файлын зураг.	30
6.3	Монгол кирилл болон монгол бичгийн виртуал гарыг тодорхойлсон файлын зураг	31
6.4	Жишээ нь эхний мөрийн кирилл үсгийн байрлалыг харуулав. Өмнө хэлж байснаар тухайн хэлний YAML файлыг үүсгэж, KBDgen хэрэгсэл ашиглан виртуал гар бий болгоноор гарын байрлал XML файл хэлбэрээр ингэж кодлогдож байна.	32

6.5	Энэ мөрөнд хэл солих, зай авах, тэмдэгт рүү шилжих, догол мөр шилжих гэсэн үндсэн товчууд дээр нэмээд дуу бичлэгийн товчийг нэмэв. Нэмсэн хэсгийнхээ xml бичиглэлийг танилцуулбал <code>keyWidth = "10%"</code> энэ код нь утасны дэлгэцүүдээс хамааран тухайн дэлгэцийн урт хэсгийн 10%-ийг эзэлнэ. Мөн <code>keyStyle="voiceKeyStyle"</code> код нь дуу бичлэгийн зураг, өнгийг агуулдаг.....	33
6.6	Дуу бичлэгийн зураг, харагдах хэсэг нь бүх товчны style-уудыг багтаасан хэсэгт байрлах ба ашиглах газраа нэмж өгнө	33
6.7	Үсгээс эхлэн мөрийг тоолбол тухайн гар нь 4-н мөртэй ба 4-р мөрөнд дуу бичлэгийн товч харагдаж байна.	34
6.8	Дуу бичих зөвшөөрөл байгаа эсэхийг шалгаж буй зураг.	34
6.9	Чимэгэ виртуал гарын Manifest дотор интернэт болон дуу бичлэг бичих зөвшөөрлийг бичсэн байна.	35
6.10	Дуу бичлэг бичих зөвшөөрөл хүсэх кодын хэсэг. Тухайн кодын хэсгийг бичиж өгөөгүй тохиолдолд яриаг бичвэрт хөрвүүлэх үйлчилгээг ашиглахын тулд хэрэглэгч өөрөө утасныхаа тохиргоо хэсэг рүү орж чимэгэ виртуал гарын дуу бичлэг бичих зөвшөөрлийг механик байдлаар хийж өгөх юм.	35
6.11	17-р зурган дээрх кодыг бичиж өгснөөр дуу бичлэгийн товчин дээр дарахад хэрэглэгчээс дуу бичих эрхийг зөвшөөрлийг хүссэн диалог гарч ирнэ.	36
6.12	Интернэт байхгүй тохиолдолд мэдэгдэл ашигласан ба тухайн мэдэгдлийг хэрхэн гаргаж байгааг тусдаа дэлгэрэнгүй тайлбарлана.	37
6.13	Firestore дахь токены бүтэц	37

6.14	Wave аудионы эхний 44-н байтын мэдээлэл.	38
6.15	Хөрвүүлэлт хийх хүсэлтийг хийж буй кодын хэсэг.	39
6.16	Монгол том үсэгтэй зураг.	40
6.17	Монгол жижиг үсэгтэй зураг.	41
6.18	Тоо болон тэмдэгтийн зураг.	41
6.19	Эможи зураг.	42
6.20	Яриа бичиж буй үйлдлийн зураг.	42
7.1	Яриа бичиж буй үйлдлийн зураг.	43
A.1	Дадлагын удирдагчийн тодорхойлолт.	46

ХҮСНЭГТИЙН ЖАГСААЛТ

2.1	Харьцуулалтын хүснэгт. Тайлбар: ' + ' - тэмдэгт нь тухайн үйлчилгээ байгааг төлөөлнө.	6
4.1	Users хүснэгт.	13
4.2	Products хүснэгт.	14
4.3	UsageHistory хүснэгт.	14
4.4	SubscriptionType хүснэгт.	15
4.5	OrderHistory хүснэгт.	16

1. СИСТЕМИЙН ТАНИЛЦУУЛГА

1.1 Компани танилцуулга

Болорсофт ХХК нь судалгаанд суурилсан хөгжүүлэлт (R&D), найдвартай, чанартай, анхдагч байдлыг бий болгосон программ хангамжийн бүтээгдэхүүнүүд үйлдвэрлэхээр зорин анх 2008 онд байгуулагдсан. 2008-2013 оныг хүртэл мэдээлэл технологийн өрсөлдөөний давлагаанд чанартай байдлыг эрхэмлэн зогсож чадсан ба бодит мэдээлэл дээр үндэслэн программ хангамжаа хөгжүүлдэг билээ. Тиймээс тухайн компанийн бүтээгдэхүүнүүд салбартаа үргэлж анхдагч болдог бөгөөд зохиогчийн эрхээр хамгаалагдсан байдаг. Болорсофтын хамт олон Монгол улс болон Герман улсад мэргэшсэн чадварлаг, ажлаа бүрэн мэддэг залуусаас бүрдэхээс гадна салбартаа тэргүүлэх программ хангамжийн архитектурч зөвлөхөөр ажиллаж байна. Энэ компани нь олон улсын түвшинд үйл ажиллагаа явуулдаг бөгөөд АНУ болон Герман улсад төлөөлөгчид нь ажилладаг тул дэлхий нийтийн мэдээлэл технологийн хөгжлөөс огтхон ч хоцордоггүй.

1.2 Ямар үйлчилгээ үзүүлдэг вэ?

Болорсофт компани нь төсөл ба энтерфрайс систем, веб, программын нутагшуулалт, хэл шинжлэлтэй холбоотой томоохон ажлуудыг хийж гүйцэтгэж байна. Тухайн компанийн бүтээгдэхүүнүүд болон хийж байсан ажлуудыг дурдвал үг зүйн судалгаан дээр суурилсан “Болор” алдаа алдаа хянах программ, Монгол улсын ерөнхийлөгчийн цахим хуудсыг босоо монгол бичгээр урласан, Болор толь буюу онлайн толь бичгийн систем, цахим хэл шинжлэлийн олон жилийн судалгааныхаа үр дүнг мэдээллийн технологийн хөгжлийн чиг хандлага болж буй машин сургалтын технологитой хослуулан “Eduge.mn” бүтээсэн, “Кудос” медиа мониторингийн систем хөгжүүлсэн, хиймэл оюун ухаан ашиглан монгол

болон бусад цөөн хэдэн улсын дуу яриаг боловсруулах технологи болох “Чимэгэ” системийг хөгжүүлээд байна.

1.3 Ямар систем дээр голчлон төвлөрдөг вэ?

Чимэгэ систем нь хиймэл оюун ашиглаж, Монгол болон бусад цөөн үндэстний дуу яриа боловсруулах технологийг хүнтэй дүйцэх түвшинд хүргэж хөгжүүлэхээр зорьж байна. Өнөөдөр дэлхий даяар хэлний боловсруулалт хүчээ аван хөгжиж, техник хэрэгслүүдтэй хүний хэлээр харилцах хэмжээнд хүрч байна. Робот, ухаалаг байшин, машин гээд бүхий л техник хэрэгслүүдийг яриагаар удирддаг болжээ. Иймд одоогоор монгол хэлний хувьд яриаг бичвэр, бичвэрийг яриа болгох технологи дээр төвлөрөн хөгжүүлж байна. Одоогоор бид Монгол хэл дээр төвлөрч танилтыг 96%-с багагүй хэмжээнд хүргээд байна.

1.4 Системийн ямар хэсэг дээр төвлөрч байгаа вэ?

Монгол хэлний хувьд яриаг бичвэр болгох технологи нь нэлээн ярвигтай асуудал байсан. Тэгвэл энэ компани нь тус системийн танилтыг хувийг 90% давууллаа. Цааш системийнхээ танилтыг улам сайжруулах ажлууд хийж байгаа ба танилт 1% сайжрахад хэрэглэгчийн бүтээмж багадаа 10% дээшилнэ гэж үзэж байна

1.5 Системээс хамаарсан ямар сайжруулалт хийх вэ?

Чимэгэ систем нь бичвэрийг компьютерт уншуулж яриулах технологи, яриаг компьютерт уншуулж бичвэр болгох технологи, англиас монгол руу орчуулах, монголоос англи руу орчуулах гэх мэт үйл ажиллагаагаараа ялгаатай дэд хэсгүүдтэй. Эдгээр технологиудыг ашиглан хэрэглээнд нэвтрүүлэх нь цаашдын зорилго болоод байна. Тус технологийн хэрэглээнээс дурдвал автомат транскрипци буюу дуу ярианы болон видео бичлэгийг автоматаар текст рүү хөрвүүлэх, аудио хайлт, дуудлагын төв, автомат хариулагч ба ярианаас бичвэрт хөрвүүлэх технологийг ашигласан утасны монгол виртуал гар гэх зэрэг

бүтээгдэхүүнүүдийг хийж байна. Миний бие дадлагын ажлын хүрээнд андройд үйлдлийн системтэй утсанд зориулан монгол хэл ба яриаг бичвэрт хөрвүүлэх функцтэй виртуал гарыг хийж гүйцэтгэн, сайжруулахад төвлөрөн ажилсан.

2. ИЖИЛ ТӨСТЭЙ СИСТЕМИЙН СУДАЛГАА

Андройд 1.5 хувилбараас эхлэн Андройд платформ нь оролтын аргын фреймворкийг санал болгож эхэлснээр тухайн утсаны дэлгэцтэй нийцэн ажиллах виртуал гаруудын кодчиллыг томоохон программ хангамжийн компаниуд болон программистууд хэрэглэгчдэд хялбар бөгөөд ойлгомжтой виртуал гаруудыг гаргах болсон. Тухайн сэдвийн хүрээнд ижил төстэй системийн судалгаагаар Google-ээс гаргасан Gboard, Айкон.мн(АЙКОН), ЮБИнфо.мн (ЮБИНФО), Чимэгэ системс ХХК-ы виртуал гар буюу дадлагын ажлаараа хийсэн виртуал гар гэх 4-н төрлийн виртуал гаран дээр судалгаа хийсэн.

2.1 Виртуал гаруудын үйл ажиллагааны судалгаа

2.1.1 *Gboard виртуал гар*

Gboard бол гар утасны android болон ios үйлдлийн системүүдэд Google-ээс зориулан гаргасан виртуал гар юм. Энэ гар нь анх ios дээр 2016 оны 5-р сард гаргасан бөгөөд үүний дараа 2016 оны 12-р сард Android дээр гаргасан ба одоог хүртэл шинэчлэлүүдийг хийсээр байна. Энэ виртуал гарын хувьд Google-ийн хайлтын системийг тухайн гарандаа нэмж өгсөн. Ингэснээр тухайн виртуал гар нь өөр дээрээ хайлт хийх боломжтой болсоноос шалтгаалан GIF болон Emoji зургуудыг хялбар хайх, хайлтын утгаас хамааран дараагийн үгийг санал болгодог таамаглах системүүд нэмэгдсэн. Мөн шинэчлэлт хийгдсэнээр тухайн бичвэрээс хамааран GIF зураг санал болгодог, виртуал гарын арын хэсгийн дэвсгэр зургийг өөрийн хүссэн зургаараа өөрчлөх, дуут бичвэр гэх мэт функцуудыг дэмждэг болсон. Одоогоор тухайн виртуал гар нь 100 гаруй хэлийн бичвэрийг дэмждэг болоод байна. Google компани нь өөрсдийн хийсэн виртуал гаран дээрээ Google орчуулагч болон хэл боловсруулалтын хувьд яриаг бичвэрт хөрвүүлэх гэх мэт зүйлүүдийг нэмж өгснөөр бусад виртуал гаруудаас ялгарч байгаа юм.

2.1.2 Айкон виртуал гар

Айкон.мн-ы хамт олон 2014 оны 9-р сарын 21-нд зөвхөн iPhone, iPad төхөөрөмжүүд дээр ажиллах буюу ios үйлдлийн систем дэмжсэн виртуал гарыг гаргасан. Тухайн виртуал гар нь зөвхөн монгол хэлийг дэмжсэн ба үг урьдчилан санал болгох системийг хөгжүүлснээрээ монгол хэрэглэгчдийн дунд нэлээн давуу талтай юм. Энэ виртуал гарын үндсэн боломжууд нь урьдчилан үг санал болгох систем хийснээр тест бичих хурдыг хамгийн багадаа 30% нэмэгдүүлэх, зарим түгээмэл бичигддэг тэмдэгтүүдийг товч солихгүйгээр оруулах, цөөн тэмдэгтийн нийлэмжээр бүтэн өгүүлбэр бичих, apple-ийн үндсэн загварыг хэвээр хадгалснаар хэрэглэгчийн дадал зуршилд дадал зуршилд өөрчлөлт оруулахгүй, тексттэй харьцдаг бүр апплейшнуудад тохирох, монгол хэл дээр алдаатай бичих магадлалыг бууруулах гэх мэт функцуудтай. Удахгүй хийх шинэчлэлтүүдийн хувьд үг бүрийг дууссаны дараа дараагийн бичигдэх магадлалтай үгсийг түрүүлж санал болгох, Андроид ухаалаг утасны хувилбар гаргана гэсэн зорилтуудыг тавьсан байна.

2.1.3 UBInfo виртуал гар

UBInfo гар утасны апплейкшн таны хэрэглээг улам хялбар, уян хатан болгохын тулд шинэ зүйлсийг хийсээр, шинэчилсээр байна. Тухайн компани нь тэмдэг болон үсгүүдийг нэг дор багтаасан хэрэглэхэд хялбар гар утасны монгол виртуал гарыг хийсэн байна. Google Gboard гарын харагдах загварыг ашигласнаар тухайн гарыг хэрэглэдэг хэрэглэгчдэд хялбар болгосон. Үндсэн боломжуудын хувьд үсэг тоог хамтаар нь багтаасан, бичих явцдаа үсэг, тоо болон тусгай тэмдэгтүүдийн хооронд гарын байрлалаа солих шаардлагагүй хурдан бичих боломжтой. Мөн виртуал гарын зүүн дээд хэсэгт байрлах UBInfo.mn-ы лого байрлах ба уг логон дээр дарахад UBInfo апплейкшн рүү шилжиж байгаа нь бүтээгдэхүүнүүдээ хооронд нь зохицон ажиллах чадварыг нэмэгдүүлсэн байна

2.2 Үйл ажиллагааны харьцуулалт

Gboard, Айкон, ЮБИнфо, Чимэгэ виртуал гаруудын ажиллаж байгаа зарчим ба нэмэлт функцуудыг харьцуулсан ба тухайн харьцуулалтыг нийтэд ил болсон мэдээлэл, туршилтдээр үндэслэн гаргалаа.

Table 2.1: Харьцуулалтын хүснэгт. Тайлбар: ' + ' - тэмдэгт нь тухайн үйлчилгээ байгааг төлөөлнө.

	Gboard	Айкон	Чимэгэ	ЮБИнфо
Зөв бүтэн гарны байрлал	+	+	+	+
Emoji	+		+	
GIF	+			
Үг санал болгох систем	+	+	+	
Олон хэл дэмжсэн байдал	+			
Бичвэрээс ярианд хөрвүүлэх систем(STT)	+		+	
Үгсийн алдаа шалгах систем	+		+	
Андройд үйлдлийн систем дээр дэмжсэн байдал	+		+	
IOS үйлдлийн систем дээр дэмжсэн байдал	+	+		

Харьцуулалтаас дүгнэж хэлэхэд виртуал гар нь хэрэглэгчдэд хэрэглэхэд хялбар ба маш олон функцүүдтэй болж байгаа нь сайшаалтай. Харьцуулсан бүх виртуал гарууд зөв бүтэн

гарын байрлалтай байна. Мөн Чимэгэ болон Gboard виртуал гарын хувьд олон хэл дэмжсэн бол Айкон.мн-ы виртуал гар урьдчилан үг санал болгох систем үнэхээр сайн ажиллаж байна. ЮБИнфо болон Gboard-ын гарууд утасны андройд болон ios үйлдлийн системүүд дээр дэмжсэн байна. Тухайн гаруудын хувьд хялбар болгож байгаа хамгийн зохимжтой шийдэл нь бичвэрээс ярианд хөрвүүлэх(SST) функц юм. Энэ нь тухайн нэг хэлний хувьд маш том ажил бөгөөд тусдаа хэл боловсруулалт гэх маш том судалгаа, туршилтын ажил явагдаж байгаа юм. Тухайлбал монгол хэл дээрх хэл боловсруулалтыг (Natural Language Processing) Монгол улсын их сургууль(МУИС), Өвөр монголын өөртөө засах орны их сургууль(ӨМӨЗОИС), Чимэгэ системс ХХК(ЧИМЭГЭ), Айкон.мн(АЙКОН) хийж байна. Ийм нэмэлт функцийг шигтгэж өгсөн Gboard болон Чимэгэ виртуал гарууд бусад гаруудаас ялгарч байна

3. СИСТЕМИЙН ШААДЛАГА

3.1 Хэрэглэгчийн шаардлага

3.1.1 Хэрэглэгчийн функциональ шаардлага

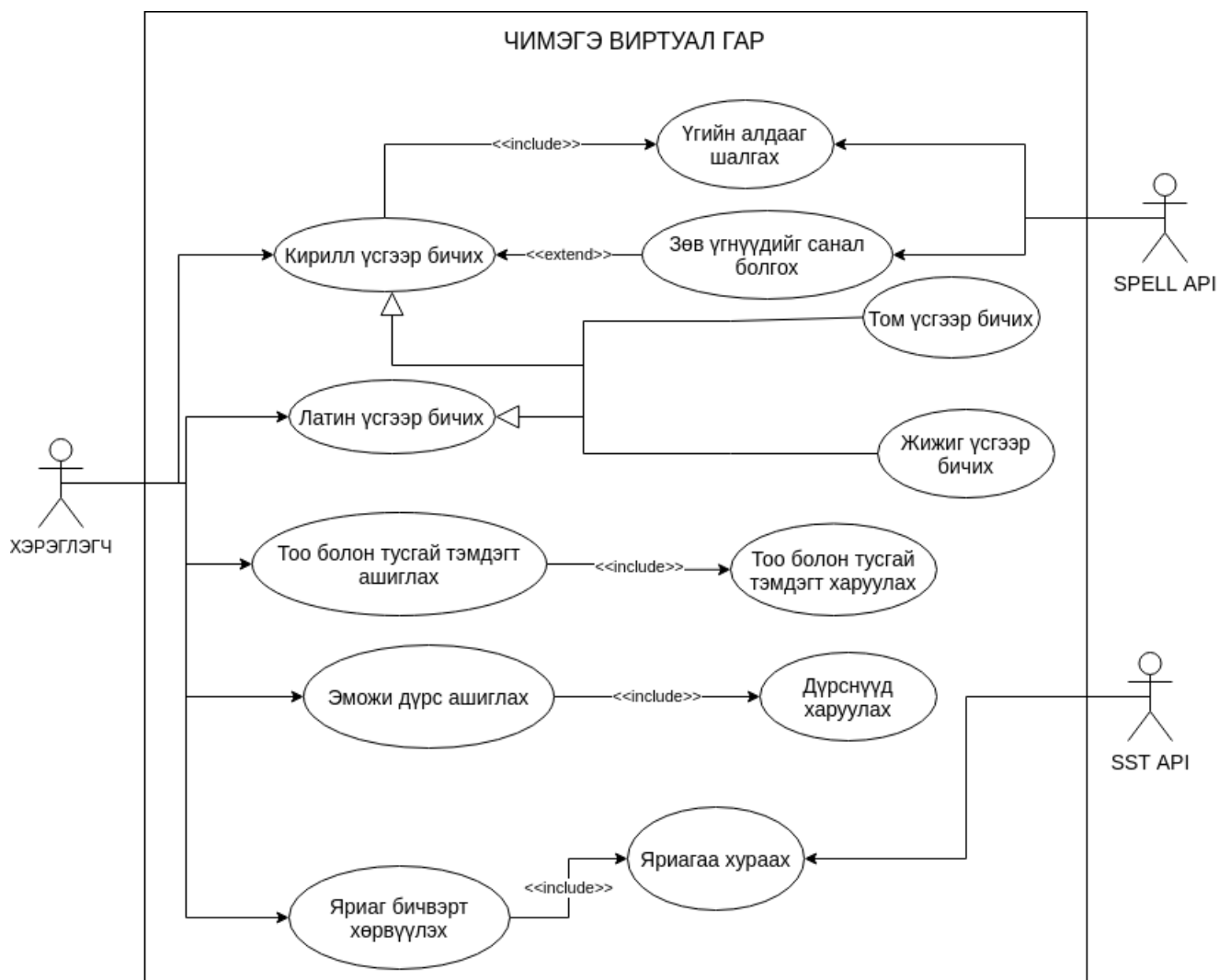
1. Хэрэглэгч монгол хэлний том болон жижиг бүх үсгүүдийг ашиглах боломжтой байх ёстой.
2. Хэрэглэгч яриаг бичвэр болгон хөрвүүлэх технологийг ашиглах боломжтой байх ёстой.
3. Хэрэглэгч англи хэлний том болон жижиг бүх үсгүүдийг ашиглах боломжтой байх ёстой.
4. Хэрэглэгч тусгай тэмдэгт ашилах боломжтой байх ёстой.
5. Хэрэглэгч цифр ашиглах боломжтой байх ёстой.
6. Хэрэглэгч үг бичихэд зөв бичиглэлтэй бүх үгийг санал болгодог байх ёстой.
7. Хэрэглэгч үг бичихэд тухайн үгийн алдааг шалгадаг байх ёстой.
8. Виртуал гарын ашиглах заавар гарын суулгацад байх ёстой.
9. Виртуал гар нь андройд үйлдлийн систем дээр ажилдаг байх ёстой.
10. Виртуал гарын суулгац монгол болон англи хэлэн дээр байх ёстой.

3.1.2 Хэрэглэгчийн функциональ бус шаардлага

1. Хэрэглэгч виртуал гарын харагдац болох 4-н өнгөнөөс сонгох боломжтой байх.
2. Хэрэглэгч виртуал гарын товчин дээр дарахад жижиргээ өгдөг байх.

3. Хэрэглэгч виртуал гарын товчин дээр дарахад чимээ өгдөг байх.
4. Хэрэглэгч виртуал гарын товчин дээр дарахад үсэг поп хэлбэрээр гардаг байх.
5. Хэрэглэгч эможи ашиглах боломжтой байх.
6. Хэрэглэгч тоо болон үсгийг нэг дэвсгэр дээр ашиглах боломжтой байх.
7. Виртуал гар нь олон хэл дэмжих боломжтой байх.
8. Виртуал гарын загвараас хамааран нэг товчин дээр 2 үсэг байх боломжтой байх.
9. Виртуал гарын доголдол нь 365 хоногт 24-н цаг байж болно.
10. Яриаг бичвэрт хөрвүүлэх API токеноо байнга шинэчилдэг байх хэрэгтэй.
11. Яриаг бичвэрт хөрвүүлэх функц хэрэглэгчид хариу өгөх хугацаа 3-н секундаас хэтрэхгүй байх.
12. Виртуал гарыг бүрэн ачаалах хугацаа 0.5 секунд байх.
13. Виртуал гар нь гар утасны хэмжээ төрлөөс хамаарч харагдац адил байх хэрэгтэй.
14. Виртуал гарын шинэчлэлтийг 6 - н сар тутам хийдэг байх.
15. Виртуал гар хэрэглэгчээс зөвхөн шаардлагтай нөөцийг ашиглах зөвшөөрлийг авдаг байх.

3.2 Use case diagram (Ажлын явцын диаграм)

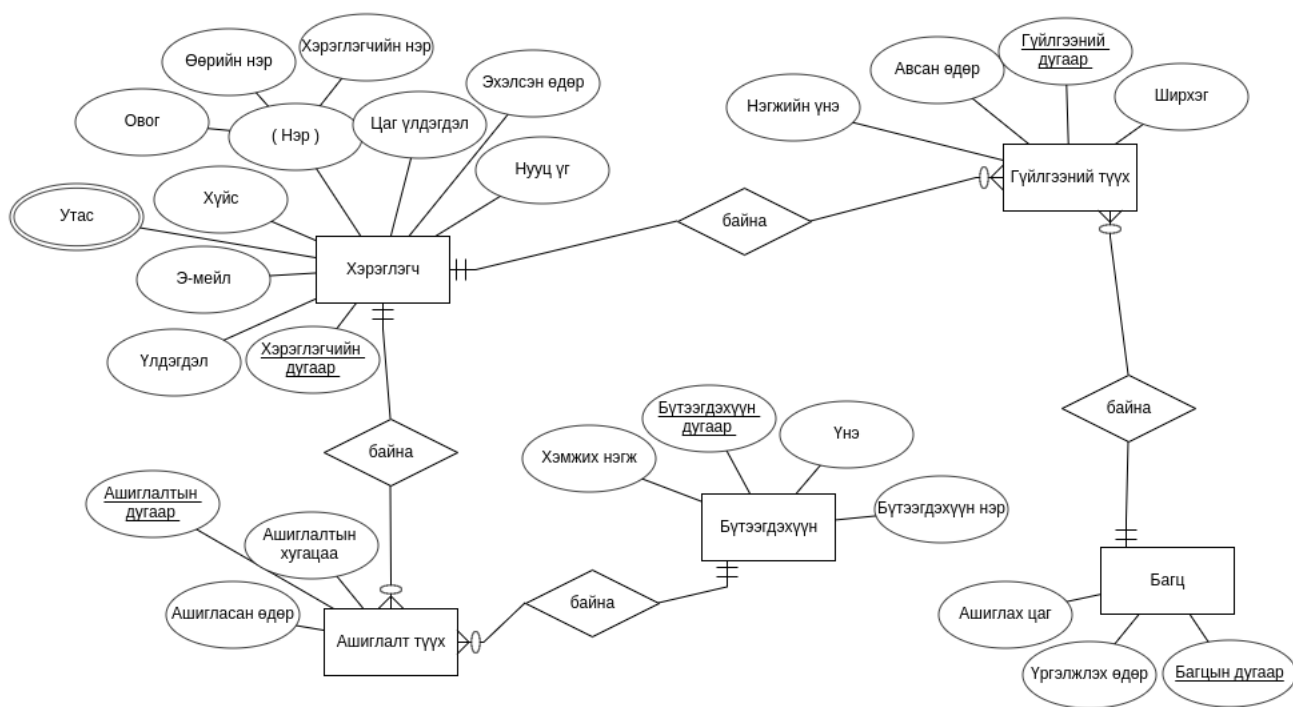


Зураг 3.1: Чимэгэ виртуал гарны ажлын явцын диаграмыг үүсгэв.

4. СИСТЕМИЙН ШИНЖИЛГЭЭ ЗОХИОМЖ

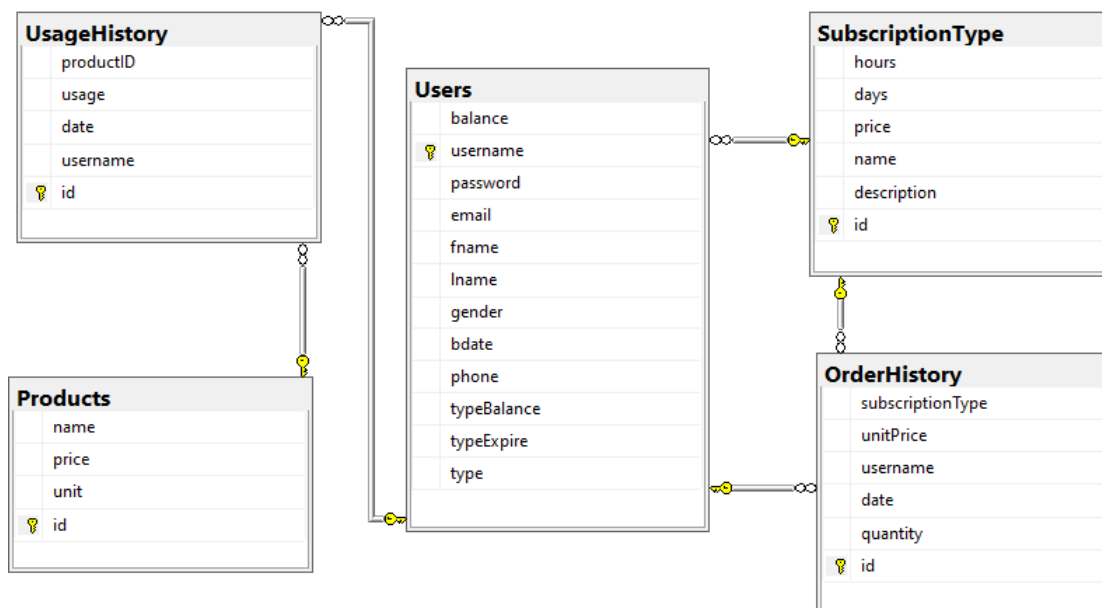
Миний хийж буй чимэгэ виртуал гарын хувьд өгөгдлийн сантай холбогдох үйл ажиллагаа байхгүй байсан ба яриаг бичвэр болгох функцийг ашиглахын тулд дадлага хийсэн компаний маань хэрэглэгч байх гэсэн шаардлага тавьж хэрэглэгч чимэгэ системээс гаргаж буй бүтээгдэхүүнүүдтэй ажиллах, ашиглах эрх авах үйл явцын диаграммыг зурлаа.

4.1 Өгөгдлийн сангийн диаграмм



Зураг 4.1: Чимэгэ системийн бүтээгдэхүүн хэрэглэгчийн өгөгдлийн сан.

4.2 Өгөгдлийн сангийн холбоост диаграмм



Зураг 4.2: ER диаграмаас гарч ирсэн холбоост диаграмм.

4.3 Өгөгдлийн толь бичиг

4.3.1 Users хүснэгт

User хүснэгт нь хэрэглэгчийн мэдээлэл болон хэрэглэгчийн төрлийг тодорхойлох мэдээллийг агуулна.

	Талбарын нэр	Өгөгдлийн төрөл	Тайлбар
1	userName	varchar(50)	Энэ багана нь Primary key, Хэрэглэгч системд ашиглах нэр
2	balance	BigInteger	Системийн үйлчилгээг ашиглах мөнгөн үлдэгдэл
3	lName	varchar(50)	Хэрэглэгчийн эцэг/эхийн нэр
4	fName	varchar(50)	Хэрэглэгийн өөрийн нэр
5	password	varchar(50)	Хэрэглэгчийн нууц үг
6	email	varchar(50)	Хэрэглэгчийн цахим хаяг
7	gender	char(1)	Хэрэглэгчийн хүйс
8	bdate	date	Хэрэглэгчийн төрсөн он, сар, өдөр
9	phone	char(8)	Хэрэглэгчийн утасны дугаар
10	typeBalance	float	Багцын үлдэгдэл, цагаар тооцно
11	typeExpire	dateTime	Багц дуусах хугацаа
12	type	int	Хэрэглэгчийн төрлийг тодорхойлно

Table 4.1: Users хүснэгт.

4.3.2 Products хүснэгт

Products хүснэгт нь яриа бичвэр, бичвэрийг яриа болгох гэх мэт үйлчилгээнүүдийн үнэ болон бусад мэдээллийг агуулна.

	Талбарын нэр	Өгөгдлийн төрөл	Тайлбар
1	id	varchar(10)	Бүтээгдэхүүний дугаар, РК
2	name	varchar(50)	Бүтээгдэхүүний нэр
3	unit	char(10)	Бүтээгдэхүүний хугацааг хэмжих тогтмол нэгж
4	price	bigInteger	Бүтээгдэхүүний нэгжийн үнэ

Table 4.2: Products хүснэгт.

4.3.3 UsageHistory Хүснэгт

UsageHistory хүснэгт нь хэрэглэгч ямар нэг бүтээгдэхүүнийг ашигласан хугацаа, өдөр гэх мэт мэдээллийг агуулна

	Талбарын нэр	Өгөгдлийн төрөл	Тайлбар
1	id	varchar(10)	Ашиглалтын дугаар, РК
2	usage	dateTime	Хэрэглэгч бүтээгдэхүүнийг ашигласан хугацаа
3	date	date	Ашигласан он, сар, өдөр
4	userName	varchar(50)	Хэрэглэгчийн систем дэхь нэр
5	productID	varchar(10)	Бүтээгдэхүүний дугаар

Table 4.3: UsageHistory хүснэгт.

4.3.4 *SubscriptionType* хүснэгт

Энэ хүснэгтэд бүтээгдэхүүнүүдийг ашиглах эрх авах буюу өөр өөр багцуудын мэдээллийг агуулна.

	Талбарын нэр	Өгөгдлийн төрөл	Тайлбар
1	id	varchar(10)	Багцын дугаар, PK
2	hour	int	Багцад агуулагдах цаг
3	date	int	Багц дуусах хоногийн тоо
4	name	varchar(50)	Багцын нэр
5	price	bigInteger	Багцын үнэ
6	description	text	Багцыг тайлбарласан талбар

Table 4.4: SubscriptionType хүснэгт.

4.3.5 *OrderHistory* хүснэгт

Энэ хүснэгт нь хэрэглэгчийн авсан багцын түүхийг хадгалах үүрэгтэй.

	Талбарын нэр	Өгөгдлийн төрөл	Тайлбар
1	id	varchar(10)	Гүйлгээний түүхийн дугаар, РК
2	subscriptionType	varchar(10)	Багцын дугаар
3	userName	varchar(50)	Систем дэхь хэрэглэгчийн нэр
4	unitPrice	bigInteger	Нэгж багц эсвэл цагийн үнэ
5	quantity	int	Хэдэн ширхэгийг багц эсвэл цаг авсаныг харуулсан тоо
6	date	date	Гүйлгээ хийсэн өдөр

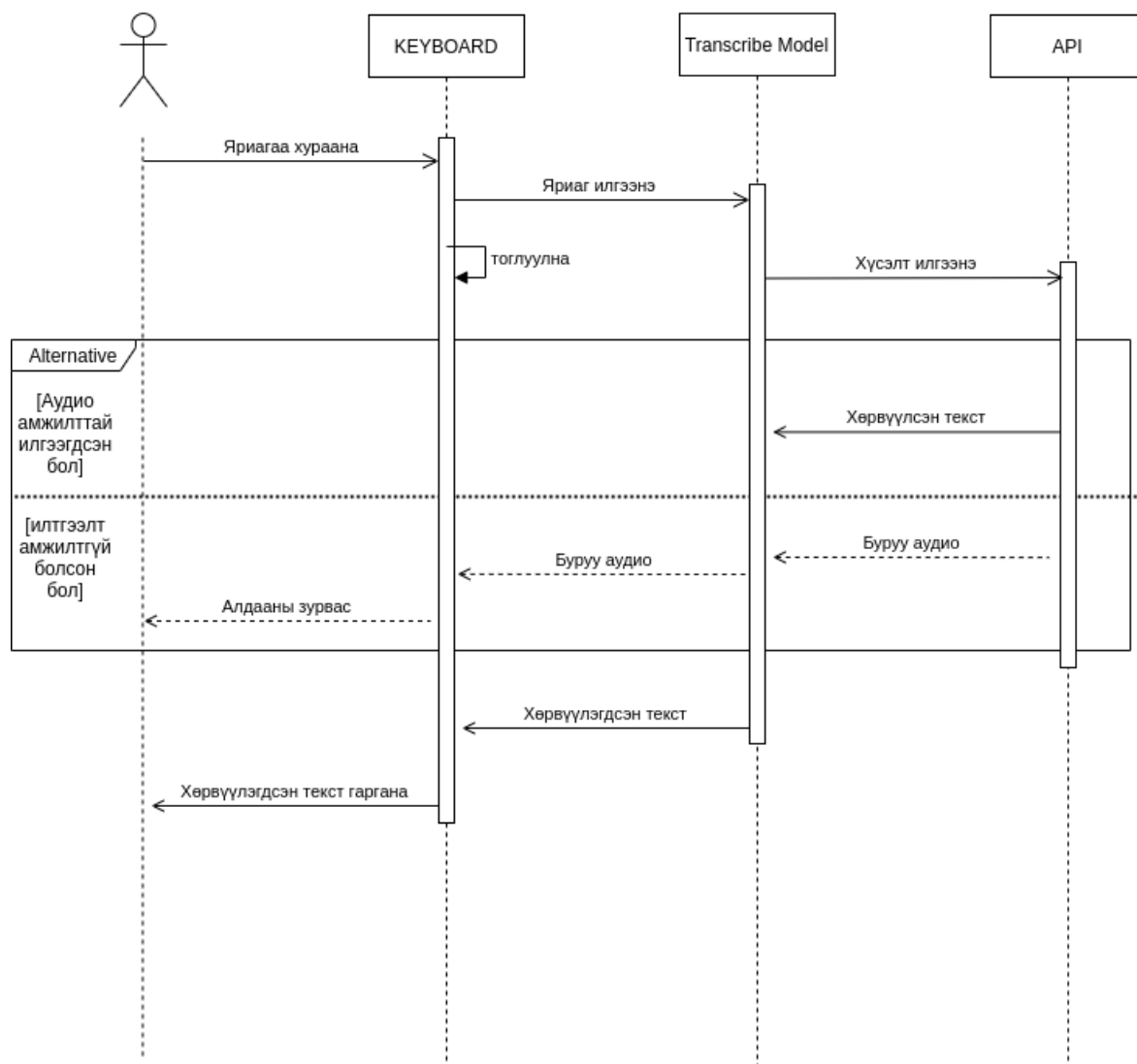
Table 4.5: OrderHistory хүснэгт.

Тухайн компанийн бүтээгдэхүүнд яриаг бичвэр болгох, орчуулах гэх мэт үйлчилгээнүүд багтана. Хэрэглэгч эдгээр үйлчилгээнүүдийг ашиглахын тулд заавал бүртгэлтэй байх ба бүртгэл дээрээ тулгуурлан бүтээгдэхүүнүүдийг ашиглах эрх болон ашиглах эрхийн гүйлгээг хийнэ. Үүний дагуу ER диаграммаас хамааран үндсэн 3-н хүснэгт үүссэн ба тухайн хүснэгтүүдийн харьцаануудаас хамааран OrderHistory, UsageHistory гэсэн хүснэгтүүд үүссэн. Users хүснэгт нь хэрэглэгчийн тухайн мэдээллийг агуулах, Products хүснэгт үйлчилгээнүүдийн үйл ажиллагаа болон үнийн мэдээллийг агуулах, UsageHistory хүснэгт нь хэрэглэгч аль нэг бүтээгдэхүүнийг хэдний өдөр, хэр хугацаагаар ашигласан мэдээллийг агуулах, SubscriptionType хүснэгт нь бүтээгдэхүүн бүрийн нэр, үнэ гэх мэт мэдээллийг агуулах бол OrderHistory хүснэгт нь гүйлгээний түүхийн мэдээллийг агуулж байна.

4.4 Дарааллын диаграмм (Sequence diagram)

Яриаг бичвэр болгон хувиргах үйлчилгээний диаграммыг зурав.

Яриаг бичвэрт хөрвүүлэх



Зураг 4.3: Яриаг бичвэр болгох үйлчилгээний дарааллын диаграмм.

5. ТЕХНОЛОГИЙН СУДАЛГАА

5.1 Үндсэн үгнүүдийн орчуулга

1. Input Method - Оролтын арга
2. Input Method Framework - Оролтын аргын фреймворк
3. Input Method Editor - Оролтын аргын эдитор
4. Soft keyboard - Уян хатан гар
5. Hand writing recognizers - Гар бичмэл танигч
6. Hard keyboard translator - Хатуу гарын орчуулагч

5.2 Дэлгэцэн дээрх оролтын арга

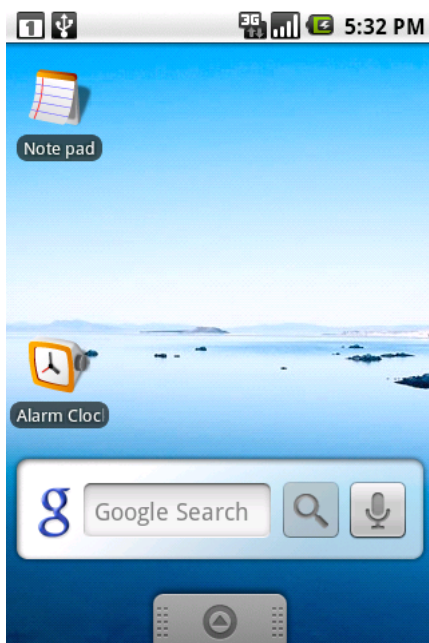
Андройд 1.5 хувилбараас эхлэн Андройд платформ нь оролтын аргын фреймворкийг санал болгодог болсон. Энэ фреймворк нь программ хангамжийн гар гэх мэт оролтын аргыг дэлгэцэн дээр үүсгэдэг. Оролтын аргын фреймворк нь техник хангамжийн гаргүй Андройд төхөөрөмжүүдийг дэмжих зорилготой.

Тухайн программ хангамжийн гарын судалгаагаар Оролтын аргын эдитор гэж юу болох түүнтэй сайн ажиллахын тулд программ юу хийх шаардлагатайг харуулна.

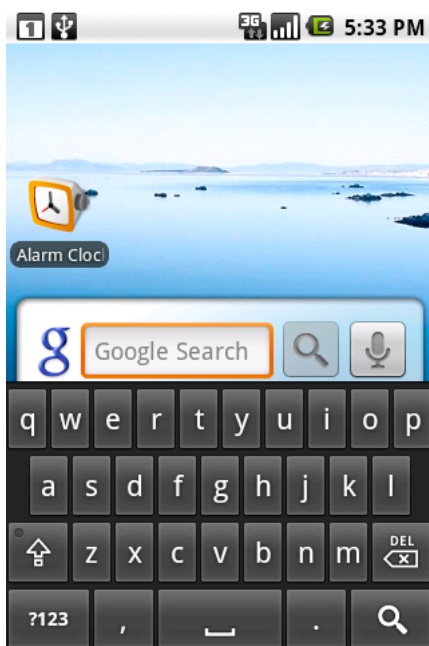
5.3 Оролтын арга гэж юу вэ?

Андройд оролтын аргын фреймворк нь олон төрлийн оролтын аргын эдиторууд болох уян хатан гар, гар бичмэл танигч, хатуу гарын орчуулагчийг дэмжихэд зориулагдсан юм. Энэ судалгааны ажлаар зөвхөн уян хатан гарыг судлах болно.

Доорх зурган дээр харуулсны дагуу үндсэн дэлгэцэн дээрх засварлах талбар дээр товшиж одоогийн оролтын аргын гар луу хандах боломжтой:



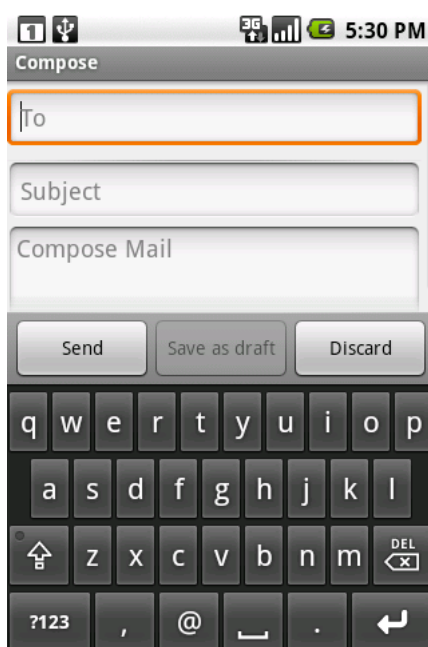
Зураг 5.1: Ухаалаг утасны үндсэн зураг.



Зураг 5.2: Виртуал гар дэлгэцэн дээр идэвхжсэн байдал.

Уян хатан гар нь дэлгэцийн доод хэсэгт программын цонхон дээр байрлана. программ болон оролтын аргын хоорондох зайг зохион байгуулахын тулд бид зарим аргыг ашигладаг. Дээрх зурган дээр харуулсан аргыг пан ба скан гэж нэрлэдэг бөгөөд зөвхөн одоо байгаа фокусыг харагдуулахын тулд программын цонхыг гүйлгэж харах боломжтой болно. Энэ арга нь одоо байгаа программуудын хувьд хамгийн аюулгүй тул өгөгдмөл горим юм.

Ихэнхдээ дэлгэцийн байрлалыг илүүд үздэг хэмжээ нь хэмжигдэхүүн бөгөөд энэ нь программын цонхыг бүхэлд нь харагдахаар хэмжээг нь өөрчилдөг. И-мэйл мессеж бичихдээ жишээг энд харуулав:



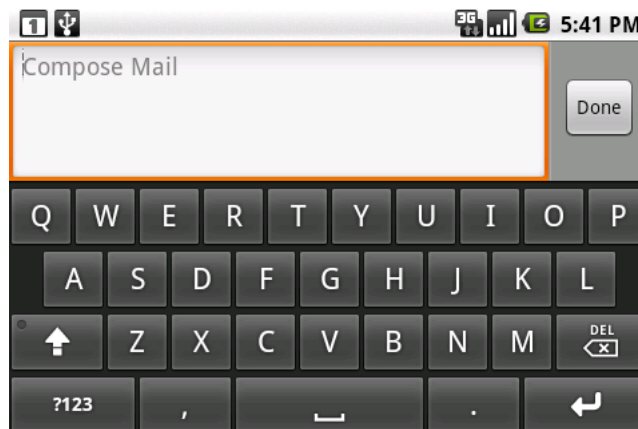
Зураг 5.3: Зөвхөн өөрчлөх талбар дээр төвлөрсөн байдал.

Программын цонхны хэмжээ өөрчлөгдөхөд оролтын арга засварлагчийг нуухгүй бөгөөд программ болон оролтын арга засварлагч хоёуланд нь хандах боломжийг олгоно. Энэ нь зөвхөн хангалттай зай гаргахын тулд багасгаж болох хэмжээг өөрчлөх боломжтой программуудад л ажиллана.

Сүүлийн горим бол бүтэн дэлгэц эсвэл хэсэгчлэн авах горим юм. Энэ горим нь андройд төхөөрөмж хэвтээ байрлалд байхад оролтын арга эдитор программын цонхны орон зайнаас

хэтэрхий их зай эзлэх тохиолдолд ашиглагддаг.

Энэ тохиолдолд программын цонх ижил хэвээр байх бөгөөд оролтын арга засварлагч нь доор харуулсны дагуу программын цонхыг тэр чигт нь эзэлснийг харуулав:



Зураг 5.4: Утас хөндлөн үед виртуал гар дэлгэцэнд идэвхжих.

Оролтын арга засварлагч нь програмын цонхыг хамардаг тул тухайн програмд агуулагдаж байгаа бичвэрийг үзүүлэх засварлах талбартай байдаг. Энэ тохиолдолд хязгаарлагдмал боломжуудтай байдаг учир нэмэлтээр “Done” товчлуур болон оролтын арга засварлагчыг далдлах түлхүүрийг хийж өгсөн.

5.4 Оролтын арга эдиторыг удирдах үндсэн XML шинж чанарууд

Систем нь одоо байгаа програмуудыг оролтын арга засварлагчтай аль болох ажиллахад туслахын тулд олон зүйлийг хийдэг. Жишээ нь:

- Хэмжээг өөрчлөх горим нь жагсаалтыг харах, гүйлгэх гэх мэт хэлбэрээр ажиллана гэж таамаглаж чадахгүй бол пан ба скан горимыг ашигладаг.
- Уян хатан гарын товчлуурыг зөв байрлуулахад туслах агуулгын төрлийг (тоо, энгийн текст гэх мэт) таахын тулд одоо байгаа TextView-ийн янз бүрийн шинж чанарыг шинжлэх.
- Бүтэн дэлгэцэн дээр байгаа оролтын арга засварлагчид хэдэн үндсэн үйлдлийг зааж

өгдөг. Жишээ нь “next field” ба “done” гэх мэт

Мөн өөрийнхөө программд зарим энгийн үйлдэл хийх боломжтой бөгөөд энэ нь тэдний хэрэглэгчийн туршлагыг ихээхэн сайжруулдаг. Тодорхой дурдаагүй бол тэдгээрийг Android платформын аль ч хувилбар дээр ашиглах боломжтой байх болно (Android 1.5-н өмнөх хувилбарууд ч гэсэн).

5.5 EditText удирдлага бүрийн оролтын төрлийг тодорхойлох

программын хамгийн чухал зүйл бол EditText тус бүр дээр шинэ android:inputType шинж чанарыг ашиглах явдал юм. Энэ шинж чанар нь текстийн агуулгын талаарх дэлгэрэнгүй мэдээллийг өгдөг. This attribute actually replaces many existing attributes (android:password, android:singleLine, android:numeric, android:phoneNumber, android:capitalize, android:autoText, and android:editable). Хэрэв та хуучин шинж чанар ба шинэ android: inputType шинж чанарыг зааж өгвөл систем нь android:inputType -ийг ашиглаж бусад шинж чанаруудыг үл тоомсорлодог. Android: inputType шинж чанар нь гурван хэсэгтэй:

- Энэ класс(class) нь тэмдэгтүүдийн ерөнхий тайлбар юм. Одоогийн байдлаар дэмжигдсэн классуудад text (энгийн текст), number (аравтын орон), phone (утасны дугаар), болон time (огноо эсвэл цаг) орно.
- Хувилбар(variation) нь ангийн талаархи илүү боловсронгуй шинж чанар юм. Анги шинж чанар дээр та ангилал, хувилбарыг хамтад нь зааж өгөн уг анги нь угтвар болно. Жишээлбэл, textEmailAddress нь текст оруулах талбар бөгөөд хэрэглэгчид энэ хаягийг имэйл хаягаар оруулна (foo@bar.com), тиймээс түлхүүр байрлал нь '@' тэмдэгтийг хялбархан нэвтрэх боломжтой газарт байрлуулдаг бол numberSigned бол тоон талбар юм. Хэрэв зөвхөн class-г зааж өгсөн бол default / generic variation-г авах болно.
- Цаашид боловсронгуй болгох үүднээс бусад flag зааж өгч болно. Эдгээр flag-ууд нь классын зориулалттай. Жишээлбэл, text классын зарим flag нь textCapSentences, tex-

tAutoCorrect ба textMultiline юм.

Жишээлбэл, энэ нь IM программын мессежийг харахад зориулагдсан шинэ EditText:

```
<EditText android:id="@+id/edtInput"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:inputType="textShortMessage|textAutoCorrect|textCapSentences|textMultiLine"
    android:imeOptions="actionSend|flagNoEnterAction"
    android:maxLength="2000"
    android:hint="@string/compose_hint"/>
```

Зураг 5.5: Уян хатан гар нь хэрэглэгчийн оруулсан текстэд тааруулан хамгийн сайн гарын байрлалыг ашиглахын тулд боломжтой оролтын зөв төрлийг ашиглах нь чухал юм.

5.6 Хэмжээг өөрчлөх горим болон бусад цонхны онцлог чанаруудыг идэвхжүүлэх

Программын хоёр дахь чухал зүйл бол оролтын аргатай харьцах цонхны ерөнхий байдлыг тодорхойлох юм. The most visible aspect of this is controlling resize vs. pan and scan mode, but there are other things you can do as well to improve your user experience.

AndroidManifest.xml дээрх бүр android:windowSoftInputMode -ийн шинж чанарыг тодорхойлдог. Оролтын төрөлтэй адилаар хоёр өөр өгөгдлийг нэгтгэн энд зааж өгөх боломжтой:

- Үйлдэл хийх үед хэрэглэгч оролтын аргын эдитор автоматаар гарч ирэх эсэхийг хянах, мөн хэрэглэгч өөр нөхцөл байдалд шилжих эсэхийг хянах боломжтой. Анхдагч байдлаар, систем нь оролт арын эдиторыг автоматаар харуулдаггүй гэхдээ программын цонх ачаалахад шууд оролтын аргын эдитор гарч ирэх тохиргоог апплейкшнаас идэвхжүүлж болдог. Үүний хэрэгжүүлэхийн гаргахын тулд stateVisible -г оноож өгнө.

Оролт гаралтын эдиторыг автоматаар гаргах тохиргоо:

```
<activity name="EditContactActivity"
    android:windowSoftInputMode="stateVisible|adjustResize">
    ...
</activity>
```

Зураг 5.6: Автомат тохиргоо.

5.7 Оролтын аргын эдитор -г хянах API

IME-г илүү боловсронгуй болгохын тулд ашиглаж болох олон янзын шинэ API байдаг. Үндсэн API нь шинэ `android.view.inputmethod.InputMethodManager` класс бөгөөд үүнийг `Context.getSystemService()`-г ашиглан сэргээх боломжтой. Энэ нь одоогийн IME-ийн оруулах хэсгийг далдлах эсвэл харуулах гэх мэт глобал оролтын аргын төлөвтэй харилцах боломжийг олгоно.

Оруулах аргуудын харилцан үйлчлэлийг хянадаг зарим шинэ цонхны flag-ууд байдаг бөгөөд тэдгээрийг одоо байгаа `Window.addFlags()` функц ба `Window.setSoftInputMode()` шинэ функцээр удирдах боломжтой.

Идэвхтэй оролтын аргын эдитор болон программын хоорондох харилцан үйлчлэлийн ихэнх нь `android.view.inputmethod.InputConnection` классаар биелэгддэг. Энэ бол программ дээр хэрэгжүүлсэн API бөгөөд оролтын аргын эдиторийг программ дээр зохих засварлах үйлдлүүдийг хийдэг.

5.8 Оролтын арга үүсгэх

Оролтын аргын эдитор (IME) нь хэрэглэгчдэд текст оруулах боломжийг олгодог хэрэглэгчийн удирдлага юм. Андроид нь оролтын аргын фреймворкийг ашигласнаар программын дэлгэцэн дээр гар эсвэл яриа оруулах гэх мэт нэмэлт аргуудыг нэмэх боломжийг олгодог. Шаардлагатай оролтын аргын эдиторыг(IME) суулгасны дараа хэрэглэгч IME-г системийн тохиргооноос сонгож, системд ашиглах боломжтой ба нэг удаад зөвхөн ганц IME-г идэвхжүүлэх боломжтой. IME-г Андроид системд нэмэхийн тулд `InputMethodService`-ээс удамшсан класс агуулсан

Андройд программыг үүсгэнэ. Мөн IME services-ийн тохиргоог удирдах “settings” activity үүсгэх боломжтой. IME-ийн амьдралын мөчлөгийг тодорхойлох диаграммыг доорх зургаар харуулав.



Зураг 5.7: IME-ийн амьдралын мөчлөг.

5.9 Manifest доторх IME-ийн бүрэлдэхүүнийг зарлах

Андройд системд IME нь тусгай IME service агуулсан андройд программ юм. Программын манифест файл нь үйлчилгээг зарлаж, шаардлагатай зөвшөөрлийг хүсэх, `action.view.InputMethod`-т зориулсан `intent filter`-ийг өгч, IME-ийн шинж чанарыг тодорхойлдог мета өгөгдлийг өгөх ёстой. Үүнээс гадна, энэ нь хэрэглэгчдэд IME-ийн ажиллах горимыг өөрчлөх боломжийг олгодог тохиргооны интерфэйсийг өгөхийн тулд системийн тохиргооноос эхлүүлж болох “settings” activity-г тодорхойлж болно.

Доорх зурган дээр IME service-г зарлаж байгаа бөгөөд энэ нь IME-ийг системд холбох, `android.view.InputMethod` зориулсан `intent filter` тохируулах, IME-ийн мета өгөгдлийг тодорхойлох зөвшөөрөл өгөхийн тулд `BIND_INPUT_METHOD` зөвшөөрлийг шаарддаг:

Доорх зурган дээр IME service-г зарлаж байгаа бөгөөд энэ нь IME-ийг системд холбох, `android.view.InputMethod` зориулсан `intent filter` тохируулах, IME-ийн мета өгөгдлийг тодорхойлох зөвшөөрөл өгөхийн тулд `BIND_INPUT_METHOD` зөвшөөрлийг шаарддаг:

```
<!-- Declares the input method service -->
<service android:name="FastInputIME"
    android:label="@string/fast_input_label"
    android:permission="android.permission.BIND_INPUT_METHOD">
    <intent-filter>
        <action android:name="android.view.InputMethod" />
    </intent-filter>
    <meta-data android:name="android.view.im"
        android:resource="@xml/method" />
</service>
```

Зураг 5.8: IME-ийн амьдралын мөчлөг.

6. ХЭРЭГЖҮҮЛЭЛТ

21-р зуунд хүмүүс дэлхийн хаана ч байсан хоорондоо харилцах болсон ба хоорондоо харилцахад хамгийн их хэрэглэдэг хэрэгслийн нэг бол ухаалаг утас юм. Түүнийгээ дагаад утасны жижиг хэмжээнд уян хатан ажиллах оролтын аргын эдитор буюу виртуал гар нь хүний хэрэглээнд нийцэн утастай маш сайн зохицон ажилдаг болоод байна. Тухайн виртуал гарын хөгжүүлэлтийг тухайн ухаалаг утас хийж буй компаниуд болон хүний хэрэгцээнд тааруулан шинэчилж буй хөгжүүлэгчид юм. Үүний нэг төлөөл бол Норвегийн улсын их хурал, Засгийн газрын яам, Шинжлэх ухааны яам, санхүүжүүлэн Сами хэлний технологийн төвтэй хамтран Divvun төслөөр норвегийн бүс нутгийн хүрээнд хамрагдах орнуудыг виртуал гаруудыг багцлан хийсэн юм. Утасны виртуал гараас гадна компьютер дээрх виртуал гар, норвегийн бүс нутгийг хүрээнд яриаг бичвэр хөрвүүлэх технологи, орчуулга, толь бичиг, хэлний сургалт гэх мэт олон үйл ажиллагааг хийсэн байна. Тухайн Divvun төсөл нь GitHub буюу Git гэх программ хангамж боловсруулах явцад эх кодын өөрчлөлтийг хянах зорилгоор тархсан системийг ашигласан программ хангамжийн хөгжүүлэлтэд үйлчилгээ үзүүлдэг систем дээр виртуал гарын эх кодыг тавьж, хөгжүүлэгчдэд тухайн виртуал гар дээр өөрсдийн хийх гэж буй хэрэгжүүлэлт, хөгжүүлэлт хийх боломжийг олгосон Миний бие дадлагынхаа ажлын хүрээнд Divvun төслийн виртуал гарыг ашиглан монгол хэл виртуал гарыг хийж гүйцэтгэв. Мөн “Чимэгэ” системийн яриаг бичвэрт хөрвүүлэх технологи болон “Болор” үгийн алдаа шалгагчийг ашиглан тухайн виртуал гар дээрээ нэмэлт функц болгон хөгжүүлэлтийн ажлыг хийж гүйцэтгэснээр монгол хэл ба монгол хэлний яриаг бичвэрт хөрвүүлэх хөрвүүлэгчтэй, үгийн алдааг шалгаад зөв үгнүүдийг санал болгодог боломжтой ухаалаг утасны виртуал гарыг бий болгов.

6.1 Системийг бичихэд ашигласан технологи

Болорсофт ХХК нь судалгаанд суурилсан хөгжүүлэлт (R&D), найдвартай, чанартай, анхдагч байдлыг бий болгосон програм хангамжийн бүтээгдэхүүнүүд үйлдвэрлэхээр зорин анх 2008 онд байгуулагдсан. 2008-2013 оныг хүртэл мэдээлэл технологийн өрсөлдөөний давлагаанд

чанартай байдлыг эрхэмлэн зогсож чадсан ба бодит мэдээлэл дээр үндэслэн програм хангамжаа хөгжүүлдэг билээ. Тиймээс тухайн компанийн бүтээгдэхүүнүүд салбартаа үргэлж анхдагч болдог бөгөөд зохиогчийн эрхээр хамгаалагдсан байдаг. Болорсофтын хамт олон Монгол улс болон Герман улсад мэргэшсэн чадварлаг, ажлаа бүрэн мэддэг залуусаас бүрдэхээс гадна салбартаа тэргүүлэх програм хангамжийн архитектурч зөвлөхөөр ажиллаж байна. Энэ компани нь олон улсын түвшинд үйл ажиллагаа явуулдаг бөгөөд АНУ болон Герман улсад төлөөлөгчид нь ажилладаг тул дэлхий нийтийн мэдээлэл технологийн хөгжлөөс огтхон ч хоцордоггүй.

6.1.1 Kbdgen гэж юу вэ?

Kbdgen нь тухайн хэлний гарын байрлалыг тодорхойлсон YAML файл ашиглан виртуал гар бий болгодог хэрэгсэл(tool) юм. Энэ хэрэгсэл нь Python 3.7 ба түүнээс дээш хувилбаруудыг шаарддаг. YAML нь “YAML Ain’t Markup Language” үгийн товчлол бөгөөд нь хүний уншихад хялбар өгөгдлийн цуврал хэл юм. Энэ нь ихэвчлэн өгөгдөл хадгалах, дамжуулахад зориулагдсан тохиргооны файлууд болон программуудад ашиглагддаг.

6.1.2 Kbdgen хэрэгсэл ашиглахын тулд юу бэлдэх шаардлагатай вэ

Виртуал гарыг бий болгохын тулд kbdgen хэрэгсэл нь project.yaml-ийг уншин ажилдаг ба тухайн файл дотор сонгосон хэлийн тодорхойлолт байна. Project.yaml дотор файлын зохиогчийн нэр болон бусад тодорхойлолтууд, үүсгэх гэж буй гарын хэлний нэр болон ямар үйлдлийн систем дээр бий болгох талаарх мэдээлэл бүгд багтсан байдаг.

```

locales:
  en:
    name: &bundleName "Mongolian Keyboard (Bolorsoft)"
    description: "Keyboard for Mongolian script."
  mn:
    name: "Монгол бичгийн гар (Болорсофт)"
    description: "Болорсофтын Монгол бичгийн гар."

author: Badral Sanlig
email: badral@bolorsoft.com

copyright: (c) 2019 Bolorsoft LLC. MIT License.
organisation: Bolorsoft LLC.

layouts: [moncy, monbs]

version: "1.0"
build: "1"

targets:
  android:
    packageId: com.bolorsoft.keyboards.mon
    icon: image/icon-monbs.pdf
    #keyStore: /Users/bolorsoft/git/kbdgen/mon_keyboard.keystore
    keyAlias: monbs


```

Зураг 6.1: Монгол кирилл болон монгол бичгийн виртуал гарыг тодорхойлсон файлын зураг

Сонгосон хэл дээрх project.yaml файлаа үүсгэсний дараа тухайн хэлний ухаалаг утсан дээр үсгүүдийн харагдах байрлалыг нь тодорхойлж өгөх бөгөөд хэдэн хэл сонгосон тус бүр дээр .yaml өргөтгөлтэй файл үүснэ.

```

internalName: moncy

displayNames:
  en: Mongolian (Cyrillic, Bolorsoft)
  mn: Монгол (Кирилл, Болорсофт)
  cmg: 

locale: mn

modes:
  mobile-default: |
    ф ц у | ж э н г ш ү з к ъ
    й ы б ө а х р о л д п
    я ч ё с м и т ь в ю
  mobile-shift: |
    Ф Ц У Ж Э Н Г Ш Ү З К Е
    Й Ы Б Ө А Х Р О Л Д П
    Я Ч Ё С М И Т Ь В Ю
  iso-default: |
    ( № - " № : . _ , % ? е щ
    ф ц у ж э н г ш ү з к ъ
    й ы б ө а х р о л д п !
    = я ч ё с м и т ь в ю
  iso-shift: |
    ) 1 2 3 4 5 6 7 8 9 0 Е Ы

```

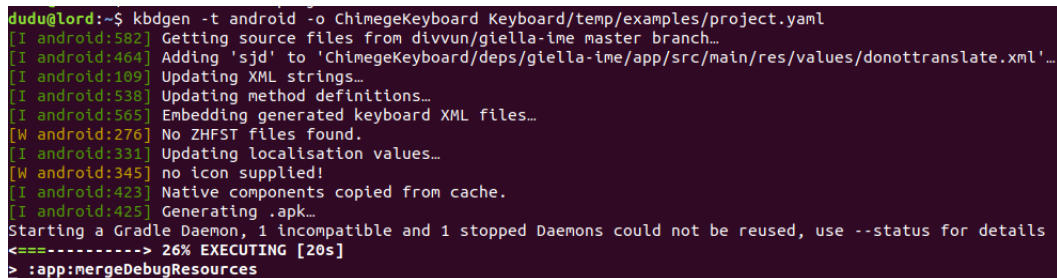
Зураг 6.2: Монгол кирилл бичгийн виртуал гар дээрх үсэг болон тэмдэгтүүдийг зааж өгсөн файлын зураг.

6.1.3 *Андройд үйлдлийн систем дээр виртуал гар бий болгоход дараах алхмуудыг гүйцэтгэнэ.*

- Python 3.7 ба түүнээс дээш хувилбарыг шаардах хувилбарыг суулгасан байх
- Сонгосон хэлний ерөнхий тодорхойлолт буюу project.yaml файлыг. үүсгэсэн байх
- Тухайн хэлний үсгүүдийг гарын байрлал дээр тодорхойлох lan_name.yaml файлыг үүсгэсэн байх
- Ингэснээр доорх командыг терминал дээрээ ажилсуулснаар андройд прожект үүсгэх боломжтой.

- kbdgen -t <target> -o <output-dir> <path/to/project.yaml>
- Kbdgen - Python дээр байдаг хэрэгсэл
- <target> - суулгах гэж буй зорилтот үйлдлийн системийн нэр. Android, iOS гэх мэт
- <output-dir> - Андроид прожект үүсэх байршлыг өгөх
- <path/to/project.yaml> - Сонгосон хэлийг тодорхойлж өгсөн

Монгол кирилл бичгийн виртуал гарыг үүсгэж буй командыг доорх зурган дээр харуулав.



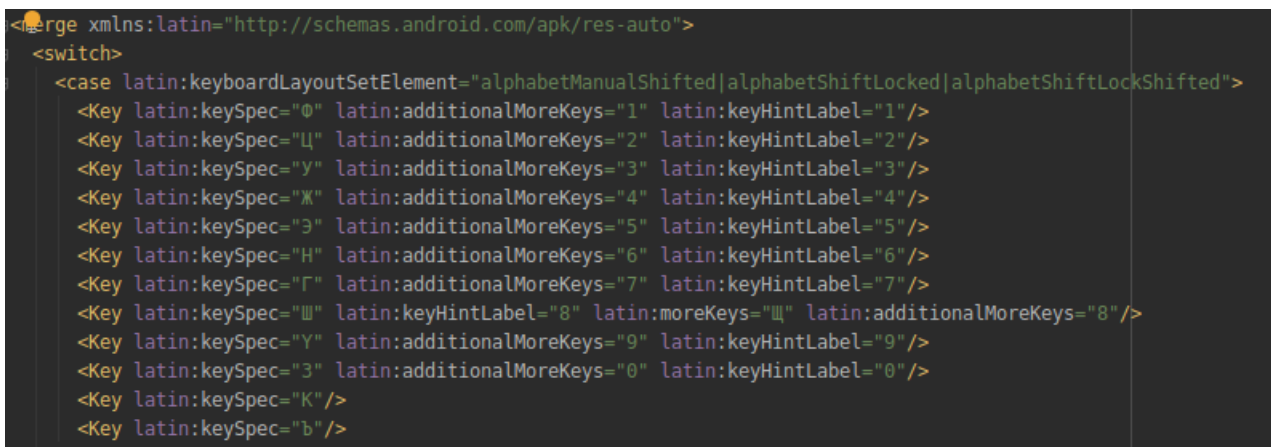
```
dudu@lord:~$ kbdgen -t android -o ChimegeKeyboard Keyboard/temp/examples/project.yaml
[I android:582] Getting source files from divvun/giella-ime master branch...
[I android:464] Adding 'sjd' to 'ChimegeKeyboard/deps/giella-ime/app/src/main/res/values/donottranslate.xml'...
[I android:109] Updating XML strings...
[I android:538] Updating method definitions...
[I android:565] Embedding generated keyboard XML files...
[W android:276] No ZHFSST files found.
[I android:331] Updating localisation values...
[W android:345] no icon supplied!
[I android:423] Native components copied from cache.
[I android:425] Generating .apk...
Starting a Gradle Daemon, 1 incompatible and 1 stopped Daemons could not be reused, use --status for details
<====-> 26% EXECUTING [20s]
> :app:mergeDebugResources
```

Зураг 6.3: Монгол кирилл болон монгол бичгийн виртуал гарыг тодорхойлсон файлын зураг

6.1.4 Андроид прожект дээрх кодын хэрэгжүүлэлт

Тухайн ерөнхий прожект нь Норвегийн технологийн сургуулийн хамт олон хийсэн, нэлээн том кодын бүтэцтэй, утасны үйлдлийн системтэй харилцан сайн ажилдаг виртуал гар юм. Ердийн кирилл үсэгтэй монгол виртуал гаран дээр яриаг бичвэрт хөрвүүлэх хөрвүүлэгч болон алдаатай үгс дээр зөв бичиглэлийг санал болгох нэмэлт функцуудыг нэмсэн болно. Миний бие тухайн виртуал гарын кодын хэсгийг онолын судалгаан дээрээ үндэслэн кодыг нягтлан харж, дибаг хийн ойлгосон ба хаана аль хэсэгт хэрэгжүүлэлт хийх ёстойг мэдэж авсан.

- Монгол виртуал гарын зураглал XML буюу товчхон хэлбэл өгөгдлийг хадгалах, дамжуулахад зориулагдсан хэлбэрээр кодлогдсон байдаг.



```
<?xml xmlns:android="http://schemas.android.com/apk/res-auto">
<switch>
  <case android:keyLayout="alphabetManualShifted|alphabetShiftLocked|alphabetShiftLockShifted">
    <Key android:keySpec="Ө" android:additionalMoreKeys="1" android:keyHintLabel="1"/>
    <Key android:keySpec="Ц" android:additionalMoreKeys="2" android:keyHintLabel="2"/>
    <Key android:keySpec="У" android:additionalMoreKeys="3" android:keyHintLabel="3"/>
    <Key android:keySpec="Ж" android:additionalMoreKeys="4" android:keyHintLabel="4"/>
    <Key android:keySpec="Э" android:additionalMoreKeys="5" android:keyHintLabel="5"/>
    <Key android:keySpec="Н" android:additionalMoreKeys="6" android:keyHintLabel="6"/>
    <Key android:keySpec="Г" android:additionalMoreKeys="7" android:keyHintLabel="7"/>
    <Key android:keySpec="Ш" android:additionalMoreKeys="8" android:keyHintLabel="8"/>
    <Key android:keySpec="Ү" android:additionalMoreKeys="9" android:keyHintLabel="9"/>
    <Key android:keySpec="З" android:additionalMoreKeys="0" android:keyHintLabel="0"/>
    <Key android:keySpec="К"/>
    <Key android:keySpec="Ь"/>
  </case>
</switch>
```

Зураг 6.4: Жишээ нь эхний мөрийн кирилл үсгийн байрлалыг харуулав. Өмнө хэлж байснаар тухайн хэлний YAML файлыг үүсгэж, KBDgen хэрэгсэл ашиглан виртуал гар бий болгоноор гарын байрлал XML файл хэлбэрээр ингэж кодлогдож байна.

- Яг дээд талын зургийн адилаар 4 дахь мөрөн дээр яриаг бичвэрт хөрвүүлэх дуу бичлэгийн товчийг нэмсэн.


```

31         latin:keyboardLayout="@xml/key_comma" />
32     <include
33         latin:keyXPos="25%p"
34         latin:keyboardLayout="@xml/key_space_5kw" />
35     <!--         lundeg voice key nemegdev-->
36     <Key
37         latin:keyStyle="voiceKeyStyle"
38         latin:keyWidth="10%p"
39     />
40 <Key
41     latin:keyStyle="enterKeyStyle"
42     latin:keyWidth="fillRight" />
43 </Row>

```

Зураг 6.5: Энэ мөрөнд хэл солих, зай авах, тэмдэгт рүү шилжих, догол мөр шилжих гэсэн үндсэн товчууд дээр нэмээд дуу бичлэгийн товчийг нэмэв. Нэмсэн хэсгийнхээ xml бичиглэлийг танилцуулбал `keyWidth = "10%"` энэ код нь утасны дэлгэцүүдээс хамааран тухайн дэлгэцийн урт хэсгийн 10%-ийг эзэлнэ. Мөн `keyStyle="voiceKeyStyle"` код нь дуу бичлэгийн зураг, өнгийг агуулдаг.

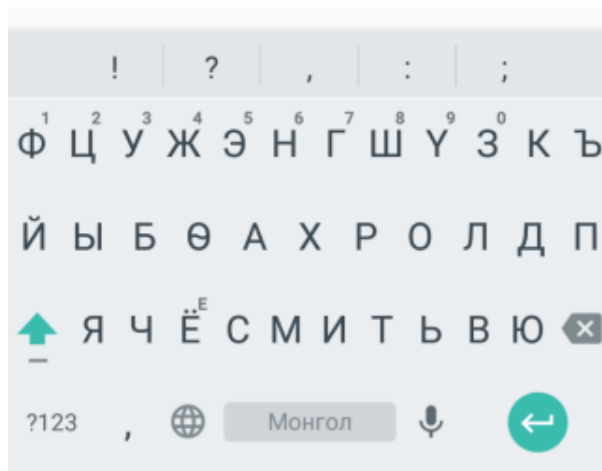
```

<!--         voice key lundeg -->
<key-style
    latin:styleName="voiceKeyStyle"
    latin:keySpec="!icon/shortcut_key|!code/key_shortcut"
    latin:keyActionFlags="noKeyPreview"
/>
<!--         -->

```

Зураг 6.6: Дуу бичлэгийн зураг, харагдах хэсэг нь бүх товчны style-уудыг багтаасан хэсэгт байрлах ба ашиглах газраа нэмж өгнө .

- Дуу бичлэгийн товчийг нэмж өгөн виртуал гарыг утсан дээр ажиллуулан доорх зурагт харуулав.



Зураг 6.7: Үсгээс эхлэн мөрийг тоолбол тухайн гар нь 4-н мөртэй ба 4-р мөрөнд дуу бичлэгийн товч харагдаж байна.

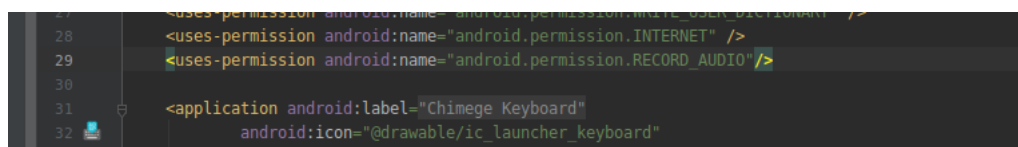
- Үүний дараагаар java виртуал гарын товчны үзэгдлүүдийг хийж байгаа java кодын хэсгийг олсон ба виртуал гарын key буюу товч болгон java код дээрээ дахин давтахгүй байдлаар кодолсон ба дуу бичлэгийн кодыг нь дибаг хийж мэдсэн ба товчны даралтыг сонсдог хэсэгт кодоор шалгаж дуу бичлэгийн хэсгийг хэрэгжүүлсэн. Доод талын зурагт жишээ байдлаар тухайн хэсгийн зургийг харуулав.

```
@Override
public void onKeyPressed(@NonNull final Key key, final boolean withPreview) {
    key.onPressed();
    invalidateKey(key);
    Integer keyCode = key.getCode();
    if(keyCode == -7){
        if(ContextCompat.checkSelfPermission(getContext(),Manifest.permission.RECORD_AUDIO) == PackageManager.PERMISSION_DENIED){
            requestPermission();
            showHandlerMessage( info: "Та яриа бичих зөвшөөрөлөө олгоно уу!");
        }
        else {
            if (isNetworkAvailable()) {
                if (!checkSetToken) {
                    onUpdateToken();
                    checkSetToken = true;
                }
            }
        }
    }
}
```

Зураг 6.8: Дуу бичих зөвшөөрөл байгаа эсэхийг шалгаж буй зураг.

Дуу бичлэгийн кодыг -7 болохыг дибаг хийж мэдэж авсан ба тухайн хэсэгт дуу авиаг бичвэрт хөрвүүлэх кодын зохиомж дараах алхмуудаас бүрдсэн болно. Үүнд:

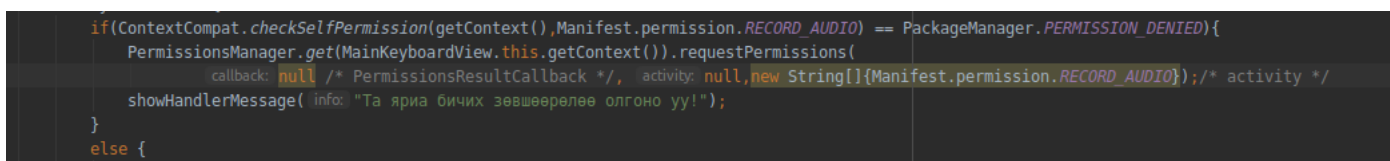
1. Ямар нэгэн товч дарагдахад дуу бичлэгийн код мөн эсэхийг шалгаж хэрэв дуу бичлэгийн товч дарагдсан тохиолдолд permission-ий зөвшөөрөл буюу хэрэглэгчээс аудиоогоо бичүүлэх эрхээ өгөхийг санал болгоно. Permission буюу зөвшөөрөлийн зорилго нь андройд ухаалаг утас хэрэглэгийн хувийн нууцыг хамгаалах явдал юм. Андройд аппликейшнууд нь хэрэглэгчийн нууцлалтай өгөгдөл (мессеж, бүртгэл, нууц үг гэх мэт) болон системийн камер, интернет, микрофон гэх мэт зүйлүүдээс нэвтрэх зөвшөөрөл автоматаар эсвэл хэрэглэгчийн хүсэлтээр асуух шаарлагатай байдаг.



```
27 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
28 <uses-permission android:name="android.permission.INTERNET" />
29 <uses-permission android:name="android.permission.RECORD_AUDIO" />
30
31 <application android:label="Chimege Keyboard"
32             android:icon="@drawable/ic_launcher_keyboard"
```

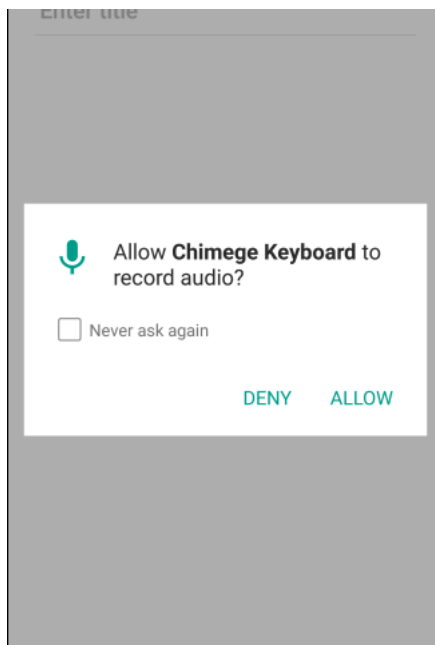
Зураг 6.9: Чимэгэ виртуал гарын Manifest дотор интернэт болон дуу бичлэг бичих зөвшөөрлийг бичсэн байна.

Харин Андройд 6.0 (API 23) ба түүнээс дээш хувилбартай апплейкшн суулгах үед хэрэглэгчдэд зөвшөөрлийг мэдэгдэхгүй ба апплейкшнг хэрэглэх явцад заавал хэрэглэгчээс асуух зөвшөөрлийг харуулах ёстой болсон. Жишээлбэл хүний яриаг бичихэд заавал хэрэглэгчээс асуух шаардлагатай ба доорх зурган дээр гарч байгаа диалог гарч ирэх ба тухайн үйл ажиллагааг зөвшөөрвөл дуу бичлэг бичээд бичвэрт хөрвүүлэх үйл явц нээгдэх ба татгалзах тохиолдол диалог алга болно.



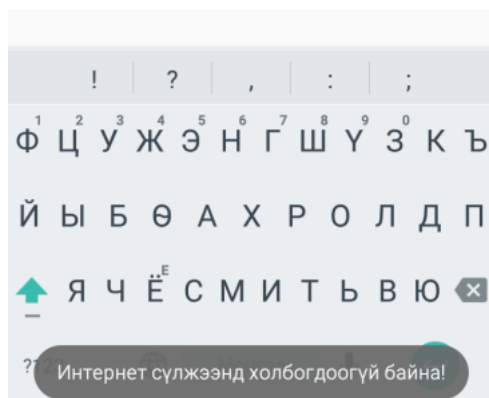
```
if(ContextCompat.checkSelfPermission(getContext(),Manifest.permission.RECORD_AUDIO) == PackageManager.PERMISSION_DENIED){
    PermissionsManager.get(MainKeyboardView.this.getContext()).requestPermissions(
        callback: null /* PermissionsResultCallback */, activity: null,new String[]{Manifest.permission.RECORD_AUDIO});/* activity */
    showHandlerMessage( info: "Та яриа бичих зөвшөөрөлөө олгоно уу!");
}
else {
```

Зураг 6.10: Дуу бичлэг бичих зөвшөөрөл хүсэх кодын хэсэг. Тухайн кодын хэсгийг бичиж өгөөгүй тохиолдолд яриаг бичвэрт хөрвүүлэх үйлчилгээг ашиглахын тулд хэрэглэгч өөрөө утасныхаа тохиргоо хэсэг рүү орж чимэгэ виртуал гарын дуу бичлэг бичих зөвшөөрлийг механик байдлаар хийж өгөх юм.






Зураг 6.11: 17-р зурган дээрх кодыг бичиж өгснөөр дуу бичлэгийн товчин дээр дарахад хэрэглэгчээс дуу бичих эрхийг зөвшөөрлийг хүссэн диалог гарч ирнэ.

2. Хэрэглэгч аудио бичих эрхээ өгсөн тохиолдолд дуу бичлэгийг бичвэрт хөрвүүлэх технологи хэрэгжих ба одоогоор тухайн үйлчилгээ нь интернэттэй үед ашиглагдаж байгаа учир заавал интернэт байгаа эсэхийг шалгах код бичиж өгсөн. Үүнд андройдод зориулсан жавагийн `connectivityManager` класс ашиглан гүйцэтгэсэн ба тухайн класс нь сүлжээний холболтын төлөвийн талаарх мэдээллийг өгдөг. Энэ нь мөн сүлжээний холболт салах үед апплейкшнуудад мэдэгддэг байна. WIFI, GPRS, UMTS гэх мэт сүлжээний холболтуудыг хянах чадвартай.



Зураг 6.12: Интернет байхгүй тохиолдолд мэдэгдэл ашигласан ба тухайн мэдэгдлийг хэрхэн гаргаж байгааг тусдаа дэлгэрэнгүй тайлбарлана.

3. Бид яриаг бичвэрт хөрвүүлэх технологийн API гаргаж өгсөн ба ямар нэгэн гэмт халдлагаас хамгаалж apiKey буюу токеноо цагийн хязгаартай шинэчилж байхаар болсон ба тухайн токеноо Firestore байршуулан хадгалж, тогтмол цагийн хязгаарт токен өөрчлөгддөг байдлаар хийж гүйцэтгэсэн. Cloud Firestore бол Firebase болон Google Cloud Platform-с гар утас, веб болон серверийн үйлчилгээнд зориулсан боловсруулалтын уян хатан, өргөжүүлэх боломжтой өгөгдлийн сан юм. Токений өөрчлөлтийг Firestore руу холбогдон зөвхөн өөрчлөгдөх тохиолдолд л дуудагдах функцийг хэрэгжүүлсэн. Ингэснээр хэрэглэгч яриагаас бичвэрт хөрвүүлэх хүсэлт илгээх болгонд токен зөв эсэхийг шалгах шаардлагагүй болох юм.

 bolor-dictionary-b8ced	 token	 token
+ Start collection	+ Add document	+ Start collection
<div>token ></div>	<div>token ></div>	+ Add field
		<pre>key: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoieMjAyMC0"</pre>

Зураг 6.13: Firestore дахь токены бүтэц .

4. Хүсэлт илгээх токеноо шалгасны дараагаар хэрэглэгчийн яриаг бичих кодын хэрэгжүүлэлтийг

хийсэн. Яриаг бичих кодын шаардлага бол яриагаа файл хэлбэрээр бичигдэхгүй зөвхөн утасны түр санах ойд буфферлэгдэж байт хэлбэрээр бичигдэх ба тухайн аудио нь wave файл буюу энгийнээр хэлбэл ямар ч засвар ороогүй, шахагдаагүй цэвэр файл байх гэсэн шаардлагуудыг тавьсан. Үүний дагуу андройдын AudioRecord классыг ашигласан ба энэ класс нь төхөөрөмжийн оролтын эд ангийг удирдан, яриаг бичих үйл явцыг хийдэг. AudioRecord классын read(byte[], offset, size) гишүүн функц нь байт цуваанд бичлэгийг бичдэг ба нийт бичлэгийг нэгтгэхдээ ByteArrayOutputStream классын write(byte[]) гишүүн функцийг ашигласан. Wave файлын эхний 44-н байт нь тухайн файлыг wave аудио мөн эсэхийг нь тогтоох өгөгдөл байдаг тул аудио хадгалж байгаа буфферын эхний 44-н байтад нь wave аудионы мэдээллийг нэмж өгнө.

```

/** Тухайн бичлэгнийхээ эхний 44-н байтад .wav файлын тодорхойлтыг өгөх функц */
private void writeWaveFileHeader(long totalAudioLen,
                                long totalDataLen, long longSampleRate, int channels, long byteRate)
{
    header = new byte[44];
    header[0] = 'R'; // RIFF/WAVE header
    header[1] = 'I';
    header[2] = 'F';
    header[3] = 'F';
    header[4] = (byte) (totalDataLen & 0xff);
    header[5] = (byte) ((totalDataLen >> 8) & 0xff);
    header[6] = (byte) ((totalDataLen >> 16) & 0xff);
    header[7] = (byte) ((totalDataLen >> 24) & 0xff);
    header[8] = 'W';
    header[9] = 'A';
    header[10] = 'V';
    header[11] = 'E';
    header[12] = 'f'; // 'fmt ' chunk

```

Зураг 6.14: Wave аудионы эхний 44-н байтын мэдээлэл.

5. Ярьсан бичлэгээ буфферт хадгалсны дараагаар тухайн яриаг бичвэр болгохын тулд энэ үйлчилгээг хийдэг Чимэгэ системийн API-г ашигласан ба уг үйлчилгээг авахын тулд хэрэглэгч болон серверийн хоорондох холболтыг тогтоож, идэвхжүүлдэг HTTP хүсэлтийг хэрэгжүүлсэн. Хэрэглэгчдийн бичсэн яриаг сервер лүү илгээх шаардлагатай тул HTTP хүсэлтийн POST аргыг ашигласан. POST арга нь нөөцийг үүсгэх, шинэчлэх зорилгоор сервер рүү өгөгдөл илгээхэд хэрэглэгддэг. Хүсэлтийг илгээхэд сервер хүлээн

авч бичвэр болгон боловсруулж буцаана. Яриаг бичвэр болгох үйлчилгээ нь үндсэн виртуал гарын үсэг дарагдах үйл явцыг гацаах тул энэ хэсгийг AsyncTask классыг удамшуулсан классыг үүсгэсэн. Энэ класс нь doInBackground(), onPostExecute() функцүүдийг дахин тодорхойлох шаардлагатай байдаг. Иймд doInBackground() функцэд яриаг бичвэр болгох хүсэлтийг хэрэгжүүлсэн ба onPostExecute() функцэд хүсэлтийн хариу буюу текстийг хүлээн авдаг байхаар хэрэгжүүлсэн. Мөн хоёр классын хооронд өгөгдөл дамжуулах үйл явцыг callback буюу товчны үзэгдэл сонсож байгаа гол класс дээрээ хариу авах функцийг хэрэгжүүлж өгснөөр яриаг бичвэрт хөрвүүлэх үйл явц виртуал гарын үйл ажиллагаанд саад учруулахгүй юм.

```
@Override
protected String doInBackground(TaskParam... params) {
    byte[] recordData = params[0].byteArray;
    String token = params[0].token;
    OkHttpClient client = new OkHttpClient();
    String speechToText;
    RequestBody body = RequestBody.create(recordData);
    Request request = new Request.Builder()
        .url(urlString)
        .header("apiKey", token)
        .addHeader("Content-Type", "WaveFile")
        .post(body)
        .build();
    try (Response response = client.newCall(request).execute()) {
        String jsonString = response.body().string();
        System.out.println(jsonString);
        JSONObject json = new JSONObject(jsonString);
        if (json.has("success")) {
            speechToText = json.getString("text");
            if (speechToText.equals("")) {
                error = "Яриаг таньсангүй, та дахин оролдоно уу!";
                System.out.println("speechToText: " + speechToText);
            } else if (speechToText != null) {
                return speechToText;
            }
        } else {

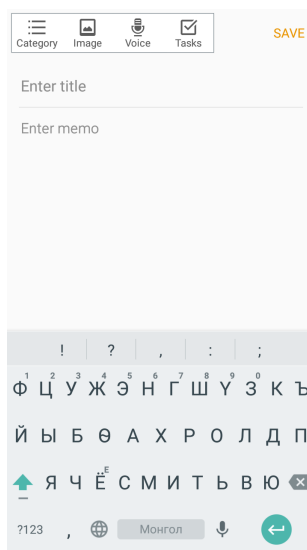
```

Зураг 6.15: Хөрвүүлэлт хийх хүсэлтийг хийж буй кодын хэсэг.

6.2 Ажлын явцын диаграммын тохиолдол бүрийн хэрэглэгчийн интерфейс

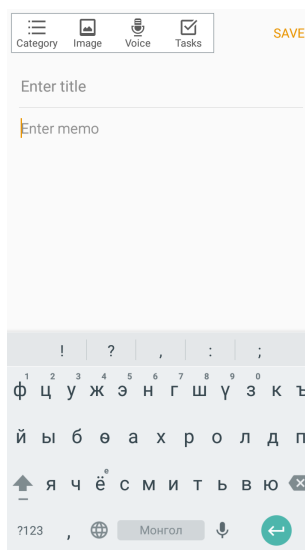
Хэрэглэгчийн интерфейс нь программ хангамж эсвэл нэг үгээр хэлбэл компьютерын оролцоо орсон төхөөрөмжүүдэд гадаад төрх, хэв маягт төвлөрч интерфейс хийх процесс юм. Виртуал гарын хувьд олон төрлийн өнгө дүрс, хэв маягтай утасны апплейкшнүүд дээр ажилдаг учраас ерөнхий хэлбэр хэмжээ, өнгө дүрсийг баримтлах нь чухал ч хэрэглэгчийн хэрэглээнд маш сайн тохирох шаардлагатай. Мөн виртуал гарын хувьд сонгосон хэлний үсэг том болон жижиг хэлбэрээрээ байх нь хамгийн чухал юм.

- Монгол кирилл том үсэгтэй виртуал гарын интерфейс



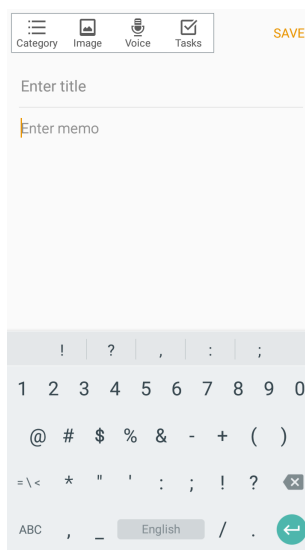
Зураг 6.16: Монгол том үсэгтэй зураг.

- Монгол кирилл жижиг үсэгтэй виртуал гарын интерфейс



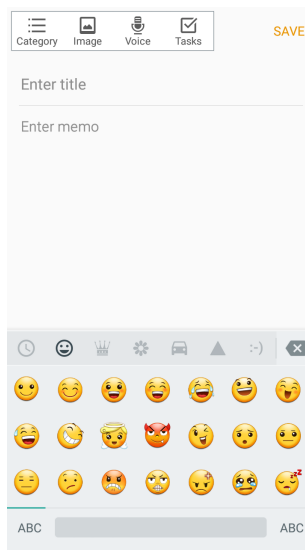
Зураг 6.17: Монгол жижиг үсэгтэй зураг.

- Тоо болон тэмдэгтэй виртуал гарын интерфейс



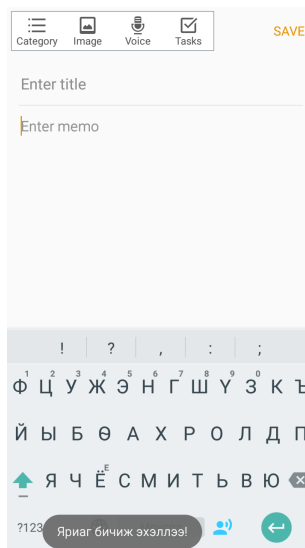
Зураг 6.18: Тоо болон тэмдэгтийн зураг.

- Төрөл бүрийн эможи зургаас бүрдсэн виртуал гарын интерфейс



Зураг 6.19: Эможи зураг.

- Яриаг бичвэрт хөрвүүлэх хэсгийн өөрчлөлтийг харуулсан зураглал

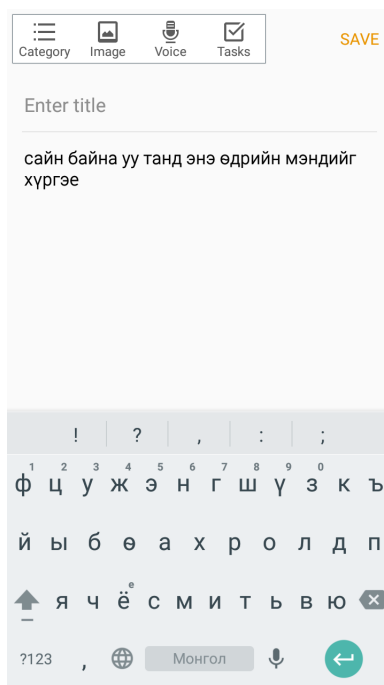


Зураг 6.20: Яриа бичиж буй үйлдлийн зураг.

7. ДҮГНЭЛТ

7.1 Үр дүн

Яриаг бичвэр болгох функцээ ашиглан ”Сайн байна уу? Таньд энэ өдрийн мэнд хүргэе” гэж хэлсэн ярианы үр дүнгийн зураглалыг харна уу.



Зураг 7.1: Яриа бичиж буй үйлдлийн зураг.

7.2 Үр дүнгийн тайлан

Миний бие дадлагынхаа хугацаанд мэдээлэл технологийн компаний ажлын явц, төлөвлөгөө, зорилго, хийж буй ажлуудтай танилцсан ба тухайн компани хийж буй технологийг ашиглан бүтээгдэхүүн хийх үйл явцад нь багш нарын зааж сургасан сургалтын мэдлэгээ ашиглан, цаашид улам дэлгэрүүлэн судлан өгсөн үүрэг даалгаврыг нь цаг тухайд биелүүлж ажиллаа. Ухаалаг утсыг хэрэглэн дэлхийн хаа нэгтээ болж буй бүхий л мэдээллийг түргэн шуурхай авч

байгаагаас хамаарч хэрэглэгч ихсэж байгаа энэ үед ухаалаг утсан дээрх программ хангамжууд хөгжиж байна. Үүний дагуу тухайн компаний гаргасан яриаг бичвэр болгох хөрвүүлэгч болон монгол үгийн алдаа шалгагчтай монгол виртуал гарыг оюутан миний бие хийж гүйцэтгэлээ. Ерөнхий виртуал гарын кодчиллыг Норвегийн технологийн их сургуулийн кодон дээр явсан. Эхний ажлын үйл явц монгол гарыг бий болгосон ба энэ үндсэн кодын ажиллагаа, хэрхэн зөв кодчилох бүтэц, зохиомжийг ойлгосон. Үүний үр дүнд онолын судалгаа хийж зөвхөн кодын аль хэсэгт хэрэгжүүлэлт хийх, үндсэн гарын үйл ажиллагаанд доголдол үүсгэхгүй байх алхмуудыг баримталсан. Ингэснээр монгол яриаг бичвэр болгон хөрвүүлэх товч болон үгийн алдаа шалгагчийг зөв хэсэгт нь нэмж хийсэн. Энэ нэмэлт үйл ажиллагаатай монгол гарын хувьд бусад монгол хэлтэй гаруудаас ялгарах давуу тал нь монгол хэлэн дээрх яриаг бичвэрт хөрвүүлэх функцтэй товч юм. Тухайн виртуал гар нь одоогоор зөвхөн яриаг тогтмол хэмжээнд бичээд хөрвүүлж байна. Үүнийг цаашид хэрэглэгч ярихад шууд урсгал хийж хариуг илгээх боломжтой ба ингэж сайжруулснаар хэрэглэгчийн ярианд хурдан хариу өгөх буюу шууд боловсруулах болно. Мөн монгол үндэстний бүх бичгийг энэ виртуал гартаа нэмэх боломжтой. Дүгнэж хэлэхэд багаар ажиллах, төлөвлөгөө гаргах, цаг баримтлах, жава хэлний мэдлэгээ сайжруулах гээд олон зүйл дээр туршлага суусан дадлага боллоо.

Bibliography

- [1] <https://android-developers.googleblog.com/2009/04/updating-applications-for-on-screen.html>
- [2] <https://divvun.github.io/kbdgen/>
- [3] <https://developer.android.com/guide/topics/text/creating-input-method.html#java>
- [4] <https://developer.android.com/reference/android/view/inputmethod/MethodInfo>
- [5] <https://developer.android.com/reference/android/os/AsyncTask>
- [6] <https://developer.android.com/reference/android/media/AudioRecord>
- [7] <https://mkyong.com/java/how-to-send-http-request-getpost-in-java/>
- [8] <https://developer.android.com/guide/topics/permissions/overview>

А. ШИНЖИЛГЭЭ ЗОХИОМЖ

А.1 Дадлагын ажлын удирдагчийн тодорхойлолт

МУИС, ХШУИС-ийн МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

Мэдээллийн технологийн АНГИЙН
ОЮУТАН Ц.Цагаанзориг -ЫН
ДАДЛАГЫН АЖЛЫН УДИРДАГЧИЙН ТОДОРХОЙЛОЛТ

2020 оны 03 сарын 10

Мэдээллийн технологийн 16815450389 кодтой оюутан Ц.Цагаанзориг нь манай байгууллагад 2020 оны .. сарын ..-ны өдрөөс .. сарын ..-ны өдөр хүртэл мэргэшүүлэх дадлагыг батлагдсан удирдамж, ажлын төлөвлөгөөний дагуу гүйцэтгэлээ. Оюутан Ц.Цагаанзориг -ын удирдамжийн дагуу дадлагын ажлыг гүйцэтгэсэн байдал:

- Мэргэшүүлэх дадлага гүйцэтгэсэн байгууллага: **Болорсофт ХХК**
- Дадлага ажлын сэдэв, даалгавар: *Яриал бүлвэрт хөрвүүлэх нэгдсэн функцтэй ухаалаг гар утасны монгол гар нийл*
- Ашигласан технологиуд: *Java, Android Studio, KBDGen*
- Даалгаврын гүйцэтгэлийн хувь: *100%*

Үнэлгээний санал:

Өгсөн даалгаврын хугацаандаа амжилттай биелүүлж, ирээ сайн байсан тул 10 оноо өгөх саналтай байна.

Дадлагын удирдагч *Ц.Бат-Оргил* /Ц.Бат-Оргил/

УЛААНБААТАР ХОТ
БОЛОРСОФТ ХХК
ХТА0454 Ө 5250579

Дадлагын удирдагчийн талаарх мэдээлэл:

1. Удирдагчийн овог, нэр, регистрийн дугаар *Ц.Бат-Оргил УП97102135*
2. Алба хэлтэс, албан тушаал *Программ хангамжийн инженер*
3. Утас, и-мэйл хаяг *bat-orgil@bolorsoft.com*
4. Төлбөрийн картын дугаар *5753699844* банк *Хаян Банк*

Зураг А.1: Дадлагын удирдагчийн тодорхойлолт.

В. КОДЫН ХЭРЭГЖҮҮЛЭЛТ

Яриаг бичвэр болгох ажилгааны голлох кодуудыг оруулав.

```
1  /**          */
2  Boolean touchDown = true;
3  String resultText = null;
4  Key currentKey;
5  byte[] recordArray;
6  @Override
7  public void onKeyPressed(@NonNull final Key key, final boolean
   withPreview) {
8      key.onPressed();
9      invalidateKey(key);
10     Integer keyCode = key.getCode();
11     if(keyCode == -7){
12         if(ContextCompat.checkSelfPermission(getContext(),Manifest.
           permission.RECORD_AUDIO) == PackageManager.
           PERMISSION_DENIED){
13             requestPermission();
14             showHandlerMessage("                !");
15         }
16         else {
17             if (isNetworkAvailable()) {
18                 if (!checkSetToken) {
19                     onUpdateToken();
20                     checkSetToken = true;
21                 }
22                 if (touchDown) {
23                     /**          microphone -
24                                     */
25                     key.setIconId(KeyboardIconsSet.setIconId("
                       shortcut_key_disabled"));
26                     bufferRecorder.startRecording();
27                     timer();
28                     showHandlerMessage("                !");
29                     touchDown = false;
30                 } else {
31                     /**          microphone -          */
32                     key.setIconId(KeyboardIconsSet.setIconId("
                       shortcut_key"));
33                     try {
34                         bufferRecorder.stopRecording();
35                     } catch (IOException e) {
36                         e.printStackTrace();
37                     }
38                     showHandlerMessage("                .                !")
39                     ;
40                     TaskParam params = new TaskParam(
                       bufferRecorder.totalByteArray.
                       toByteArray(),token);
41                     bufferRecorder.clear();
```

```

40         if (!token.equals("")) {
41             ApiTask apiTask = new ApiTask(this);
42             apiTask.execute(params);
43         } else {
44             System.out.println("NO TOKEN");
45         }
46         touchDown = true;
47     }
48 }
49 else{
50     showHandlerMessage("                !");
51 }
52 }
53
54 }
55 if (withPreview && !key.noKeyPreview()) {
56     showKeyPreview(key);
57 }
58 }

```

Яриаг буфферт бичээд wave тодорхойлолттой болгох классын код

```

1 package com.android.inputmethod.keyboard.voice;
2
3 import android.media.AudioFormat;
4 import android.media.AudioRecord;
5 import android.media.MediaRecorder;
6 import java.io.ByteArrayOutputStream;
7 import java.io.IOException;
8
9 public class BufferRecorder {
10     private static final int RECORDER_BPP = 16;
11     private static final int RECORDER_SAMPLERATE = 44100;
12     private static final int RECORDER_CHANNELS = AudioFormat.
13         CHANNEL_IN_MONO;
14     private static final int RECORDER_AUDIO_ENCODING = AudioFormat.
15         ENCODING_PCM_16BIT;
16     private AudioRecord recorder = null;
17     private int bufferSize = 0;
18     private Thread recordingThread = null;
19     private boolean isRecording = false;
20     private ByteArrayOutputStream buffer;
21     public ByteArrayOutputStream totalByteArray;
22     private byte[] header;
23
24     public BufferRecorder() {
25         bufferSize = AudioRecord.getMinBufferSize(RECORDER_SAMPLERATE,
26             RECORDER_CHANNELS, RECORDER_AUDIO_ENCODING) * 3;
27         buffer = new ByteArrayOutputStream();
28         totalByteArray = new ByteArrayOutputStream();
29     }

```



```

30     public void startRecording() {
31         recorder = new AudioRecord(MediaRecorder.AudioSource.MIC,
32             RECORDER_SAMPLERATE, RECORDER_CHANNELS,
33             RECORDER_AUDIO_ENCODING, bufferSize);
34         int i = recorder.getState();
35         if (i == 1)
36             recorder.startRecording();
37
38         isRecording = true;
39
40         recordingThread = new Thread(new Runnable() {
41             @Override
42             public void run() {
43                 try {
44                     writeAudioDataToFile();
45                 } catch (IOException e) {
46                     e.printStackTrace();
47                 }
48             }
49         }, "AudioRecorder Thread");
50
51         recordingThread.start();
52     }
53
54
55     /**
56     private void writeAudioDataToFile() throws IOException {
57         byte[] recordData = new byte[bufferSize];
58         int read = 0;
59         while (isRecording) {
60             int counter = 0;
61             read = recorder.read(recordData, 0, bufferSize);
62             if (AudioRecord.ERROR_INVALID_OPERATION != read) {
63                 buffer.write(recordData);
64             }
65         }
66     }
67
68     public void stopRecording() throws IOException {
69         long totalAudioLen = 0;
70         long totalDataLen = 0;
71         long longSampleRate = RECORDER_SAMPLERATE;
72         int channels = ((RECORDER_CHANNELS == AudioFormat.
73             CHANNEL_IN_MONO) ? 1
74             : 2);
75         long byteRate = RECORDER_BPP * RECORDER_SAMPLERATE * channels /
76             8;
77         if (null != recorder) {
78             isRecording = false;
79
80             int i = recorder.getState();
81             if (i == 1)

```

```

80         recorder.stop();
81         recorder.release();
82         totalAudioLen = buffer.toByteArray().length;
83         totalDataLen = totalAudioLen + 36;
84         writeWaveFileHeader(totalAudioLen, totalDataLen,
85                             longSampleRate, channels, byteRate);
86         totalByteArray.write(header);
87         totalByteArray.write(buffer.toByteArray());
88         recorder = null;
89         recordingThread = null;
90     }
91 }
92
93 /**          */
94 public void clear(){
95     buffer.reset();
96     totalByteArray.reset();
97     header = null;
98 }
99
100 /**          44-          .wav          */
101 private void writeWaveFileHeader(long totalAudioLen,
102                                  long totalDataLen, long
103                                  longSampleRate, int channels,
104                                  long byteRate)
105 {
106     header = new byte[44];
107     header[0] = 'R'; // RIFF/WAVE header
108     header[1] = 'I';
109     header[2] = 'F';
110     header[3] = 'F';
111     header[4] = (byte) (totalDataLen & 0xff);
112     header[5] = (byte) ((totalDataLen >> 8) & 0xff);
113     header[6] = (byte) ((totalDataLen >> 16) & 0xff);
114     header[7] = (byte) ((totalDataLen >> 24) & 0xff);
115     header[8] = 'W';
116     header[9] = 'A';
117     header[10] = 'V';
118     header[11] = 'E';
119     header[12] = 'f'; // 'fmt ' chunk
120     header[13] = 'm';
121     header[14] = 't';
122     header[15] = ' ';
123     header[16] = 16; // 4 bytes: size of 'fmt ' chunk
124     header[17] = 0;
125     header[18] = 0;
126     header[19] = 0;
127     header[20] = 1; // format = 1
128     header[21] = 0;
129     header[22] = (byte) channels;
130     header[23] = 0;
131     header[24] = (byte) (longSampleRate & 0xff);

```

```

130     header[25] = (byte) ((longSampleRate >> 8) & 0xff);
131     header[26] = (byte) ((longSampleRate >> 16) & 0xff);
132     header[27] = (byte) ((longSampleRate >> 24) & 0xff);
133     header[28] = (byte) (byteRate & 0xff);
134     header[29] = (byte) ((byteRate >> 8) & 0xff);
135     header[30] = (byte) ((byteRate >> 16) & 0xff);
136     header[31] = (byte) ((byteRate >> 24) & 0xff);
137     header[32] = (byte) (((RECORDER_CHANNELS == AudioFormat.
        CHANNEL_IN_MONO) ? 1
138         : 2) * 16 / 8); // block align
139     header[33] = 0;
140     header[34] = RECORDER_BPP; // bits per sample
141     header[35] = 0;
142     header[36] = 'd';
143     header[37] = 'a';
144     header[38] = 't';
145     header[39] = 'a';
146     header[40] = (byte) (totalAudioLen & 0xff);
147     header[41] = (byte) ((totalAudioLen >> 8) & 0xff);
148     header[42] = (byte) ((totalAudioLen >> 16) & 0xff);
149     header[43] = (byte) ((totalAudioLen >> 24) & 0xff);
150 }
151
152
153 }

```

Бичсэн яриаг хөврүүлэх хүсэлт илгээх классын код

```

1 package com.android.inputmethod.keyboard.voice;
2 import android.os.AsyncTask;
3 import android.util.Log;
4
5 import androidx.annotation.NonNull;
6
7 import com.android.inputmethod.keyboard.MainKeyboardView;
8 import org.json.JSONException;
9 import org.json.JSONObject;
10
11 import java.io.IOException;
12 import java.net.SocketTimeoutException;
13 import java.net.UnknownHostException;
14
15 import okhttp3.MediaType;
16 import okhttp3.OkHttpClient;
17 import okhttp3.Request;
18 import okhttp3.RequestBody;
19 import okhttp3.Response;
20
21
22 public class ApiTask extends AsyncTask<TaskParam,Void,String>{
23     MainKeyboardView mainKeyboardView;
24     public String error = "";
25     public ApiTask(MainKeyboardView mainKeyboardView){

```

```

26         this.mainKeyboardView = mainKeyboardView;
27     }
28     String urlString = "https://chimege.mn:9995/transcribe";
29
30     @Override
31     protected String doInBackground(TaskParam... params) {
32         byte[] recordData = params[0].byteArray;
33         String token = params[0].token;
34         OkHttpClient client = new OkHttpClient();
35         String speechToText;
36         RequestBody body = RequestBody.create(recordData);
37         Request request = new Request.Builder()
38             .url(urlString)
39             .header("apiKey", token)
40             .addHeader("Content-Type", "WaveFile")
41             .post(body)
42             .build();
43         try (Response response = client.newCall(request).execute())
44         {
45             String jsonString = response.body().string();
46             System.out.println(jsonString);
47             JSONObject json = new JSONObject(jsonString);
48             if (json.has("success")) {
49                 speechToText = json.getString("text");
50                 if (speechToText.equals("")) {
51                     error = "                !";
52                     System.out.println("speechToText: " +
53                         speechToText);
54                 } else if (speechToText != null) {
55                     return speechToText;
56                 }
57             } else {
58                 error = json.getString("error");
59             }
60         } catch (UnknownHostException e) {
61             error = "                !";
62         } catch (SocketTimeoutException e) {
63             error = "                !";
64         }
65         catch (Exception e) {
66             error = "                ,                !";
67         }
68
69         return null;
70     }
71
72     @Override
73     protected void onPostExecute(String s) {
74         super.onPostExecute(s);
75

```

```
76         if(s == null){
77             mainKeyboardView.showMessage(error);
78         }
79         else{
80             mainKeyboardView.asyncResult(s);
81         }
82     }
83 }
84 }
```