

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

Манлайбаатар Түвшинтөр

**ИЛЧЛЭГ ТООЦОХ, ДЭГЛЭМ БОЛОН
ЖИНГЭЭ ХЯНАХ АППЛИКАЙШН
(APPLICATION FOR DIET AND NUTRITION)**

Програм хангамж (D061302)
Бакалаврын судалгааны ажил

Улаанбаатар

2020 оны 05 сар

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

**ИЛЧЛЭГ ТООЦОХ, ДЭГЛЭМ БОЛОН ЖИНГЭЭ
ХЯНАХ АППЛИКАЙШН
(APPLICATION FOR DIET AND NUTRITION)**

Програм хангамж (D061302)
Бакалаврын судалгааны ажил

Удирдагч: _____ Ч.Доржнамжирмаа

Гүйцэтгэсэн: _____ М.Түвшинтөр (16B1SEAS0727)

Улаанбаатар

2020 оны 05 сар

Зохиогчийн баталгаа

Миний бие Манлайбаатар Түвшинтөр ”ИЛЧЛЭГ ТООЦОХ, ДЭГЛЭМ БОЛОН ЖИНГЭЭ ХЯНАХ АППЛИКЕЙШН“ сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

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

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

УДИРТГАЛ	1
БҮЛГҮҮД	2
1 Сэдвийн судалгаа	2
1.1 Судалгааны зорилго	2
1.2 Судалгааны зорилт	2
1.3 Ижил төстэй програмын судалгаа	3
1.4 Онолын судалгаа	5
1.5 Технологийн судалгаа	10
2 Системийн шинжилгээ ба зохиомж	13
2.1 Хэрэглэгчийн шаардлага	13
2.2 Ажлын явцын диаграм	17
2.3 Классын диаграм	19
2.4 Үйл ажиллагааны диаграм	21
2.5 Дарааллын диаграм	27
3 Хэрэгжүүлэлт	34
3.1 ФатъСикрет платформ (FatSecret Platform API)	34
3.2 Фаярбэс гэрчлэл (Firebase Authentication)	34
3.3 Фаярбэс өгөгдлийн сан (Firebase Realtime Database)	34
3.4 Хэрэглэгчийн интерфейс	35
ДҮГНЭЛТ	59
НОМ ЗҮЙ	59
ХАВСРАЛТ	60
A Үучилсэн төлөвлөгөө	61
B Кодын хэрэгжүүлэлт	62

ЗУРГИЙН ЖАГСААЛТ

2.1	Ажлын явцын диаграм	17
2.2	Классын диаграм	19
2.3	Хэрэглэгч нэвтрэх болон бүртгүүлэх үйл ажиллагааны диаграм	21
2.4	Жин бүртгэх болон өөрчлөх үйл ажиллагааны диаграм	23
2.5	Усны бүртгэл хийх үйл ажиллагааны диаграм	24
2.6	Хоолны төлөвлөгөө боловсруулах үйл ажиллагааны диаграм	26
2.7	Хэрэглэгч нэвтрэх дарааллын диаграм	27
2.8	Хэрэглэгч бүртгүүлэх дарааллын диаграм	28
2.9	Жин бүртгэх үйл дарааллын диаграм	29
2.10	Усны бүртгэл хийх дарааллын диаграм	31
2.11	Хоолны төлөвлөгөө боловсруулах дарааллын диаграм	32
3.1	Хэрэглэгч нэвтрэх хуудас	35
3.2	Зорилгоо сонгох хуудас	36
3.3	Хүйс сонгох хуудас	37
3.4	Хөдөлгөөний идэвхи сонгох хуудас	38
3.5	Жингээ оруулах хуудас	39
3.6	Өндөрөө оруулах хуудас	40
3.7	Төрсөн огноо сонгох хуудас	41
3.8	Зорилтот жин сонгох хуудас	42
3.9	Хэрэглэгч бүртгүүлэх хуудас	43
3.10	Боломжит хугацаа болон илчлэгийг санал болгох хуудас	44
3.11	Нүүр хуудас	45
3.12	Усны бүртгэл хуудас	46
3.13	Хоолны төлөвлөгөө болон түүх хуудас	47
3.14	Хоолны дэлгэрэнгүй хуудас	48

3.15 Хүнс хайх хуудас	49
3.16 Хүнсний дэлгэрэнгүй хуудас	50
3.17 Хүнсний хэмжээг тохируулах хуудас	51
3.18 Хялбар цэс цонх	52
3.19 Жин бүртгэх хуудас	53
3.20 Тохиргоо хуудас	54
3.21 Хувийн мэдээллийн хуудас	55
3.22 Шимт тэжээллэг чанарын хуудас	56
3.23 Усны тохиргоо хуудас	57
3.24 Сануулгын тохиргоо хуудас	58
A.1 Үечилсэн төлөвлөгөө	61

Кодын жагсаалт

B.1	Цахим шуудангаар нь шинэ хэрэглэгч үүсгэж буй код	62
B.2	Хэрэглэгч нэвтэрч буй код	62
B.3	Хэрэглэгч Google хаягаа ашиглан нэвтэрч буй код	63
B.4	Хэрэглэгч Facebook хаягаа ашиглан нэвтэрч буй код	64
B.5	Хэрэглэгчийн өдөрт авах илчлэгийн хэмжээг тооцож буй код	65
B.6	Хэрэглэгчийн зорилгод үндэслэн боломжтой хугацаа болон илчлэгийн сонголтуудыг тооцож буй код	66
B.7	Хоолны төлөвлөгөө болон түүхийн хэсгийн код	67
B.8	Хэрэглэгчийн хайсан хүнсний жагсаалтыг fatsecret -ийн өгөгдлийн сангаас авч хэсгийн код	80
B.9	Хэрэглэгчийн өмнөх түүхийг унших хэсгийн код	81
B.10	Хүнсний хэмжээг шинэчлэх хэсгийн код	82
B.11	Хүнсийг цэснээс хасах хэсгийн код	82
B.12	Хүнсийг цэсэд нэмэх хэсгийн код	82
B.13	Хайх хүнсний url бэлдэх хэсгийн код	83
B.14	Хүнсний дэлгэрэнгүйг авах url бэлдэх хэсгийн код	83
B.15	Сануулга эхлүүлэх хэсгийн код	83
B.16	Сануулга зогсоох хэсгийн код	83

УДИРТГАЛ

Сэдэв сонгосон үндэслэл

Хэчнээн хүн өөрийн өдрийн илчлэг болон усны хэмжээ цаашлаад шимт тэжээлийн чанарыг тодорхойлж чаддаг бол ?

Эрүүл мэндийн судалгаанаас үзэхэд өөрийн жингээ тэмдэглэж явснаар хүссэн үр дүндээ илүү хурдан хүрдэг байна. Ус бол хүний биед нэн шаардлагатай зүйлсийн нэг билээ. Тэгвэл усыг бага эсвэл хэт их ууснаар эрүүл мэндэд аюултай тул өөрийн өдрийн усны хэмжээг зөв тодорхойлж тодорхой цагуудад уух нь чухал юм.

Хүн өдөрт хэрэглэх илчлэгийн хэмжээгээ зөв олж түүндээ тохирсон хоол хүнс хэрэглэх нь маш чухал. Энэ бүхнийг нэг дор хийдэг гар утасны аппликешион ховор мөн монгол хэл дээр хөгжүүлэгдээгүй тул тус сэдвийг сонгосон болно.

Дипломын ажлын зорилго

Тус судалгааны ажил нь хэрэглэгч өөрийн жингээ хянадаг, өдрийн илчлэг болон усны хэмжээг тодорхойлж түүндээ тохирсон хоол хүнсний төлөвлөгөө гаргаж болдог хэрэглэгчийн эрүүл мэндийг дэмжих зорилготой юм.

Дипломын ажлын зорилт

- RDI, BMR, BMI, BFP, TDEE зэрэг холбоотой сэдвүүдийг судлах.
- Ижил төстэй аппликэйшнүүдийг судлах.
- Системийн шинжилгээ зохиомж хийх.
- Гар утасны програмчлалын технологийг судалж, эзэмших.
- Хөгжүүлэлт хийх.

1. Сэдвийн судалгаа

1.1 Судалгааны зорилго

Бакалаврын судалгааны ажлын хүрээнд хийж гүйцэтгэх програмд бусад ижил төстэй програмаас ялгараах давуу тал, авч хэрэгжүүлж болохуйц зүйлс мөн тэдгээр програмуудын ажиллах зарчим, бүтэц зэргийг судлах явдал юм.

1.2 Судалгааны зорилт

Ижил төстэй програмын судалгаа хийхийн тулд дараах зорилтуудыг тавин хийлээ. Үүнд:

- Ижил төстэй мобайл аппликацийний давуу болон сүл талыг судлах
- Ижил төстэй мобайл аппликацийний үйл ажиллагаа, ажиллах зарчим судлах
- Ижил төстэй мобайл аппликацийний хэрэглэгчийн график интерфейсийг судлах
- Онолын судалгаа хийх
 - Хүний өдрийн шингэн болон илчлэгийн хэмжээг судлах
 - Шим тэжээлт чанаарын талаар судлах
- Технологийн судалгаа хийх
 - Java технологи судлах
 - Android studio судлах

1.3 Ижил төстэй програмын судалгаа

1.3.1 ВэеDrop (*WeightDrop*)

Тухайн апликеийшн нь хэрэглэгчийн өгсөн мэдээлэл дээр үндэслэж өдрийн илчлэгийг тооцох мөн жингээ хянахад туслах үүрэгтэй. Нүүр хуудсанд анхны жин, одоогийн жин, зорилтот жинг шугаман графикийн хамт харуулсан байна. Түүхийг харуулсан хуудсанд анх жингээ бүртгэсэн хугацаанаас одоог хүртэл бүртгэл хоорондын жингийн зөрүү харуулсан байна. Жин бүртгэх хэсэг нь огноо болон хэмжүүр гэсэн цомхон байна. Өдөр бүр тогтсон цагт жингээ бүртгэхэд туслах сануулга үүсгэх боломжтой. Мөн тус апликеийшн нь метр болон империал нэгжийг дэмждэг.

1.3.2 Мая Вэе (*My Weight*)

Жингээ тогтмол оруулж, ахиц дэвшилийг статистик, графикаар хянах боломжтой тус апликеийшн нь олон профайлыг оруулж, хянах боломжтой. Гэр бүлийн гишүүд, хань ижил, найз нөхөд, хамт ажиллагсдынхаа ахиц дэвшилийг хянаж, эсвэл зөвхөн өөрийнхөө үндсэн зорилгодоо хүрэх хүртэл хэсэгчилсэн зорилгоо биелүүлэхийн тулд зөвхөн 1-ээс олон тооны профайл үүсгэж болно. Энэ апликеийшн нь таны хамгийн тохиромжтой жинг санал болгож, одоогийн жин, өндөр, хүйс, нас зэргээс хамааран тухайн жинд хүрэх тохиромжтой хугацааг өгөх болно. Хэрэглэгч мэдээж өөрийн төлөвлөгөөнд үндэслэн санал болгосон зорилтуудыг өөрчилж болно.

Үндсэн онцлог:

- Нэг буюу олон профайлыг хянадаг
- Жингээс гадна биеийн янз бүрийн хэмжилтийг хянах
- Хэмжилтийг паунд, кило, стони, фийт, метр, калори эсвэл жоуль нэгжүүдээр харуулна

- Нас, өндөр, хүйс, биетийн хүрээ дээр тулгуурлан хамгийн тохиромжтой жинг тооцоолно
- Цаг хугацаа өнгөрөхөд жингээ хасах нийт явцыг харуулдаг бүтэн дэлгэцийн график
- BMI (Биетийн массын индекс) -ийг тооцоолсон
- Өдөр тутмын илчлэгийн хэмжээг тооцоолно
- Биетийн өөхний хувийг тооцоолно
- Нийт жингийн өөрчлөлтийг тооцоолсон
- Өдрийн дундаж өөрчлөлтийг тооцоолсон
- Долоо хоногийн дундаж өөрчлөлтийг тооцов
- Ерөнхий өөрчлөлтийг тооцоолсон
- Өмнөх өдрүүдэд шинэ жин оруулах боломжтой
- Өдөр тутмын ахиц дэвшлийн чанарыг харуулдаг үзүүлэлтүүдийг харуулна
- Нууцлалыг хангахын тулд PIN түгжээ
- Үлдсэн жинг App Icon дээр харуул
- Олон хэлний хэлний нутагшуулалт 15 гаруй хэлийг дэмжинэ
- Бичсэн бүх мэдээллээ экспортлох боломжтой

1.3.3 Дэели Водэр (*Daily Water*)

Тус аппликашн нь хэрэглэгчийн өдөрт уух шингэний хэмжээг тооцож цаг тухайд нь сануулах үүрэгтэй юм.

Үндсэн онцлог:

- Өдөр тутмын уух шингэний хэмжээгээ тооцож хянах

- Уусан шингэнээ бүртгэх
- Уух бүртээ нэг даралтаар бүртгэх боломжтой
- Уусан аяганы тоогоо харах боломжтой
- Аяганы хэмжээг тохируулах боломжтой
- Уух бүртээ хэр хэмжээний ус уухаа тохируулга /Аяганы 1/4, 1/2 эсвэл бүтэн аяга/
- Шингэн уух хувиар гаргаж сануулга үүсгэх боломжтой
- 10 өөр үнэгүй сануулгын аятай
- Түүхийн хэсэгт өдөр, долоо хоног эсвэл сарын уусан шингэний хэмжээг харуулна
- Тухайн өдөрт уусан устай аяганы тоог app icon дээр харуулна
- Хэмжилтийг oz, ml нэгжүүдээр харуулна

1.4 Онолын судалгаа

1.4.1 Илчлэг тооцох

1.4.1.1 Илчлэг

Хоол хүнсийг идэж хэрэглэхэд ялгарч бие махбодыг үйл хөдөлгөөн хийх, биохимийн хувирлуудыг явуулахад шаардагдах дулааныг ЭНЕРГИ гэдэг. Хоолны хүнсний энергийг ИЛЧЛЭГ гэдэг ба килокалори (ккал) гэдэг нэгжээр илэрхийлдэг. 1 ккал гэдэг нь 1 кг усыг энгийн агаарын даралтад 1 0C халаахад шаардагдах дулааны энергийг хэлнэ.

Илчлэгийг хэмждэг нэгжүүд:

1. КилоКалор – Ккал

2. КилоДжоуль – кДж

3. 1ккал=4.18кДж

Хоол хүнснээс авч болох илчлэгийн хэмжээг тооцоолоходоо түүнд агуулагдах энерги ялгаруулах шимт бодис тус бүрийн хэмжээг харгалзах илчийн утгаар үржүүлж, үр дүнг хооронд нь нэмэх байдлаар нийт илчлэгийн хэмжээг тодорхойлдог

Илч ялгаруулдаг шимт бодис:

- Уураг – 1г/4ккал
- Θөх тос – 1г/9ккал
- Нүүрс ус – 1г/4ккал
- Эслэг – 1г/2ккал
- Спирт (этанол) – 1г/7ккал

Илчлэг (ккал) = (Уураг (гр) x 4)+ (Нүүрс ус (гр) x 4)+ (Θөх тос (гр) x 9)+ Эслэг (гр) x 2)

1.4.1.2 Өдөрт авах илчлэгийн хэмжээг тооцох арга

Нарийвчилсан хоол тэжээлийн арга (The Precision Nutrition formula)

Энэ арга нь хөдөлгөөний идэвхээс хамаарч илчлэгийн хэмжээг нь биеийн жингийн нэг фунт тутамд 14-16 илчлэгээр үржүүлж тооцох юм.

Жишээлбэл :

- 70кг хүн
- $70\text{kg} \times 2.2 = 154$ фунт
- $154 \times 14 = 2156$ ккал
- $154 \times 16 = 2464$ ккал
- Иймд 70кг хүний өдөрт авах илчлэгийн хэмжээ 2156 -аас 2464 юм.

Кэтч-МакАрдл -ийн арга (Katch-McArdle formula)

Бодох алхамууд:

- Basal Metabolic rate (BMR) буюу идэвхтэй хөдөлгөөн хийхгүйгээр амьдрахад шаардлагатай илчлэгийн хэмжээг олно.
- BMR -ийг олсон бол идэвхтэй хөдөлгөөний тогтмолоор үржүүлнэ.
- Эрэгтэй эмэгтэй хүмүүсийг адилхан томъёогоор бодно.
- Биеийн өөхлөлтийн хувийг мэдэж байх хэрэгтэй.

Томъёо:

- $LBM = \text{жин} \times LBM\%$
- $BMR = (21.6 \times LBM) + 370$

Жишээлбэл:

- 70кг жин 24%-н биеийн өөхлөлттэй хүн
- $LBM\% = 76\% (100\% - 24\% \text{ хасна})$
- $LBM \text{ кг} = 70\text{kg} \times 0.76 = 53.2\text{kg}$
- $BMR = (21.6 \times 53.2) + 370 = 1519 \text{ ккал}$
- 1519 ккал - энэ нь дасгал хөдөлгөөнийг тооцоогүйгээр амьдралын үндсэн функцийг хадгалах хэмжээ юм.
- Нийт энерги зарцуулалт (TDEE) = BMR X 1.55 (5 өдрийн бэлтгэлийн тогтмол) = 2355
- Жин хасахийн тулд та TDEE-ийн тодорхой хувийг хасах хэрэгтэй бөгөөд ингэснээр таны зарцуулж буй илчлэг тань гаднаас авч байгаагаас их болох юм.

- Өдөрт 10% хасах = 2119 калори
- Өдөрт 15% хасах = 2002 калори

Миффлин Сент Жеор -ийн арга (The Mifflin St. Jeor formula)

Бодох алхамууд:

- Basal Metabolic rate (BMR) буюу идэвхтэй хөдөлгөөн хийхгүйгээр амьдрахад шаардлагатай илчлэгийн хэмжээг олно.
- BMR олсон бол идэвхтэй хөдөлгөөний тогтмолоор үржүүлнэ
- Эрэгтэй эмэгтэй хүмүүсийг ялгаатай томъёогоор бодно
- Биеийн өөхлөлтийн хувийг мэдэж байх шаардлагагүй.

Эрэгтэй хүний томъёо:

- $((10 \times \text{жин(кг)}) + (6.25 \times \text{өндөр(см)}) - (5 \times \text{нас}) + 5) \times \text{тогтмол}$

Эмэгтэйн томъёо:

- $((10 \times \text{жин(кг)}) + (6.25 \times \text{өндөр(см)}) - (5 \times \text{нас}) - 161) \times \text{тогтмол}$

Эрэгтэй хүний жишээ:

- 75кг жинтэй
- өндөр 170см
- 25 нас
- $BMR = (10 \times 75) + (6.25 \times 170) - (5 \times 25) + 5 = 1692$
- $1692 \times 1.55(3-5 \text{ өдрийн дасгалын тогтмол}) = 2623 \text{ ккал}$
- Өдөрт 10% хасах = 2361 ккал
- Өдөрт 10% нэмэх = 2885 ккал

1.4.2 Биеийн жингийн индекс тооцох (BMI)

$$BMI = \frac{weight(kg)}{height^2(m)}$$

1.4.3 Биеийн өөхлөлтийн хувь тооцох (BFP)

Арга-1

Эмэгтэй хүний томъёо:

- $BFP = \frac{495}{1.0324 - 0.19077 \times log10(waist-neck) + 0.15456 \times log10(weight)} - 450$

Эрэгтэй хүний томъёо:

- $BFP = \frac{495}{1.29579 - 0.35004 \times log10(waist+hip-neck) + 0.22100 \times log10(weight)} - 450$

Арга-2

Эрэгтэй хүний томъёо:

- $BFP = 1.20 \times BMI + 0.23 \times Hac - 16.2$

Эмэгтэй хүний томъёо:

- $BFP = 1.20 \times BMI + 0.23 \times Hac - 5.4$

Эрэгтэй хүүхдийн томъёо:

- $BFP = 1.51 \times BMI - 0.70 \times Hac - 2.2$

Эмэгтэй хүүхдийн томъёо:

- $BFP = 1.51 \times BMI - 0.70 \times Hac + 1.4$

1.4.4 Өдөрт авах шингэний хэмжээ тооцох

Биеийн цэвэр жин = жин [кг] x (1 - (биеийн өөхлөлт [%]/ 100))

Хөдөлгөөний идэвхтэй байдлаас хамаарч дараах байдлаар бодно.

- Бага - биеийн цэвэр жин х 0.65
- Дунд - биеийн цэвэр жин х 0.75
- Их - биеийн цэвэр жин х 0.85

1.5 Технологийн судалгаа

1.5.1 *Андройд студио (Android Studio)*

Андройд студио нь JetBrains-ийн IntelliJ IDEA програм хангамж дээр суурилан бүтээгдсэн, Android-д тусгайлан зориулагдсан Google-ийн Android үйлдлийн системд зориулагдсан албан ёсны хөгжүүлэлтийн орчин (IDE) юм.

1.5.2 *Жава (Java)*

Java бол Sun Microsystems корпорац -ийн бүтээсэн програмчлалын хэл бөгөөд хамгийн анх 1995 онд түүнийг олон нийтэд танилцуулжээ. Анх хийгдэхдээ Sun-ийн Java платформ дээр ажиллахаар байсан юм. Жава-г бүтээхдээ С ба C++ хэлний онцлог, бичиглэлийг хүчтэй тусган авчээ. Гэхдээ хамгийн гол нь объект хандалтат загварт түлхүү анхаарсан ба доод төвшний програмчлалын боломжийг бага оруулсан байна. Java програм нь бит биш байт кодуудад хөрвөн ажилладаг бөгөөд энэхүү байт кодуудыг зөвхөн Java Virtual Machine (JVM) хэмээх систем ажиллуулдаг юм. Харин JVM нь платформ болгонд зориулагдан тусдаа хийгдсэн байдаг тул таны бичсэн програм, код аль ч үйлдлийн систем, платформ дээр ажиллах болж байгаа нь түүний хамгийн өвөрмөц онцлог юм.

1.5.3 *Андройд үйлдлийн систем*

Андройд(Android) бол Линукс кернел (Linux Kernel) болон бусад нээлттэй эх кодтой програмуудаас бүтсэн үйлдлийн систем бөгөөд мэдрэгчтэй дэлгэцтэй ухаалаг утас болон таблет зэрэгт зориулж бүтээсэн. Андройд нь Google болон Open Hand Alliance гэх

хөгжүүлэгчдийн холбоо хөгжүүлсэн үйлдлийн систем юм. Анх Андройдийг ”Android Inc” хөгжүүлсэн ба 2005 онд ”Google” худалдаж авсан ба 2007 онд танилцуулсан бөгөөд 2008 онд анхны төхөөрөмжөө худалдаанд гаргасан юм. Одоогийн хувилбар нь Android 10 бөгөөд 2019 оны 9 сарын 3нд гарсан юм. Андройдийн цөм код ”Android Open Source Project (AOSP)” нь анх ”Apache License” –ийн лицензээр гарсан юм.

1.5.4 Фаярбэс (Firebase)

Firebase бол Backend-as-a Service (BaaS) юм. Энэ нь хөгжүүлэгчдэд чанартай апп хөгжүүлэх, хэрэглэгчээ өсгөх, ашиг олоход нь туслах олон төрлийн хэрэгсэл, үйлчилгээ үзүүлдэг.

Firebase нь NoSQL өгөгдлийн санд хамаардаг ба өгөгдлийг JSON бүтцээр хадгалдаг.

Гол онцлогууд

- Authentication - Хэрэглэгчдэд нууц үг, утасны дугаар, Google, Facebook, Twitter болон бусад зүйлийг ашиглан нэвтрэх хандах эрх олгох үйлчилгээ үзүүлдэг үйлчилгээ юм.
- Realtime database - Firebase -ийн “backend” болон бодит хугацааны өгөгдлийн сангийн үйлчилгээг хангадаг бүтээгдэхүүн юм.
- Hosting - Вэб программыг хурдан байршуулах боломжийг олгодог.
- Test lab - Тус программыг Google-ийн мэдээллийн төвд байрладаг виртуал болон физик төхөөрөмж дээр туршиж үзэх боломжтой.
- Notifications - Нэмэлт кодгүйгээр мэдэгдэл илгээх боломжтой.

1.5.5 Фаярбэс агуулах (Firebase Storage)

“Firebase Storage” нь “Firebase” аппликэйшинүүдэд ямар ч сүлжээтэй үед файлыг хамгаалалттайгаар татах болон хуулах(upload) боломжийг олгодог. Хөгжүүлэгч нар үүн дээр зураг, дүрс бичлэг, дуу болон бусад төрлийн контентийг ашиглах боломжтой. “Firebase Storage” нь “Google Cloud Storage”-р хамгаалагдсан байдаг.

1.5.6 Гүүгл хаягаар нэвтрэх (Google Sign-In)

“Google Sign-In” нь хэрэглэгч болон хөгжүүлэгч нарт хамгийн энгийн бөгөөд хялбар байдлаар хаягаа холбох болон үүсгэх нөхцөлийг бүрдүүлж өгдөг. Энэ нэвтрэх үйлдлээр дамжуулан хэрэглэгчийн цахим шуудангийн хаягийн нэр, профайл зураг болон цахим шуудангийн хаягийг нь авах боломжтой. Үүнийг хэрэгжүүлэхийн тулд та хэрэглэгчээс “Google Profile” руу нь нэвтрэх зөвшөөрөл асуун зөвшөөрүүлснээр профайл мэдээллийг нь ашиглах боломжтой болно.

1.5.7 Фэйсбуук хаягаар нэвтрэх (Facebook Login)

“Facebook Login” нь олон платформ хооронд аппликэйшинаар нэвтрэх болон хаяг нээх хурдан ба хэрэглэхэд хялбар арга зам, боломжийг олгодог. “IOS, Android, Web, Desktop” бас ухаалаг зурагт болон бусад интернет ашигладаг төхөөрөмжүүд дээр ашиглах боломжтой. “Facebook Login” нь нэвтрэх ”authentication” болон зөвшөөрөл ”permissions” (хэрэглэгчийн мэдээлэлд хандах эрх авах) гэсэн 2 хэсгээс бүрддэг. Ингэснээр “Facebook Login” -ийг өөрийн аппликэйшний нэвтрэлт болон өгөгдөл хандах байдлыг хялбарчлахын тулд ашиглах юм.

1.5.8 ФатъСикрет платформ (FatSecret Platform API)

Хоол хүнс, тэжээлийн талаарх нарийвчилсан мэдээллийн томоохон өгөгдлийн сан юм.

2. Системийн шинжилгээ ба зохиомж

2.1 Хэрэглэгчийн шаардлага

2.1.1 Алсын хараа, Зорилго тодорхойлолт

- Тухайн систем нь хэрэглэгчийн өдөрт авах болон авсан илчлэгийн хэмжээг тооцож харуулна.
- Хэрэглэгч тухайн илчлэгт тохирсон хоол хүнсийг сонгож хоолны төлөвлөгөөг гаргана.
- Тухайн систем нь хэрэглэгчийн жинг бүртгэж авна
- Тухайн систем нь хэрэглэгчийн бүртгэсэн жинг диаграм байдлаар харуулна.
- Тухайн систем нь хэрэглэгчийн өдөрт уух болон уусан усны хэмжээг тооцож харуулна.
- Хэрэглэгч тухайн системийг ашиглан өөрт тохирсон хоолны төлөвлөгөөг гаргадаг байна.
- Тухайн систем нь хэрэглэгч өөрийн биеийн жинг хянах мөн хүссэн зорилгод нь хүрэх боломжийг олгодог байна.

2.1.2 Суурь нөхцөл

- Тухайн програм бол хэрэглэгчийн эрүүл мэндэд туслах үүрэгтэй андройд үйлдлийн системтэй гар утасны програм юм.
- Зорилтод бүлэг бол 15 -аас 80 хүртэлх насны хүн бүхэн юм.

2.1.3 Системийн орчин, тойм

- Систем нь 7 хоногийн 24 цаг бэлэн байна.

2.1.4 Функциональ шаардлага

- Систем нь хэрэглэгчийн одоогийн биеийн жин, анхны биеийн жин болон зорилтод жинг харуулах хэрэгтэй.
- Систем нь хэрэглэгчийн биеийн жинг огноотой нь бүртгэж байх хэрэгтэй.
- Систем нь хэрэглэгчийн анхны биеийн жин одоогийн биеийн жингийн ялгааг харуулах хэрэгтэй.
- Систем нь хэрэглэгчийн зорилтот жиннд хүрэх боломжтой хугацааг санал болгох хэрэгтэй.
- Систем нь хэрэглэгчийн зорилтот жиннд хүрэхэд шаардлагатай өдрийн илчлэгийг санал болгох хэрэгтэй.
- Систем нь хэрэглэгчийн зорилтот жиннд хүрэх зорилтот хугацааг бүртгэх хэрэгтэй.
- Систем нь хэрэглэгчийн биеийн жингийн өөрчлөлтийн үйл явцыг зорилтот хугацаатай харьцуулсан байдлаар харуулах хэрэгтэй.
- Систем нь хэрэглэгчийн анхны биеийн жингээс одооны биеийн жин хүртэл хэрхэн өөрчлөгдөж буйг диаграм байдлаар харуулах хэрэгтэй.
- Систем нь хэрэглэгчид өглөөний цай, өдрийн хоол, оройн хоол, хөнгөн зууш гэсэн үндсэн цэсний дагуу хоолны төлөвлөгөө боловсруулах боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид тухайн цэс болгон дээр хоол хүнс хайж нэмэх боломжийг олгосон байх хэрэгтэй.

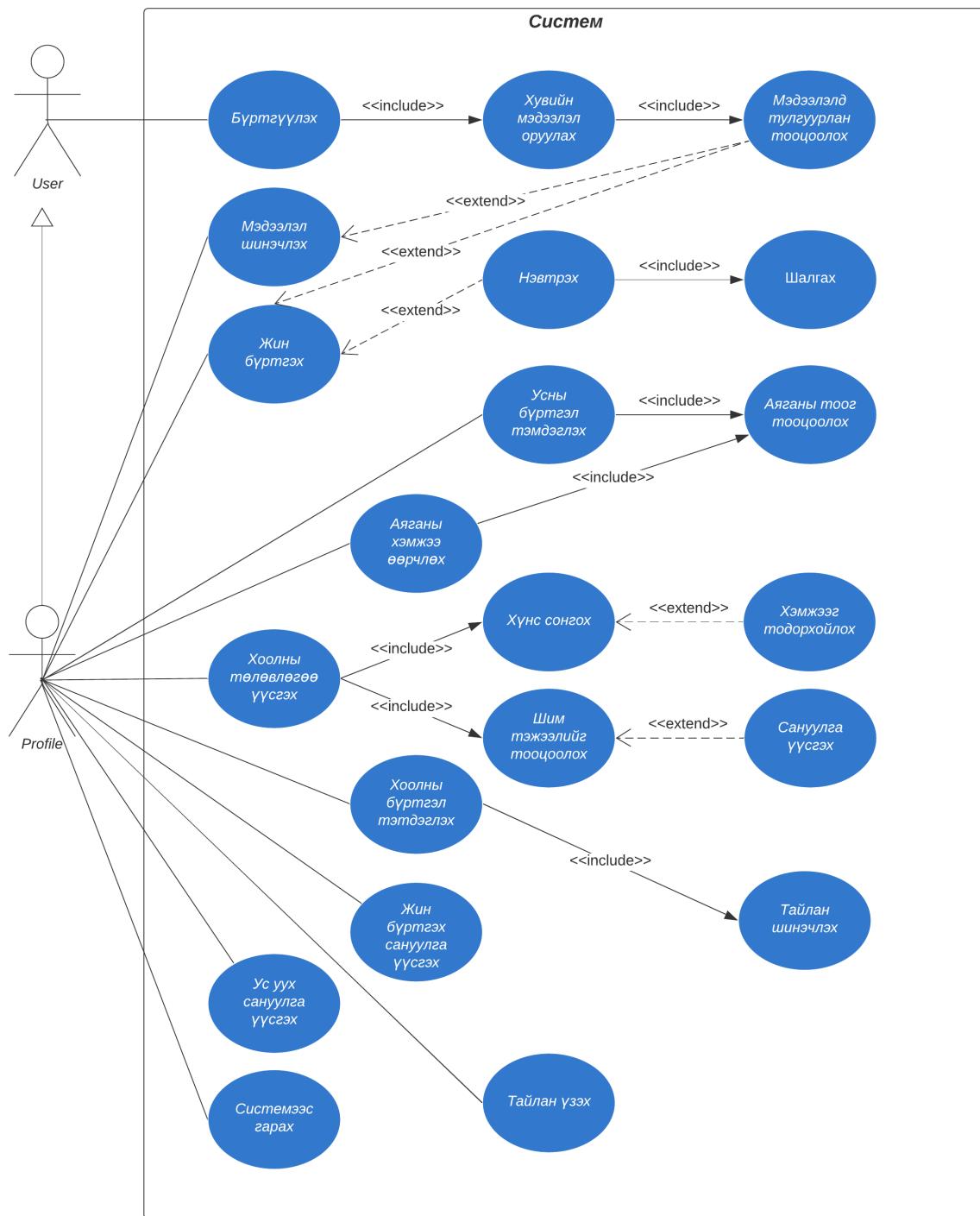
- Систем нь хэрэглэгчид тухайн цэс болгон дээр сэргүүлэг тавих боломжийг олгосон байх хэрэгтэй.
- Систем нь хоол хүнсний хайлтын илэрц дээр тухайн хоол хүнсний орц найрлага болон шим тэжээлт чанарыг харуулах хэрэгтэй.
- Систем нь сонгосон хүнсний илчлэгийг тухайн өдрийн илчлэгээс хасч харуулах ёстой.
- Систем нь тухайн өдрийн хоол хүнсний шим тэжээлт чанар болох уураг, нүүрс ус, өөх тосны хэмжээг харуулах ёстой.
- Систем нь цэс тус бүрээр нь ичлэг болон шим тэжээлт чанар болох уураг, нүүрс ус, өөх тосны хэмжээг харуулах ёстой.
- Систем нь хүнсний ичлэг болон шим тэжээлт чанар болох уураг, нүүрс ус, өөх тосны хэмжээг харуулах ёстой.
- Систем нь хүнсний шим тэжээлт чанар болох уураг, нүүрс ус, өөх тосны хэмжээг дотор нь нарийн задалж харуулах ёстой.
- Систем нь хэрэглэгчийн түүхийг харуулах ёстой.
- Систем нь сонгосон хугацаан дахь түүхийг харуулахдаа илчлэгийн хэмжээ, шим тэжээлт чанар болон идсэн хоол хүнсийг харуулах ёстой.
- Систем нь хэрэглэгчид өөрийн мэдээллийг шинэчлэх боломжийг олгох ёстой.
- Систем нь нүүр, ус, хоолны төлөвлөгөө болон тохиргоо гэсэн үндсэн хэсгүүдээс бүрдэх ёстой.
- Систем нь хэрэглэгч хүнс хайх хэсэгт тухайн цэсэнд өмнө нь сонгосон хүнсний жагсаалтыг харуулах хэрэгтэй.

- Систем нь хэрэглэгчид өмнөх өдрүүдийн хоолны төлөвлөгөөг засах боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид хоолны төлөвлөгөөг урьдчилж гаргах боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид зорилгоо шинэчлэх боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид усны хэмжээ болон аяганы хэмжээг тохируулах боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид жин бүртгэх болон ус уух сануулга үүсгэх боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид шим тэжээлт чанарын хэмжээгээ тохируулах боломжийг олгосон байх хэрэгтэй.
- Систем нь хэрэглэгчид хоол тус бүрээс авах илчлэгийн хэмжээг тохируулах боломжийг олгосон байх хэрэгтэй.

2.1.5 Функциональ бүс шаардлага

- Зөвхөн интернет сүлжээнд ажилладаг байх.
- Зөвхөн андройд үйлдлийн системтэй ухаалаг гар утсанд суурilan ажилладаг байх.
- Андройд үйлдлийн системийн хувилбар 6.0 болон түүнээс дээш хувилбартай андройд үйлдлийн системтэй утсан дээр ажилладаг байх.

2.2 Ажлын явцын диаграм

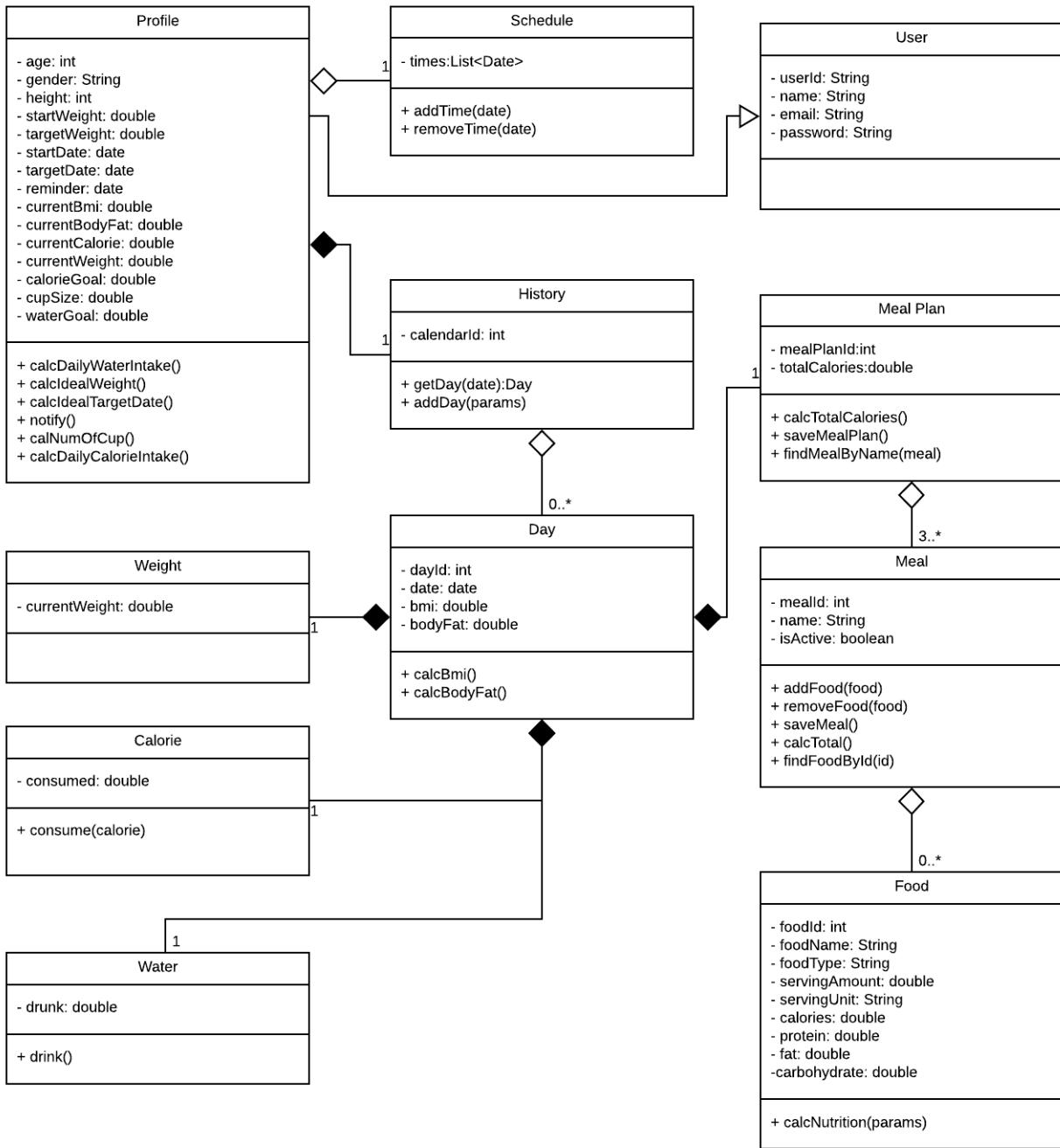


Зураг 2.1: Ажлын явцын диаграм

Тайлбар

- Бүртгэлгүй хэрэглэгч нь бүртгүүлнэ.
- Бүртгүүлэх явц нь шаардлагатай мэдээллийг боловсруулах явцаас бүрдэнэ.
- Бүртгэлтэй хэрэглэгч нь жин бүртгэх, ус бүртгэх, хоолны төлөвлөгөө гаргах сануулга үүсгэх, мэдээллээ засварлах гэсэн үндсэн эрхтэй байна.
- Усны бүртгэл хийх явц нь уусан болон үлдсэн аяганы тоог тооцоолох үйлдлээс бүрдэнэ.
- Хоолны төлөвлөгөө үүсгэх нь хүнс сонгох явцаас бүрдэнэ.
- Хайлтын үр дүнгээс хүнс сонгоходоо тухайн хүнсний хэмжээг тохируулж болно.
- Хоолны төлөвлөгөө үүсгэх нь шим тэжээлт чанарыг тооцоолох явцаас бүрдэнэ.

2.3 Классын диаграмм



Зураг 2.2: Классын диаграмм

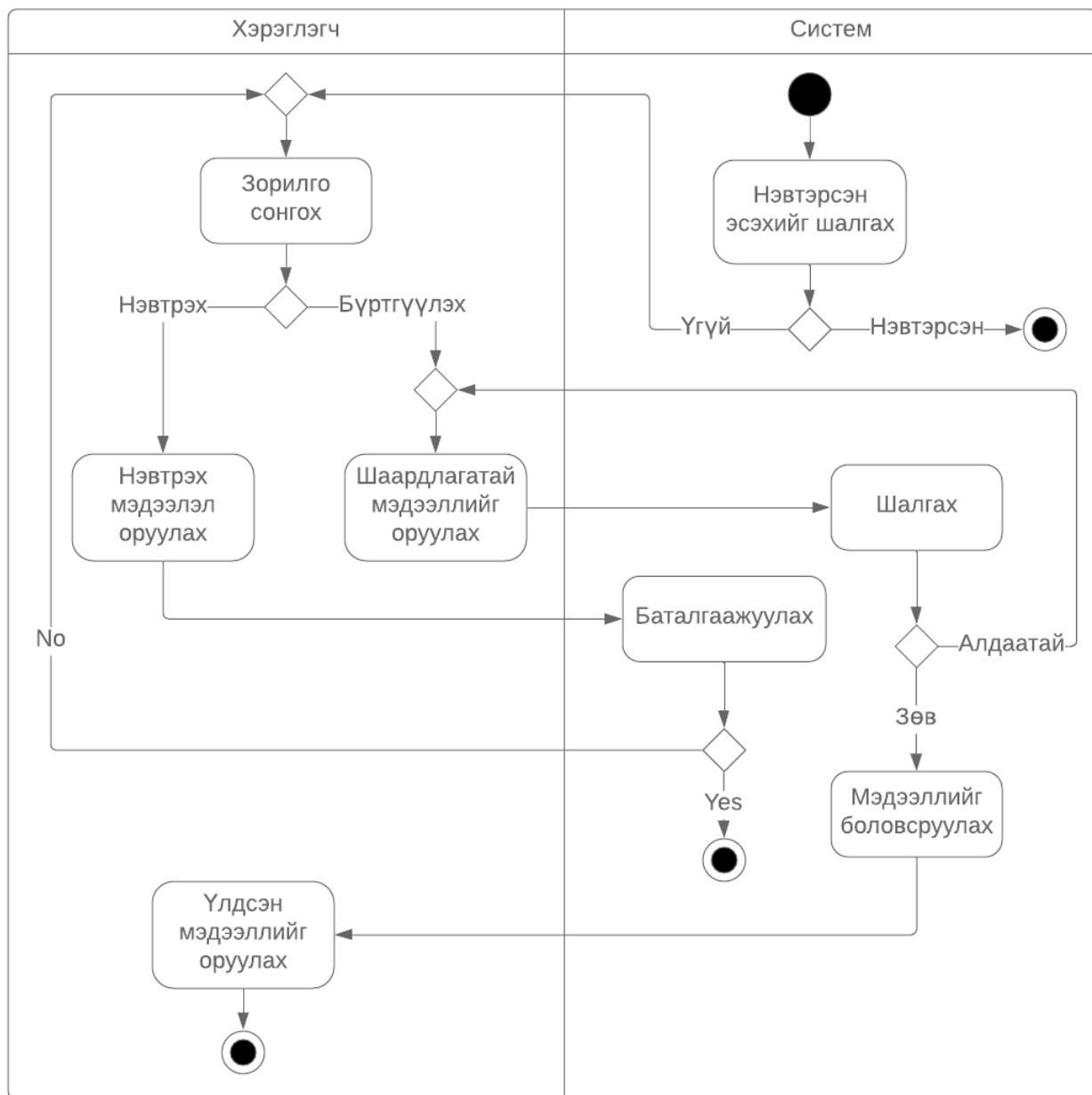
Тайлбар

- Бүртгэлтэй хэрэглэгч нь хэрэглэгчээс удамшина.

- Бүртгэлтэй хэрэглэгч нь үүсэхдээ түүхтэй үүсэх ба түүх нь цөм класс болох өдрийг агуулна.
- Өдөр класс нь үүсэхдээ хоосон, ус, илчлэг болон хоолны төлөвлөгөөтэй үүснэ.
- Өдөр класс үүсэхдээ одоогийн жинтэй жин классыг үүсгэнэ.

2.4 Үйл ажиллагааны диаграм

2.4.1 Хэрэглэгч нэвтрэх болон бүртгүүлэх

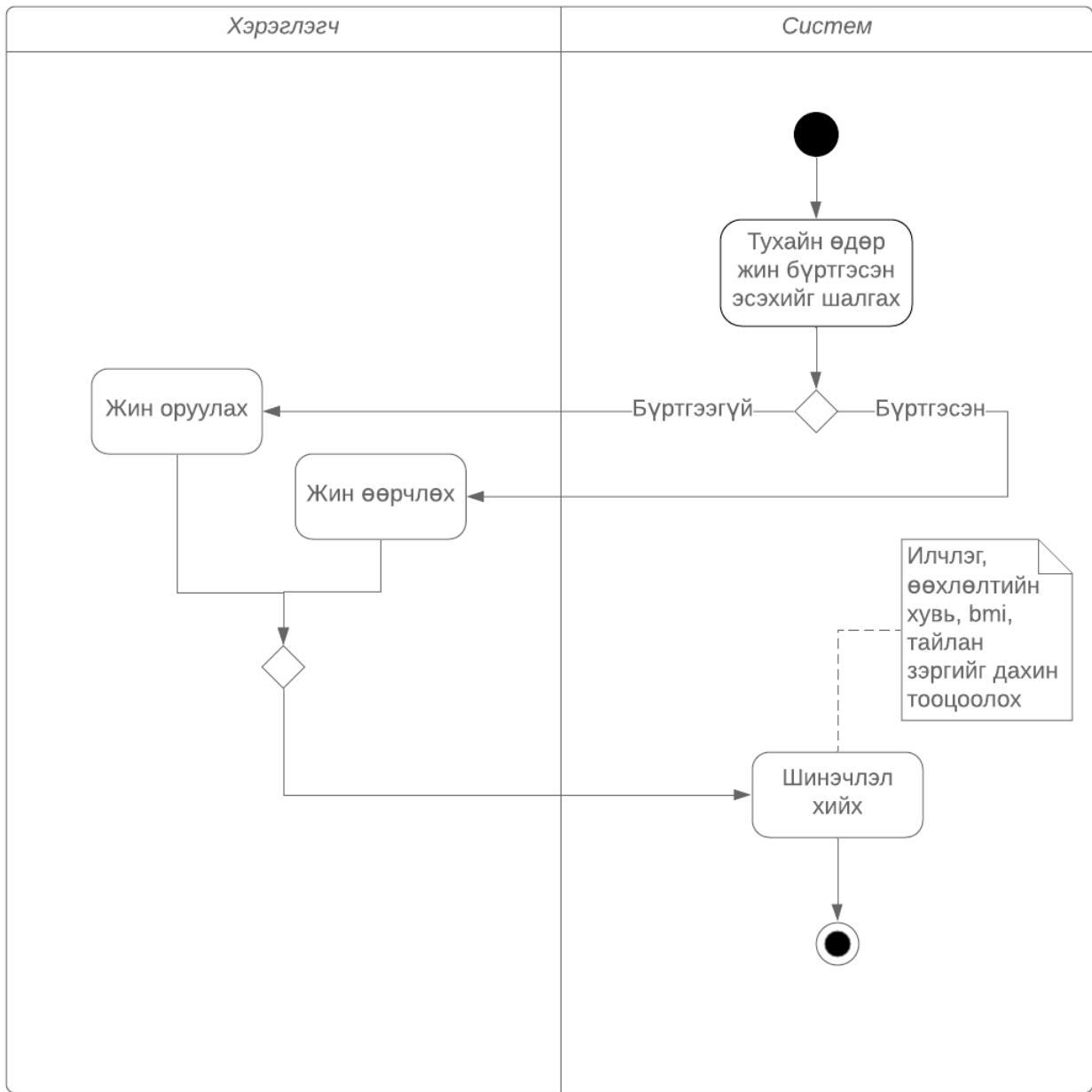


Зураг 2.3: Хэрэглэгч нэвтрэх болон бүртгүүлэх үйл ажиллагааны диаграм

Тайлбар

- Хэрэглэгч бүртгүүлэх болон нэвтрэх гэсэн сонголт хийнэ.
- Бүртгүүлэх үйл ажиллагаа нь мэдээллийг оруулах болон түүнийг шалгах үйлдлээс бүрдэх болно.
- Шалгаад алдаагүй бол мэдээллийг цааш боловсруулах бол алдаатай гэсэн үед дахин мэдээллийг оруулах үйлдэл рүү буцна
- Нэвтрэх үйл ажиллагаа нь мэдээллийг оруулах болон баталгаажуулах үйлдлээс бүрдэнэ.
- Буруу бол нүүр хуудас руу буцна.

2.4.2 Жин бүртгэх болон өөрчлөх



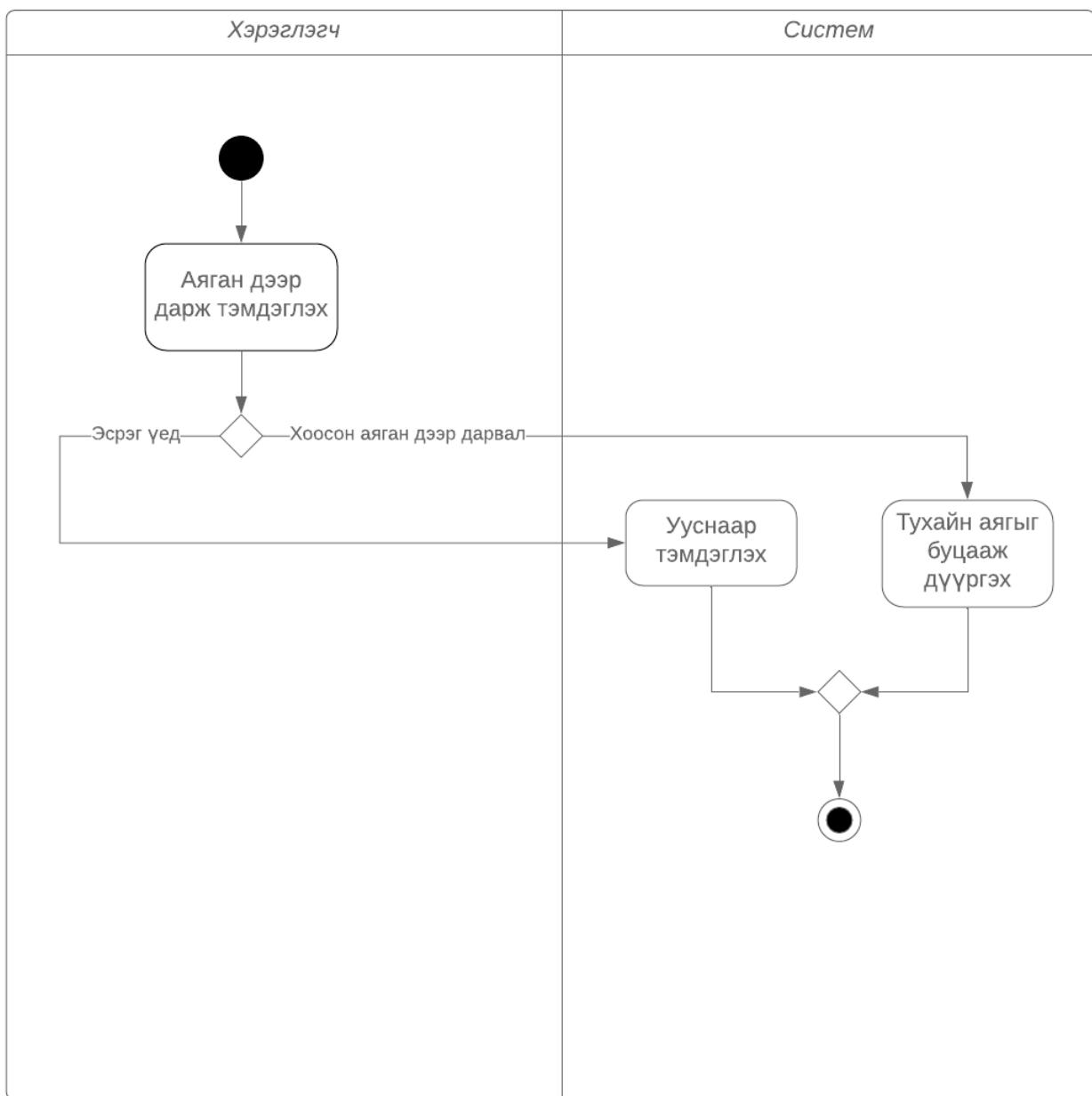
Зураг 2.4: Жин бүртгэх болон өөрчлөх үйл ажиллагааны диаграмм

Тайлбар

- Жин бүртгэх үед тухайн өдөр жин бүртгэгдсэн эсэхийг шалгана.

- Бүртгэсэн бол шинэчлэнэ.
- Бүртгээгүй бол шинээр бүртгэнэ.

2.4.3 Усны бүртгэл хийх

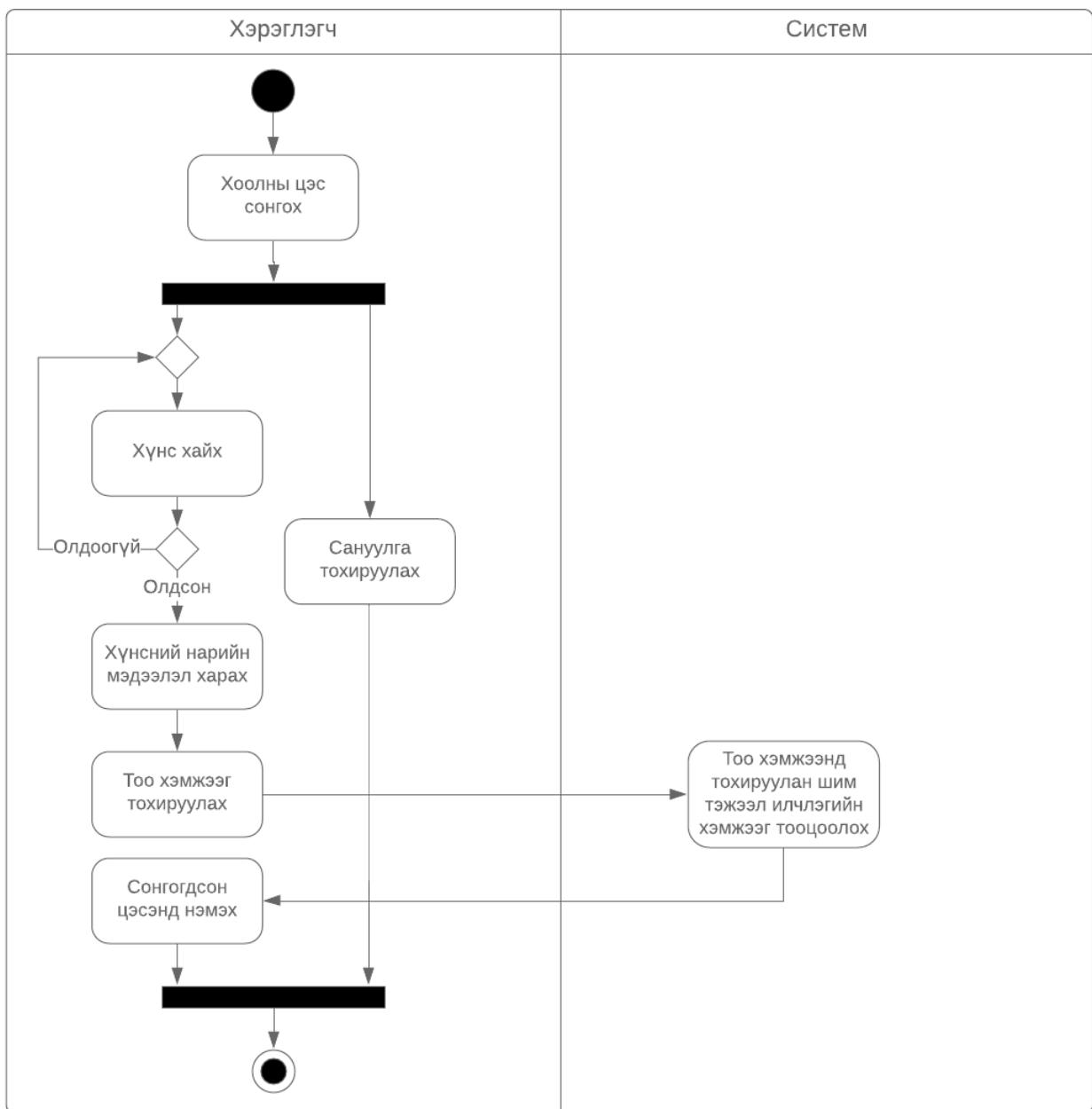


Зураг 2.5: Усны бүртгэл хийх үйл ажиллагааны диаграмм

Тайлбар

- Усны бүртгэл хийхдээ аяган дээр дарна.
- Хэрэв аяга хоосон бол аягыг уугаагүй гэж тэмдэглэнэ.
- Эсрэг үед ууснаар тэмдэглэнэ.

2.4.4 Хоолны төлөвлөгөө боловсруулах



Зураг 2.6: Хоолны төлөвлөгөө боловсруулах үйл ажиллагааны диаграмм

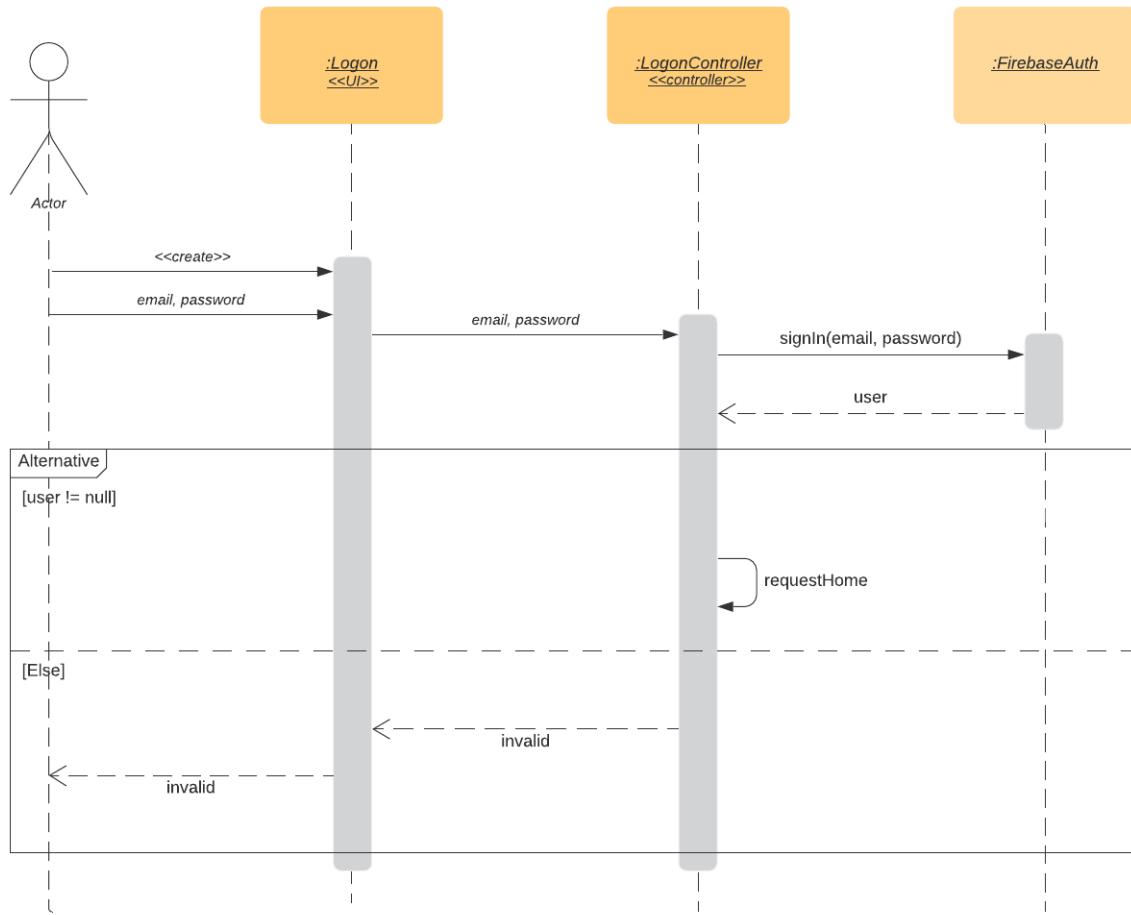
Тайлбар

- Төлөвлөгөө боловсруулах нь хүнс сонгох болон сануулга үүсгэх гэсэн хэсгүүдээс бүрдэнэ.

- Тухайн хүнс олдвол нарийн мэдээллийг харах мөн тоо хэмжээг тохируулаад сонгоно.

2.5 Дарааллын диаграм

2.5.1 Хэрэглэгч нэвтрэх



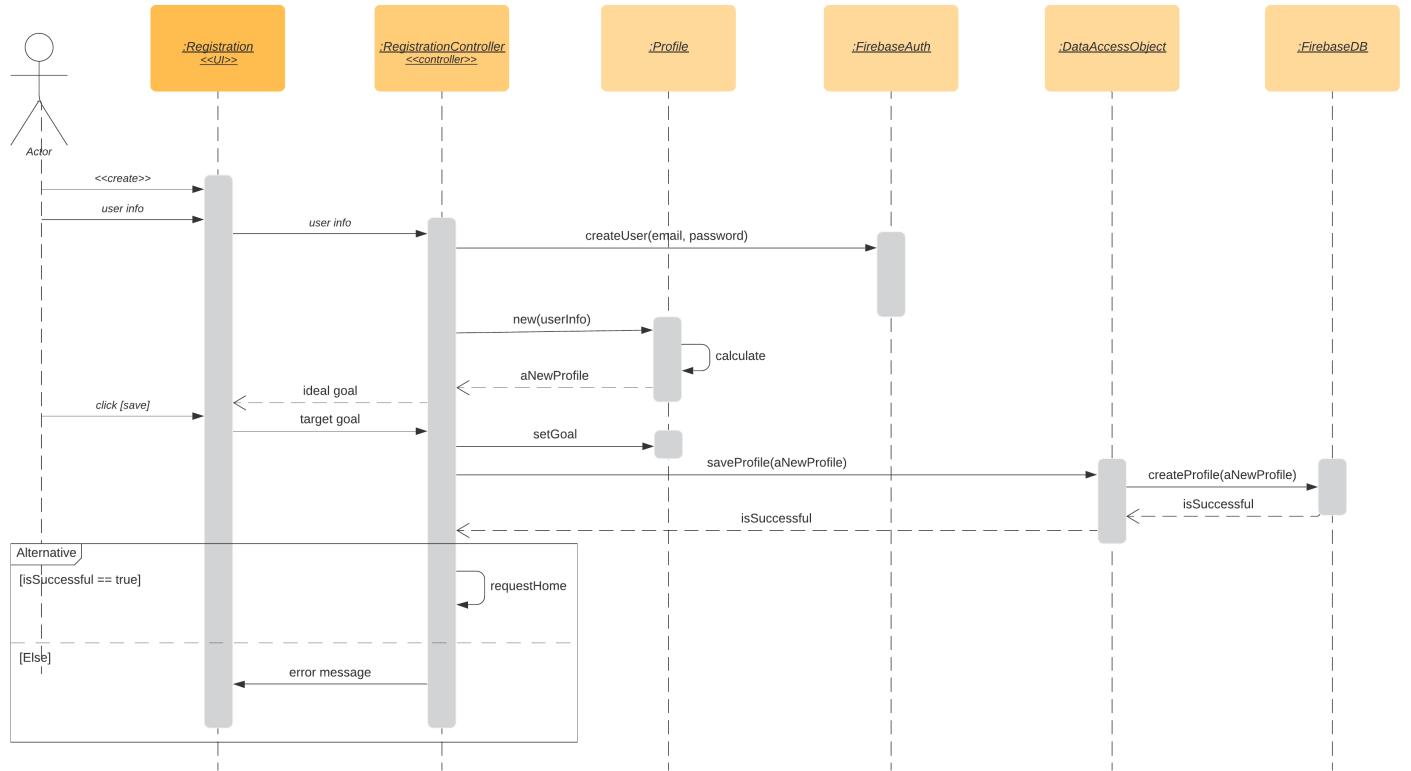
Зураг 2.7: Хэрэглэгч нэвтрэх дарааллын диаграм

Тайлбар

- Хэрэглэгч нэвтрэх хуудсанд цахим хаяг болон нууц үгийг оруулна.
- Нэвтрэх хуудсын контроллор класс нь Firebase authentication -ний тусламжтай зөв эсэхийг шалгана.

- Зөв бол нүүр хуудсыг дуудна.
- Эсрэг үед алдааны мэдээг нэвтрэх хуудсанд харуулна.

2.5.2 Хэрэглэгч бүртгүүлэх



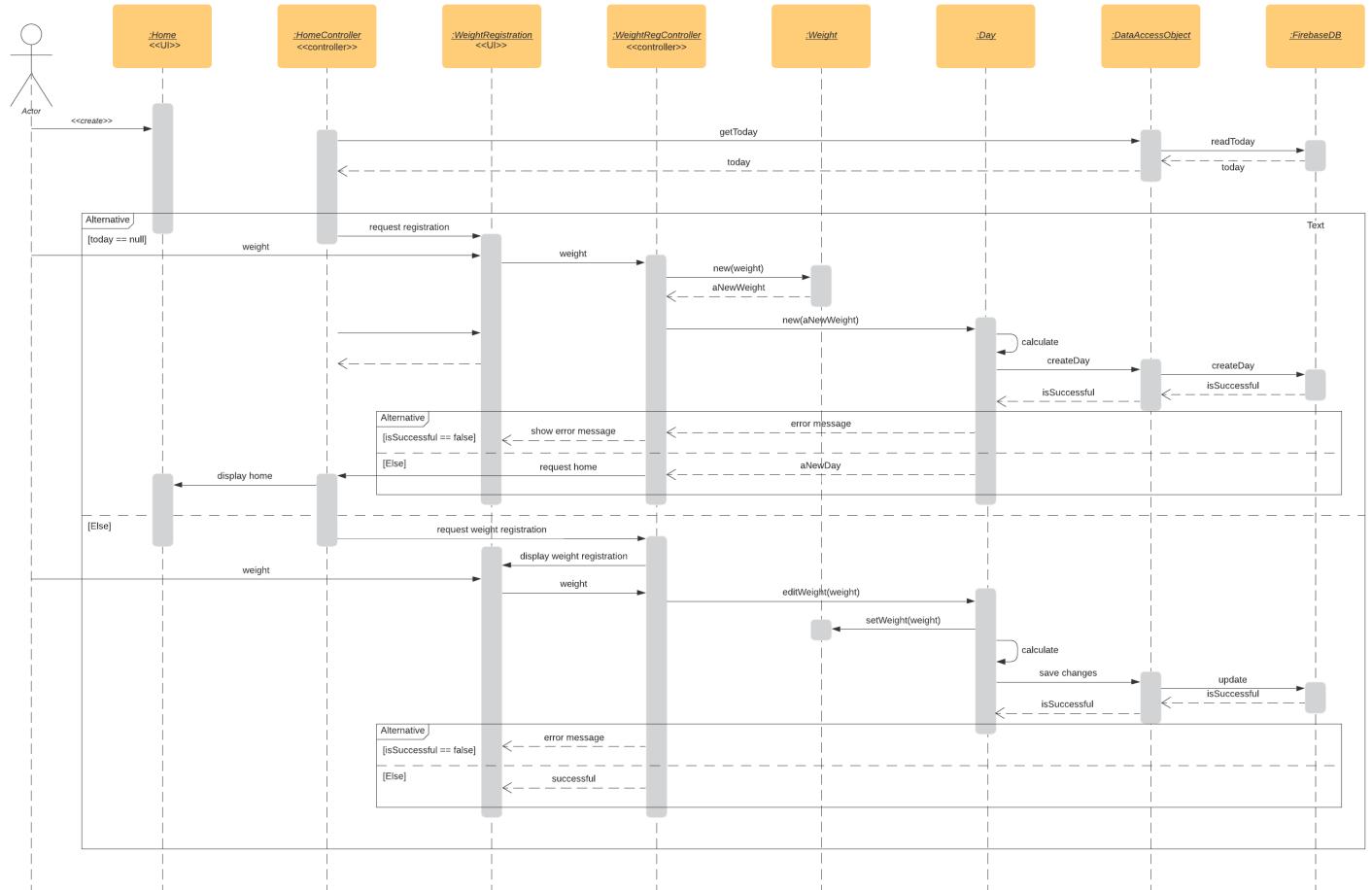
Зураг 2.8: Хэрэглэгч бүртгүүлэх дарааллын диаграмм

Тайлбар

- Хэрэглэгч бүртгүүлэх хуудсанд шаардлагатай мэдээллийг оруулна.
- Бүртгүүлэх хуудсын контроллор класс нь FirebaseAuth -ийн тусламжтай шинэ хэрэглэгч үүсгээд тохиромжтой зорилтуудыг буцаана.
- Бүртгүүлэх хуудсын контроллор класс нь хэрэглэгчийн сонгосон зорилтыг хадгална.
- Хэрэглэгчийн мэдээллийг хадгална.

- Хадгалах явц нь амжилттай болсон тохиолдолд нүүр хуудсыг дуудна.
- Амжилтгүй үед алдааны мэдээг харуулна.

2.5.3 Жин бүртгэх



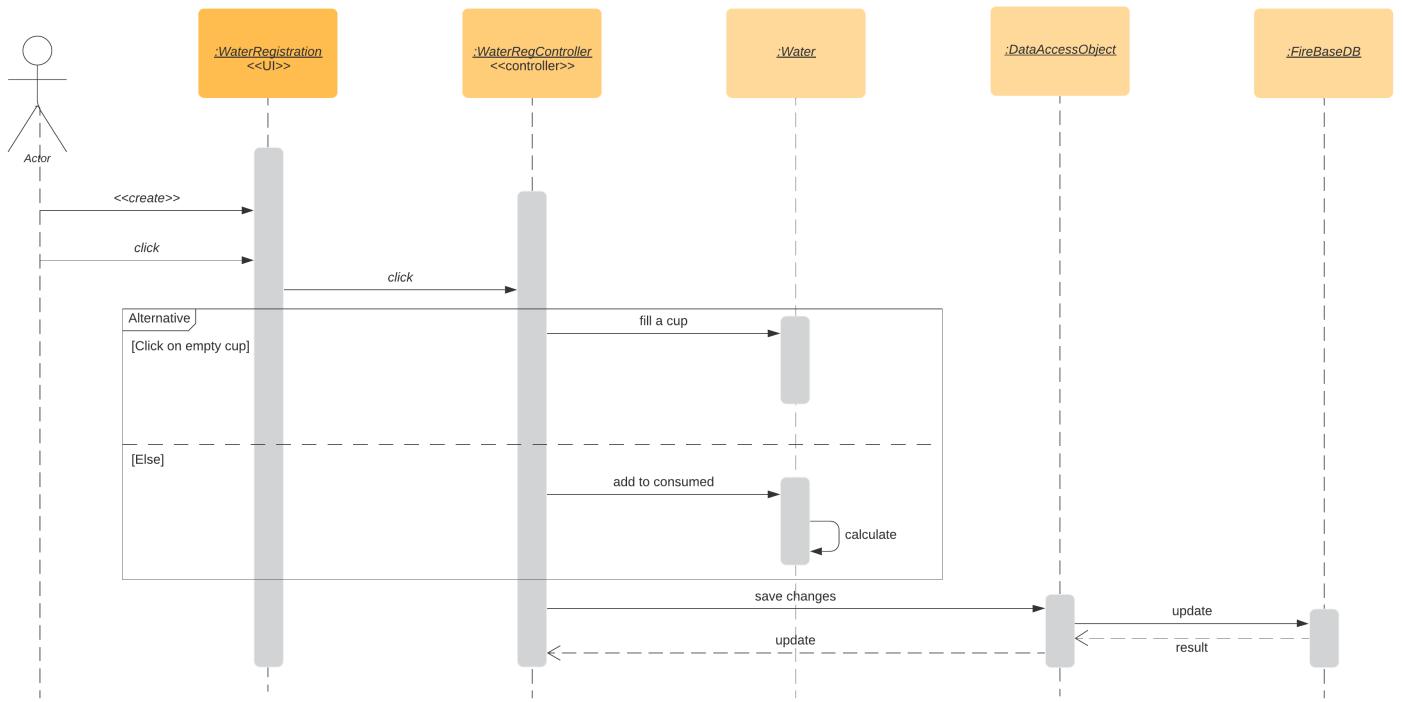
Зураг 2.9: Жин бүртгэх үйл дарааллын диаграм

Тайлбар

- Хэрэглэгч нүүр хуудсыг дуудна.

- Нүүр хуудасны контроллор класс нь өнөөдрийн түүхийг дуудна.
- Хэрэглэгч жин бүртгэх хуудсыг дуудна.
- Хэрэв өнөөдрийн түүх хоосон утга буцаавал шинэ өдөр болон жинг үүсгэнэ.
- Жин болон өдрийг хадгална.
- Хэрэв өнөөдрийн түүх байгаа тохиолдолд тухайн өдрийн жинг хэрэглэгчийн оруулсан жингээр шинэчилж хадгална.
- Хадгалах үйл явц амжилтгүй болсон тохиолдолд алдааны мэдээг буцаана.
- Амжилттай болсон тохиолдолд жин бүртгэх хуудасны контроллор класс нь нүүр хуудсыг дуудна.

2.5.4 Усны бүртгэл хийх

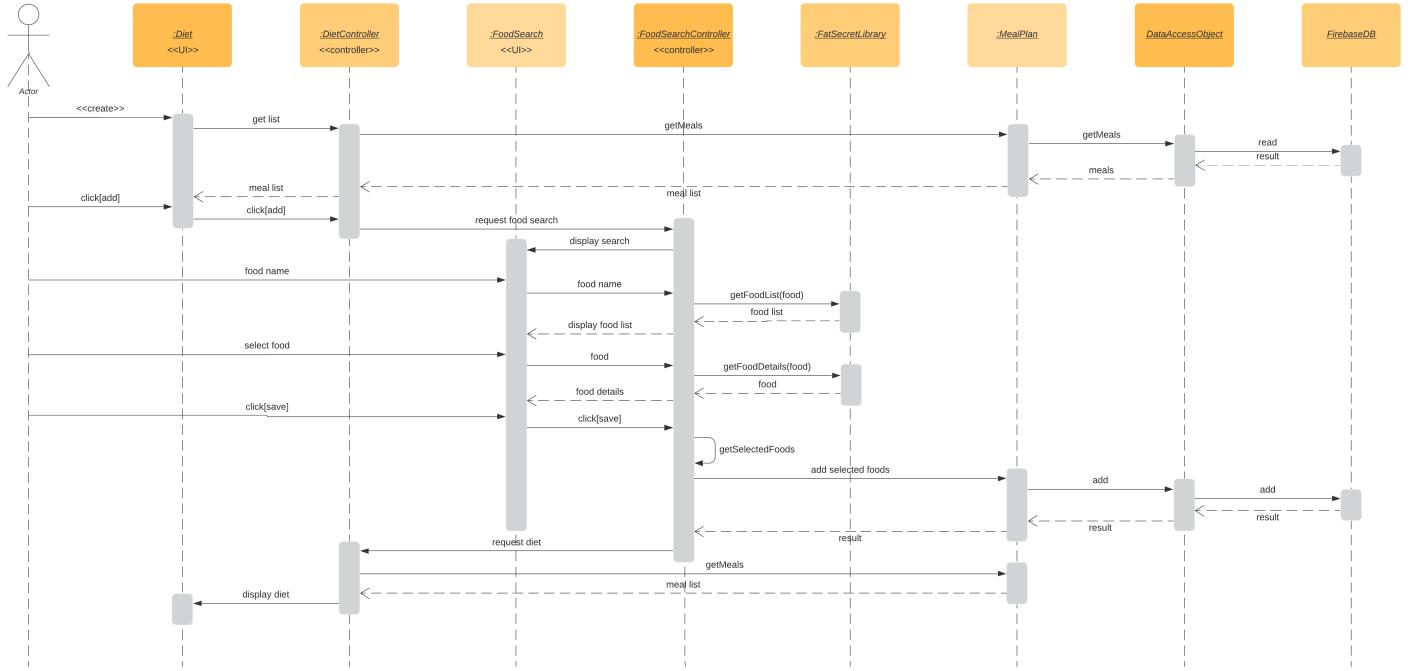


Зураг 2.10: Усны бүртгэл хийх дарааллын диаграмм

Тайлбар

- Хэрэглэгч аяга дээр дарна.
- Хэрэв аяга хоосон бол тухайн өдрийн уусан уснаас хасна.
- Хэрэв аяга дүүрэн бол тухайн өдрийн уусан усны хэмжээг нэмэгдүүлнэ.
- Өөрчлөлтийг хадгална.

2.5.5 Хоолны төлөвлөгөө боловсруулах



Зураг 2.11: Хоолны төлөвлөгөө боловсруулах дарааллын диаграмм

Тайлбар

- Хэрэглэгч хоолны төлөвлөгөөний хуудсыг дуудна.
- Хоолны төлөвлөгөөний контроллор класс нь хоолны жагсаалтыг дуудаж буцаана.
- Хэрэглэгч нэмэх товчийг дарах үед контроллор класс нь хүнс хайх хуудсыг харуулна.
- Хэрэглэгч хүнсний нэр оруулах үед хүнс хайх хуудасны контроллор класс нь ФатьСикрет сангаас тухайн хүнснийг хайна.
- Хайлтын үр дүнд ирсэн хүнсний жагсаалтыг буцаана.
- Хэрэглэгч хүнсний жагсаалтаас сонгоно.

- Контроллор класс ФатьСикрет сангаас тухайн хүнсний дэлгэрэнгүйг хайж буцаана.
- Хэрэглэгч нэмэх ивент үүсгэх үед тухайн хүнсийг хоолны төлөвлөгөөнд нэмж хадгална.

3. Хэрэгжүүлэлт

3.1 ФатъСикрет платформ (FatSecret Platform API)

Хэрэглэгч FatSecret Platform API ашиглаж хоол хүнсний талаарх нарийн мэдээллийг хайж хоолны төлөвлөгөө боловсруулах юм.

3.2 Фаярбэс гэрчлэл (Firebase Authentication)

Нэвтрэх болон бүртгүүлэхдээ Firebase Authentication-г ашигласан. Энэ нь нууцлал өндөртэй бөгөөд ашиглахад хялбар байдаг. Хэрэглэгч бүртгүүлэхдээ өөрийн и-мэйл, нэр, утасны дугаар болон нууц үгийг оруулан нэвтэрч болно. Нэр болон нууц үгээр бүртгүүлэх бол програм хэрэглэгчийн оруулсан нууц үг болон нэрийг FirebaseAuth объектын createUser() функцыг ашиглан Firebase сервер рүү явуулдаг. Мөн шинэ хэрэглэгчийг оруулсан утасны дугаараар нь баталгаажуулах(Phone Number Authentication) ажиллагааг нэмсэн. Firebase сервер нь орж ирсэн мэдээллүүдийг өөрийн нөхцөлүүдээр шалган бүртгэдэг. Бүртгэх функцын хариуг addOnSuccessListener сонсогчийн тусlamжтайгаар мэддэг. Тухайн хэрэглэгч бүртгэгдсэн тохиолдолд Firebase-н authentication хэсэгт бүртгэл нь нэмэгдэх ба тухайн хэрэглэгчийн цахим хаяг, бүртгүүлсэн огноо, хэрэглэгчийн ID-г хадгалсан байдаг. Бүртгэлтэй хэрэглэгчийн мэдээллийг админ засах боломжгүй ба зөвхөн тухайн хэрэглэгчийг идэвхгүй болгох, устгах, нууц үгийг сэргээх гэсэн функцуудыг ашиглаж болно.

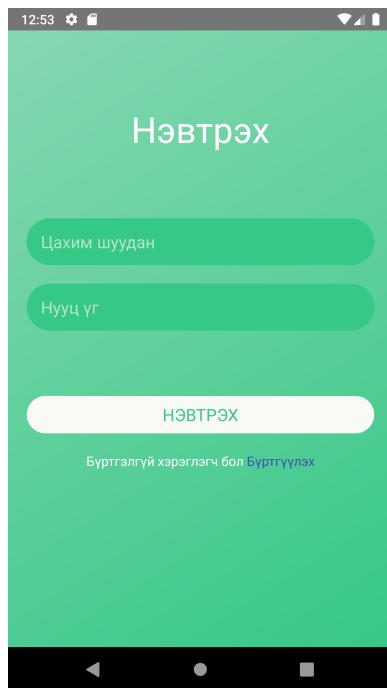
3.3 Фаярбэс өгөгдлийн сан (Firebase Realtime Database)

Firebase-ын Firestore өгөгдлийн сан нь өгөгдлийг JSON бүтцээр хадгалдаг. Энэ нь программыг өгөгдлийн сантай ажиллахад илүү хялбар болгодог. Энгийнээр тайлбарлавал

өгөгдлийг объект байдлаар хадгалдаг.

3.4 Хэрэглэгчийн интерфейс

3.4.1 Хэрэглэгч нэвтрэх

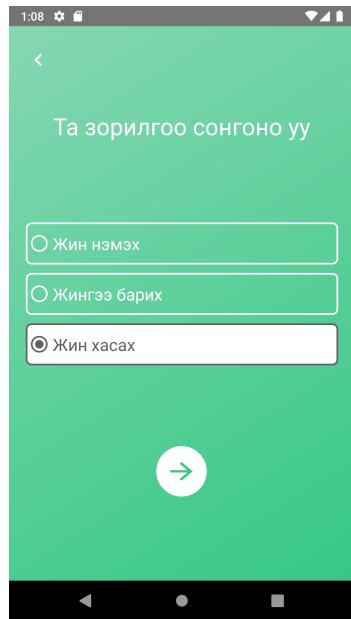


Зураг 3.1: Хэрэглэгч нэвтрэх хуудас

Тайлбар

- Хэрэглэгч өөрийн цахим хаяг болон нууц үгээр нэвтрэх хуудас.
- Бүртгэлгүй хэрэглэгч нь ”Бүртгүүлэх” дээр дарж бүртгэх үйл явцыг эхлүүлнэ.

3.4.2 Зорилгоо сонгох

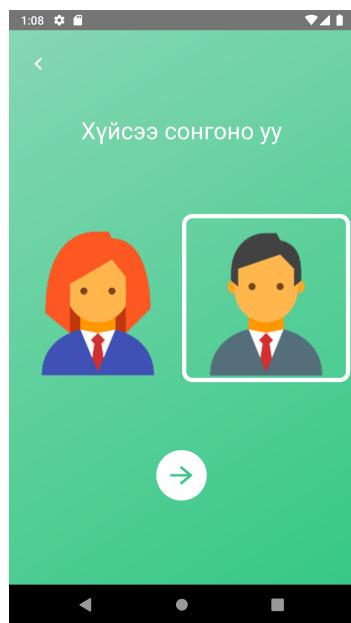


Зураг 3.2: Зорилгоо сонгох хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн зорилгыг бүртгэх хуудас.

3.4.3 Хүйс сонгох



Зураг 3.3: Хүйс сонгох хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн хүйсийг бүртгэх хуудас.

3.4.4 Хөдөлгөөний идэвхи сонгох

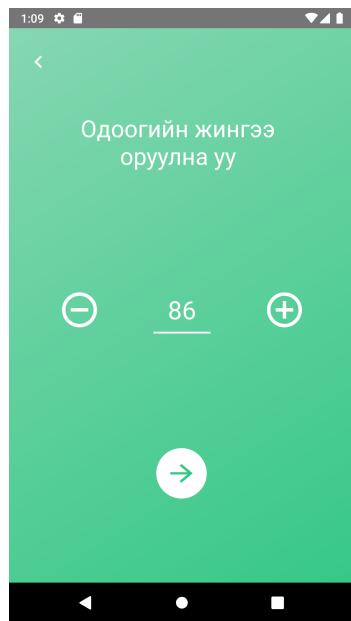


Зураг 3.4: Хөдөлгөөний идэвхи сонгох хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн хөдөлгөөний идэвхийг бүртгэх хуудас.
- Хэрэв дасгал хөдөлгөөн хийдэггүй суугаа ажил эрхэлдэг бол ”Маш бага” -ийг сонговол зохистой.
- Хэрэв 7 хоногт 1 -ээс 3 удаа бэлтгэл хийдэг бол ”Бага” -ийг сонговол зохистой.
- Хэрэв 7 хоногт 3 -аас 5 удаа бэлтгэл хийдэг бол ”Идэвхитэй” -ийг сонговол зохистой.
- Хэрэв 7 хоногт 6 -аас 7 удаа бэлтгэл хийдэг бол ”Маш идэвхитэй” -ийг сонговол зохистой.

3.4.5 Жингээ оруулах



Зураг 3.5: Жингээ оруулах хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн одоогийн жинг бүртгэх хуудас.
- Хэмжих нэгж нь килограмм байна.

3.4.6 Өндрөө оруулах



Зураг 3.6: Өндрөө оруулах хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн өндөр бүртгэх хуудас.
- Хэмжих нэгж нь сантиметр байна.

3.4.7 Төрсөн огноо сонгох

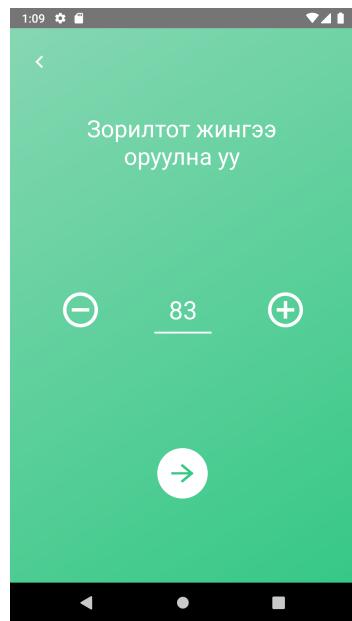


Зураг 3.7: Төрсөн огноо сонгох хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн төрсөн огноог бүртгэх хуудас.

3.4.8 Зорилтот жин сонгох

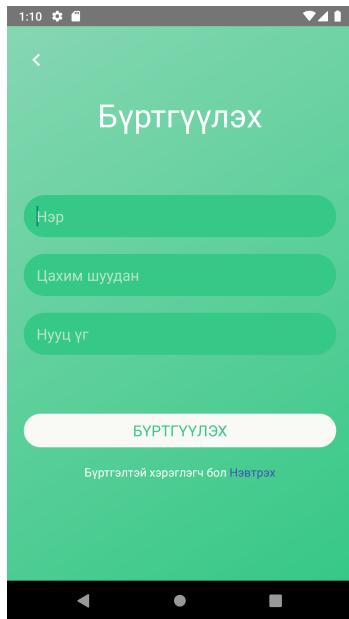


Зураг 3.8: Зорилтот жин сонгох хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчийн зорилтот жинг бүртгэх хуудас.
- Хэмжих нэгж нь килограмм байна.

3.4.9 Хэрэглэгч бүртгүүлэх



Зураг 3.9: Хэрэглэгч бүртгүүлэх хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгч нь нэр, цахим хаяг болон нууц үгээ оруулж бүртгүүлиэ.
- Хэрэв бүртгэлтэй хэрэглэгч бол ”Нэвтрэх” дээр дарж нэвтрэх хуудсыг дуудна.

3.4.10 Боломжит хугацаа болон илчлэгийг санал болгох

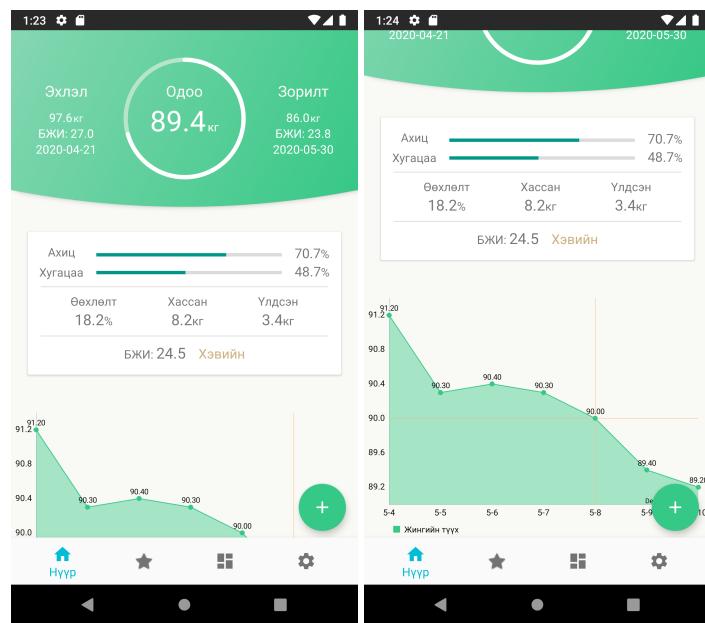


Зураг 3.10: Боломжит хугацаа болон илчлэгийг санал болгох хуудас

Тайлбар

- Шинээр бүртгүүлж буй хэрэглэгчээс авсан мэдээлэл дээр үндэслэж боломжит илчлэг болон хугацааг харуулах хуудас.
- Энэ хуудас нь бүртгэх үйл явцын эцсийн хуудас юм.
- Хэрэглэгч өөрийн хүссэн илчлэг болон хугацааг сонгосноор бүртгэх үйл явц дуусна.

3.4.11 Нүүр хуудас



Зураг 3.11: Нүүр хуудас

Тайлбар

- Хуудасны дээд талын хэсэгт хэрэглэгчийн эхлэл, одоо болон зорилтот биеийн жин болон биеийн жингийн индексийг харах боломжтой.
- Мөн жин болон хугацааг дугуй болон шугаман ахиц дэвшлийн диаграмаар харах боломжтой.
- Хуудасны дунд хэсгээс биеийн өөхлөлтийн хувь, хассан нэмсэн жингийн зөрүү болон үлдсэн жинг харах боломжтой.
- Үүний дараа биеийн жингийн индекс болон биеийн жингийн төлөвийг харах боломжтой.
- Хуудасны доод хэсэгт хэрэглэгчийн сүүлийн 7 хоногийн жинг шугаман диаграм байдлаар харах боломжтой.

3.4.12 Усны бүртгэл

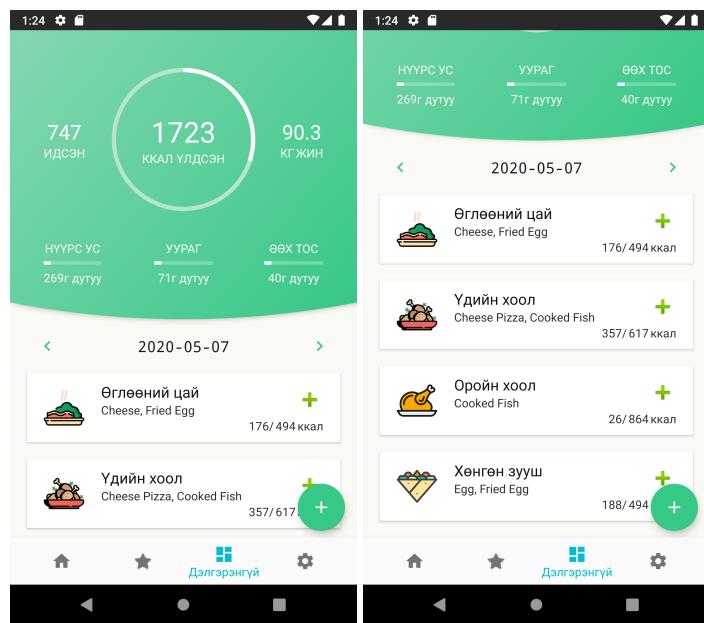


Зураг 3.12: Усны бүртгэл хуудас

Тайлбар

- Хэрэглэгчийн уух усны хэмжээг аягаар илэрхийлж харуулсан хуудас.
- Дүүрэн аяга дээр дарвал ууснаар тэмдэглэгдэж хоосон аяганы зургаар солигдоно.
- Хоосон аяга дээр дарвал уугаагүйгээр тэмдэглэгдэж дүүрэн аяганы зургаар солигдоно.
- Хуудасны доод хэсэгт уувал зохих аяганы тоо болон түүнээс уусан аяганы тоог харуулсан болно.

3.4.13 Хоолны төлөвлөгөө болон түүх



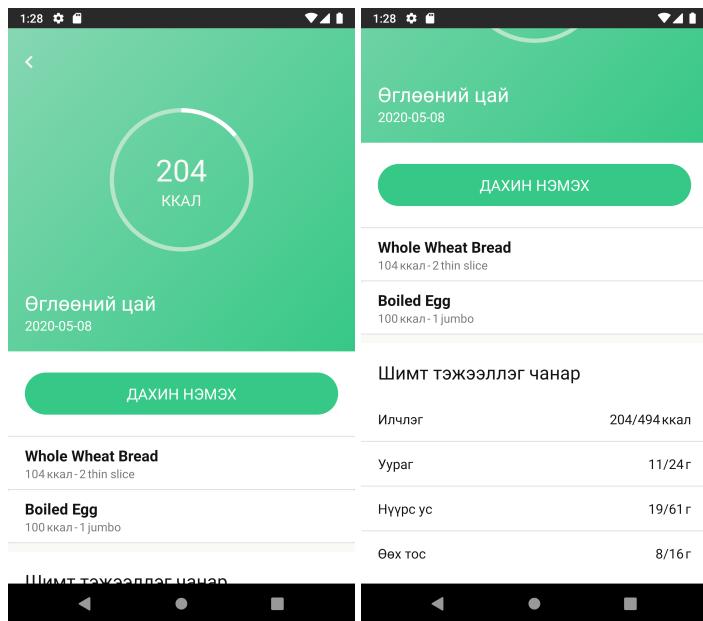
Зураг 3.13: Хоолны төлөвлөгөө болон түүх хуудас

Тайлбар

- Хуудасны дээд хэсэгт тухайн өдөрт идсэн болон үлдсэн илчлэгийн хэмжээн диаграм байдлаар харах боломжтой.
- Мөн тухайн өдрийн жинг харах боломжтой.
- Түүний дараа тухайн өдөрт хүнснээс авсан шим тэжээлт чанар болох нүүрс ус, өөх тос болон уургийн хэмжээг төлөвийн хамт диаграм байдлаар харах боломжтой.
- Огнооны хоёр талын сумны тусламжтайгаар огноог өөрчилж тухайн огноон дээрх түүхийг харах боломжтой.
- Хоолны цэс тус бүрээс тухайн цэс дээр нэмэгдсэн хүнсний нэрс болон авбал зохих илчлэг болон түүнээс авсан илчлэгийн хэмжээг харах боломжтой.

- Хэрэв тухайн өдрийн авсан илчлэг болон шим тэжээлт чанар нь авбал зохих хэмжээнээс хэтэрсэн тохиолдолд диаграм нь шар болон улаан өнгөтэй болж өөрчлөгднө.

3.4.14 Хоолны дэлгэрэнгүй

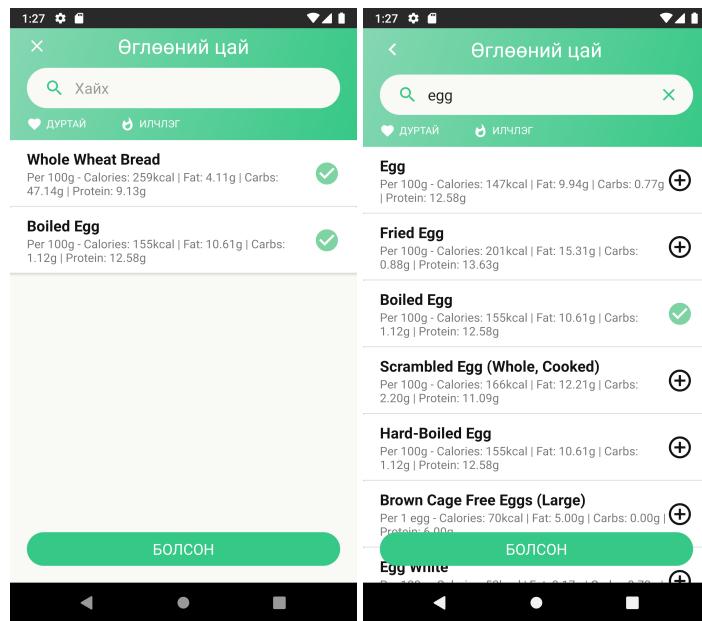


Зураг 3.14: Хоолны дэлгэрэнгүй хуудас

Тайлбар

- Хоолны цэс дээр дарснаар энэ хуудсыг нээх боломжтой.
- Энэ хуудсанд тухайн цэс дэх хүнсний жагсаалт мөн тэдгээрийн нийт шим тэжээлт чанарын хэмжээ болон авбал зохих хэмжээг харах боломжтой.
- Мөн тухайн цэсний нийт илчлэгийн хэмжээг диаграм байдлаар харах боломжтой.
- ”Дахин нэмэх” дээр дарснаар хүнс хайх хуудас руу шилжих болно.
- Хүнс дээр дарснаар тухайн хүнсний нарийн мэдээллийг харуулах хуудас руу шилжих болно.
-

3.4.15 Хүнс хайх

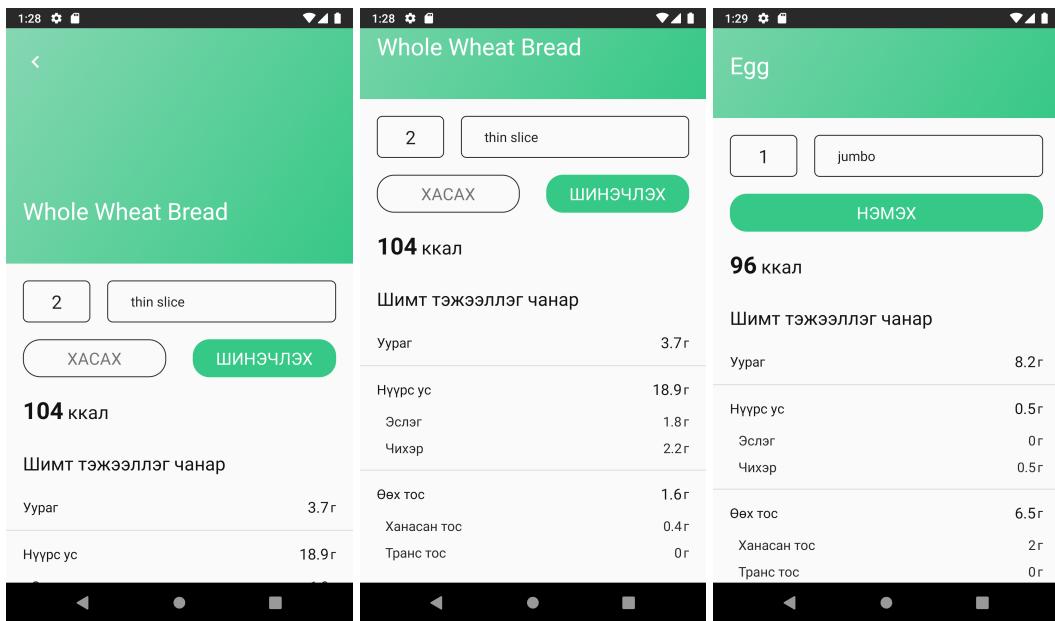


Зураг 3.15: Хүнс хайх хуудас

Тайлбар

- Хайх хуудас нь аль цэсээс шилжиж ирснээс хамаарч тухайн цэс дэх хүнсний жагсаалтыг сонгосон гэсэн төлөвтэй харуулах болно.
- Хүнсний нэр бичих үед тухайн нэрээр хүнс хайж үр дүнг харуулах болно.
- Тухайн үр дүнгийн жагсаалтаас харгалзах нэмэх тэмдгийг дарснаар тухайн хүнсийг шууд сонгох боломжтой.
- Харин тухайн хүнс дээр дарвал тухайн хүнсний хэмжээг сонгох болон нарийн мэдээллийг харах боломжтой хуудас руу шилжих болно.
- ”Болсон” эсвэл ”х” тэмдэг дээр дараах үед өөрчлөлтийг хадгалж буцах болно.

3.4.16 Хүнсний дэлгэрэнгүй

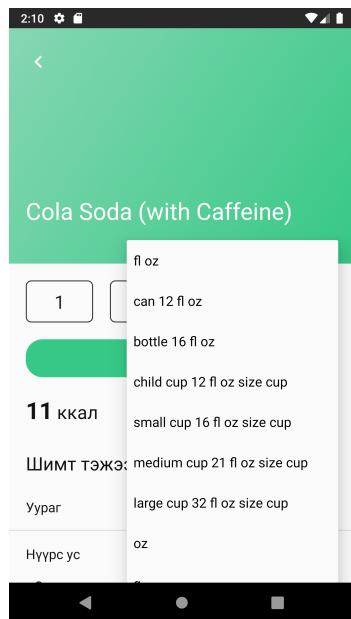


Зураг 3.16: Хүнсний дэлгэрэнгүй хуудас

Тайлбар

- Хүнсний дэлгэрэнгүй хуудсыг хаанаас дуудаснаас хамаарч нэмэх эсвэл хасах болон шинэчлэх төлөвтэй харагдана.
- Хэрэв хайх хуудаснаас дуудах үед тухайн хүнс нь тухайн цэсэнд байхгүй бол нэмэх гэсэн төлөвтэй харагдана.
- Тухайн хүнс нь тухайн цэсэнд байвал хасах болон шинэчлэх төлөвтэй харагдана.
- Энэ хуудас нь тухайн хүнсний илчлэг болон нарийн мэдээллийг сонгосон хэмжээгээр бодож харуулах үүрэгтэй.

3.4.17 Хүнсний хэмжээг тохируулах

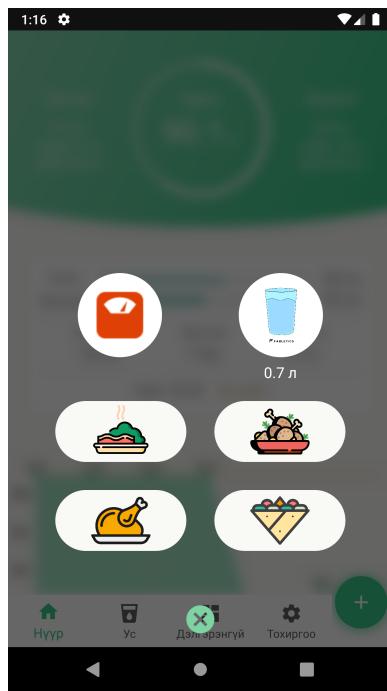


Зураг 3.17: Хүнсний хэмжээг тохируулах хуудас

Тайлбар

- Тухайн хүнсний хэмжээ болон хэмжүүрийг сонгох боломжтой.

3.4.18 Хялбар цэс

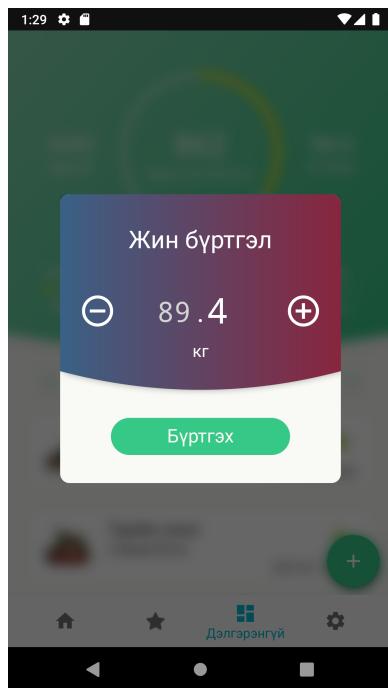


Зураг 3.18: Хялбар цэс цонх

Тайлбар

- Тухайн цонхны тусlamжтай жин бүртгэх цонхруу шилжих боломжтой.
- Мөн аяганы зураг дээр дарснаар усны бүртгэл хийх боломжтой ба тухайн өдрийн нийт уусан усны хэмжээг харах боломжтой.
- Цэс тус бүрийн зураг дээр дарснаар тухайн цэсэнд хүнс нэмэх боломжтой.

3.4.19 Жин бүртгэх

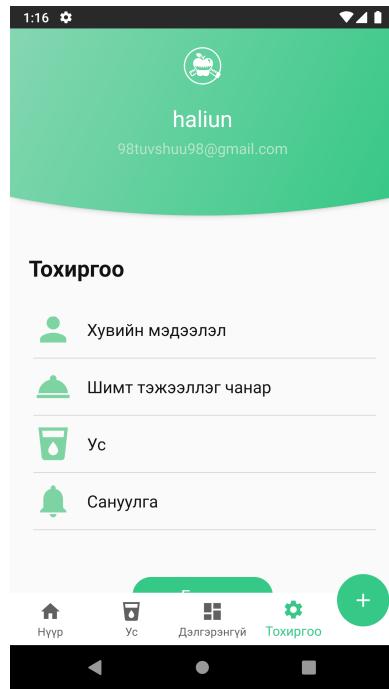


Зураг 3.19: Жин бүртгэх хуудас

Тайлбар

- Жингээ тохируулж тухайн өдрийн жинг шинэчлэх боломжтой.

3.4.20 Тохиргоо

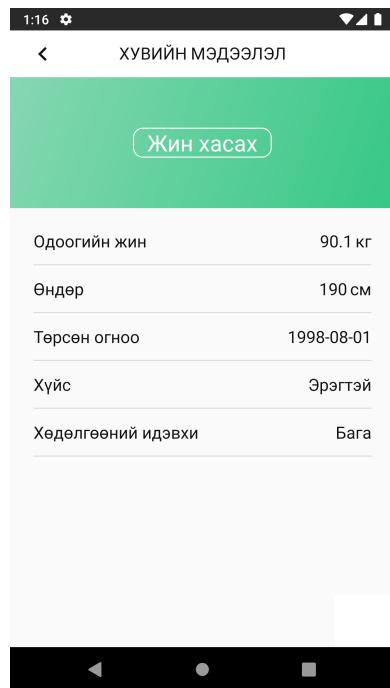


Зураг 3.20: Тохиргоо хуудас

Тайлбар

- Тухайн хуудасны дээд хэсэгт хэрэглэгчийн нэр болон цахим хаяг харагдана.
- Тохиргоо хуудас нь хувийн мэдээлэл, шимт тэжээллэг чанар, ус болон сануулгын тохиргоог агуулна.
- Хуудасны доод хэсэг дэх ”Гарах” товчийг дарснаар системээс гарах болно.

3.4.21 Хувийн мэдээлэл

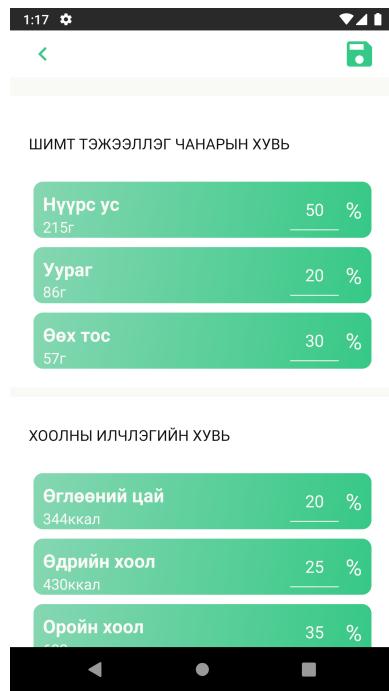


Зураг 3.21: Хувийн мэдээллийн хуудас

Тайлбар

- Тухайн хуудасны бүх хувийн мэдээллийг засах боломжтой юм.
- Мөн өөрчлөлт орох болгонд илчлэгийн хэмжээ өөрчлөгдсөнийг анхааруулах болно.
- Хэрэглэгч өөрийн зорилгыг өөрчлөх боломжтой юм.

3.4.22 Шимт тэжээллэг чанар

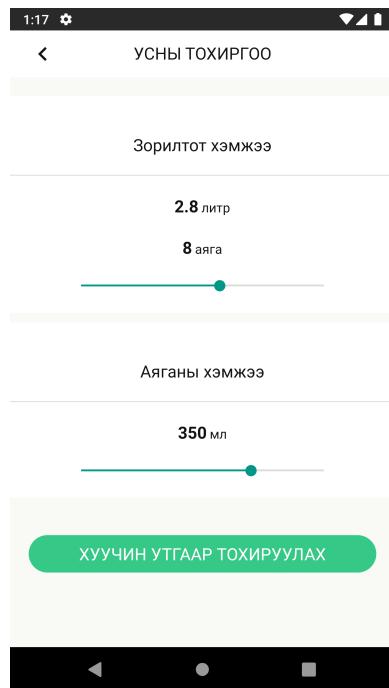


Зураг 3.22: Шимт тэжээллэг чанарын хуудас

Тайлбар

- Хуудасны эхний хэсэгт хэрэглэгч хоол хүнснээс авах шимт тэжээллэг чанарын хувийг тохируулах боломжтой.
- Энэ нь анх үүсэхдээ тогтмолоор нүүрс ус, уураг, өөх тос тус бүр харгалзан 50, 20, 30 хувьтай байна.
- Хуудасны дараагийн хэсэгт хэрэглэгч өдрийн илчлэгийн хэмжээг цэс тус бүр нь тохируулах боломжтой.
- Энэ нь анх үүсэхдээ тогтмолоор өглөөний цай, үдийн хоол, оройн хоол, хөнгөн зууш тус бүр харгалзан 20, 25, 35, 20 хувьтай байна.

3.4.23 Усны тохиргоо

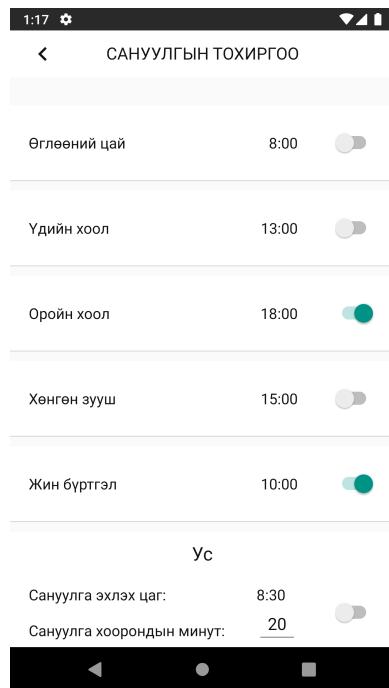


Зураг 3.23: Усны тохиргоо хуудас

Тайлбар

- Шугаман заалтын тусламжтай өдөрт уух усны хэмжээ болон аяганы хэмжээг тохируулах боломжтой.
- ”Хуучин утгаар тохируулах” товч дээр дарснаар хэрэглэгчийн уувал зохих усны хэмжээг бодож мөн аяганы хэмжээг 250 мл дээр тохируулна.
- Өдөрт уух усны дээд хэмжээ 5 литр байна.
- Аяганы дээд хэмжээ 500 мл байна.

3.4.24 Сануулгын тохиргоо



Зураг 3.24: Сануулгын тохиргоо хуудас

Тайлбар

- Цэс болон жин бүртгэх тус бүр цаг тохируулан идэвхжүүлснээр сануулга үүсгэх юм.
- Үс уух сануулга үүсгэхдээ сануулга эхлэх цаг болон сануулга хоорондын минутыг тохируулна.

ДҮГНЭЛТ

Энэхүү судалгааны ажлын хүрээнд хоол хүнсний илчлэг болон шимт тэжээллэг чанар мөн хүний өдрийн шингэн, илчлэг зэрэг хэрэгцээт зүйлсийн талаар тэдгээрийг бodoх аргуудыг судалж мэдсэн. Мөн төлөвлөгөөний дагуу ижил төстэй програмын судалгаа, онолын судалгаа, технологийн судалгаа, шинжилгээ зохиомж, хэрэгжүүлэлт гэсэн зорилгуудаа биелүүлсэн. Хэдий жижиг хэмжээний програм хөгжүүлсэн ч хэд хэдэн асуудлуутай тулгарсан. Тэдгээр асуудлыг хэрхэн шийдэх талаар судалж мэдэх нь илүү их зүйл сурх давуу тал болсон. Үүний үр дүнд өөрийн зорилтот програмыг тодорхой хэмжээнд хөгжүүлж дуусгасан. Мэдээж хэрэг 100% хөгжүүлж дуусаагүй, цааш хөгжүүлэх, сайжруулах зүйлс их байгаа билээ. Мөн жижиг гэхгүй ямар нэгэн систем, програмыг хөгжүүлэх үйл явцаас мэдэж авч туршлага хуримтлуулсан.

Bibliography

- [1] <https://firebase.google.com/docs>
- [2] <https://developer.android.com/docs>
- [3] <http://www.fatsecret4j.com/android>
- [4] <https://platform.fatsecret.com/api/Default.aspx?screen=rapih>
- [5] <https://www.tutorialspoint.com/android/index.htm>

A. ҮЧИЛСЭН ТӨЛӨВЛӨГӨӨ

№	Хийх ажил	Долоо хоног		1	2	3	4 Явц I	5	6	7	8 Явц II	9	10	11	12 Урьдчилсан хамгаалалт	13	Тайлбар
		Хийх ажил	Долоо хоног														
1	Судалгаа	Ижил тэст програм															
		Технологи															
2	Систем зохиомж	Шинжилгээ															
		Зохиомж															
3	Хэрэгжүүлэлт	Үнсэн код															
4	Шалгах, сайжруулах																
5	Тайлан	Явц															
		Эцсийн															

Зураг A.1: ҮЧИЛСЭН ТӨЛӨВЛӨГӨӨ

B. Кодын хэрэгжүүлэлт

```
1 mAuth.createUserWithEmailAndPassword(email, password)
2     .addOnCompleteListener(this, new OnCompleteListener<
3         AuthResult>() {
4             @Override
5             public void onComplete(@NonNull Task<AuthResult> task)
6             {
7                 if (task.isSuccessful()) {
8                     // Sign in success, update UI with the signed-
9                     // in user's information
10                    Log.d(TAG, "createUserWithEmail:success");
11                    FirebaseUser user = mAuth.getCurrentUser();
12                    updateUI(user);
13                } else {
14                    // If sign in fails, display a message to the
15                    // user.
16                    Log.w(TAG, "createUserWithEmail:failure", task.
17                        getException());
18                    Toast.makeText(EmailPasswordActivity.this, "
19                         Authentication failed.",
20                         Toast.LENGTH_SHORT).show();
21                    updateUI(null);
22                }
23            }
24        // ...
25    });
26}
```

Код B.1: Цахим шуудангаар нь шинэ хэрэглэгч үүсгэж буй код

```
1 mAuth.signInWithEmailAndPassword(email, password)
2     .addOnCompleteListener(this, new OnCompleteListener<
3         AuthResult>() {
4             @Override
5             public void onComplete(@NonNull Task<AuthResult> task)
6             {
7                 if (task.isSuccessful()) {
8                     // Sign in success, update UI with the signed-
9                     // in user's information
10                    Log.d(TAG, "signInWithEmail:success");
11                    FirebaseUser user = mAuth.getCurrentUser();
12                    updateUI(user);
13                } else {
14                    // If sign in fails, display a message to the
15                    // user.
16                    Log.w(TAG, "signInWithEmail:failure", task.
17                        getException());
18                    Toast.makeText(EmailPasswordActivity.this, "
19                         Authentication failed.",
20                         Toast.LENGTH_SHORT).show();
21                }
22            }
23        }
24    });
25}
```

```

15         updateUI(null);
16     }
17
18     // ...
19 }
20 );

```

Код В.2: Хэрэглэгч нэвтэрч буй код

```

1 GoogleSignInOptions gso = new GoogleSignInOptions.Builder(
2     GoogleSignInOptions.DEFAULT_SIGN_IN)
3     .requestIdToken(getString(R.string.default_web_client_id))
4     .requestEmail()
5     .build();
6
7 private void signIn() {
8     Intent signInIntent = mGoogleSignInClient.getSignInIntent();
9     startActivityForResult(signInIntent, RC_SIGN_IN);
10 }
11
12 @Override
13 public void onActivityResult(int requestCode, int resultCode,
14     Intent data) {
15     super.onActivityResult(requestCode, resultCode, data);
16
17     // Result returned from launching the Intent from
18     // GoogleSignInApi.getSignInIntent(...);
19     if (requestCode == RC_SIGN_IN) {
20         Task<GoogleSignInAccount> task = GoogleSignIn.
21             getSignedInAccountFromIntent(data);
22         try {
23             // Google Sign In was successful, authenticate with
24             // Firebase
25             GoogleSignInAccount account = task.getResult(
26                 ApiException.class);
27             firebaseAuthWithGoogle(account);
28         } catch (ApiException e) {
29             // Google Sign In failed, update UI appropriately
30             Log.w(TAG, "Google sign in failed", e);
31             // ...
32         }
33     }
34 }
35
36 private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
37     Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());
38
39     AuthCredential credential = GoogleAuthProvider.getCredential(
40         acct.getIdToken(), null);
41     mAuth.signInWithCredential(credential)
42         .addOnCompleteListener(this, new OnCompleteListener<
43             AuthResult>() {

```

```

36     @Override
37     public void onComplete(@NonNull Task<AuthResult>
38         task) {
39         if (task.isSuccessful()) {
40             // Sign in success, update UI with the
41             // signed-in user's information
42             Log.d(TAG, "signInWithCredential:success");
43             FirebaseUser user = mAuth.getCurrentUser();
44             updateUI(user);
45         } else {
46             // If sign in fails, display a message to
47             // the user.
48             Log.w(TAG, "signInWithCredential:failure",
49                   task.getException());
50             Snackbar.make(findViewById(R.id.main_layout),
51                           "Authentication Failed.", Snackbar.
52                           LENGTH_SHORT).show();
53             updateUI(null);
54         }
55     }
56 }

```

Код В.3: Хэрэглэгч Google хаягаа ашиглан нэвтэрч буй код

```

1 mCallbackManager = CallbackManager.Factory.create();
2 LoginButton loginButton = findViewById(R.id.buttonFacebookLogin);
3 loginButton.setReadPermissions("email", "public_profile");
4 loginButton.registerCallback(mCallbackManager, new FacebookCallback
5     <LoginResult>() {
6         @Override
7         public void onSuccess(LoginResult loginResult) {
8             Log.d(TAG, "facebook:onSuccess:" + loginResult);
9             handleFacebookAccessToken(loginResult.getAccessToken());
10        }
11
12        @Override
13        public void onCancel() {
14            Log.d(TAG, "facebook:onCancel");
15            // ...
16        }
17
18        @Override
19        public void onError(FacebookException error) {
20            Log.d(TAG, "facebook:onError", error);
21            // ...
22        }
23    });
24

```

```

25 protected void onActivityResult(int requestCode, int resultCode,
26     Intent data) {
27     super.onActivityResult(requestCode, resultCode, data);
28
29     // Pass the activity result back to the Facebook SDK
30     mCallbackManager.onActivityResult(requestCode, resultCode, data
31     );
32 }
33
34 private void handleFacebookAccessToken(AccessToken token) {
35     Log.d(TAG, "handleFacebookAccessToken:" + token);
36
37     AuthCredential credential = FacebookAuthProvider.getCredential(
38         token.getToken());
39     mAuth.signInWithCredential(credential)
40         .addOnCompleteListener(this, new OnCompleteListener<
41             AuthResult>() {
42             @Override
43             public void onComplete(@NonNull Task<AuthResult>
44                 task) {
45                 if (task.isSuccessful()) {
46                     // Sign in success, update UI with the
47                     // signed-in user's information
48                     Log.d(TAG, "signInWithCredential:success");
49                     FirebaseUser user = mAuth.getCurrentUser();
50                     updateUI(user);
51                 } else {
52                     // If sign in fails, display a message to
53                     // the user.
54                     Log.w(TAG, "signInWithCredential:failure",
55                         task.getException());
56                     Toast.makeText(FacebookLoginActivity.this,
57                         "Authentication failed.",
58                         Toast.LENGTH_SHORT).show();
59                     updateUI(null);
60                 }
61
62                 // ...
63             }
64         });
65 }

```

Код В.4: Хэрэглэгч Facebook хаягаа ашиглан нэвтэрч буй код

```

1 public double calcIdealRdi() {
2     if (gender.equals("    ")) {
3         return (10 * currentWeight + 6.25 * height - 5 * (
4             startDate.get(Calendar.YEAR) - dateOfBirth.get(
5                 Calendar.YEAR)) + 5) * calcActivityLevel();
6     } else {
7         return (10 * currentWeight + 6.25 * height - 5 * (
8             startDate.get(Calendar.YEAR) - dateOfBirth.get(
9

```

```

        Calendar.YEAR)) - 161) * calcActivityLevel();
6
7    }
8
9    public double calcActivityLevel() {
10       switch (activityLevel) {
11           case " ":
12               return 1.2;
13           case " ":
14               return 1.375;
15           case " ":
16               return 1.55;
17           default:
18               return 1.725;
19       }
20   }

```

Код В.5: Хэрэглэгчийн өдөрт авах илчлэгийн хэмжээг тооцож буй код

```

1  public LinkedHashMap<Double, Calendar> calcIdealTargetDate() {
2      LinkedHashMap<Double, Calendar> result = new LinkedHashMap
3          <>();
4      double rdi = calcIdealRdi();
5      if (goal.equals(" ")) {
6          double diff = targetWeight - currentWeight;
7          int days = (int) ((diff / 0.25) * 7);
8          Calendar targetDate1 = Calendar.getInstance();
9          targetDate1.add(Calendar.DAY_OF_MONTH, days);
10         result.put(rdi + 250, targetDate1);
11
12         days = (int) ((diff / 0.5) * 7);
13         Calendar targetDate2 = Calendar.getInstance();
14         targetDate2.add(Calendar.DAY_OF_MONTH, days);
15         result.put(rdi + 500, targetDate2);
16
17         days = (int) (diff * 7);
18         Calendar targetDate3 = Calendar.getInstance();
19         targetDate3.add(Calendar.DAY_OF_MONTH, days);
20         result.put(rdi + 1000, targetDate3);
21     } else if (goal.equals(" ")) {
22         double diff = currentWeight - targetWeight;
23         int minRdi;
24         if (gender.equals(" ")) {
25             minRdi = 1500;
26         } else {
27             minRdi = 1200;
28         }
29         if (rdi - 250 > minRdi) {
30             int days = (int) ((diff / 0.25) * 7);
31             Calendar targetDate1 = Calendar.getInstance();
32             targetDate1.add(Calendar.DAY_OF_MONTH, days);
            result.put(rdi - 250, targetDate1);
}

```

```

33     }
34     if (rdi - 500 > minRdi) {
35         int days = (int) ((diff / 0.5) * 7);
36         Calendar targetDate1 = Calendar.getInstance();
37         targetDate1.add(Calendar.DAY_OF_MONTH, days);
38         result.put(rdi - 500, targetDate1);
39     }
40     if (rdi - 1000 > minRdi) {
41         int days = (int) (diff * 7);
42         Calendar targetDate1 = Calendar.getInstance();
43         targetDate1.add(Calendar.DAY_OF_MONTH, days);
44         result.put(rdi - 1000, targetDate1);
45     } else {
46         double maxRdiDif = rdi - minRdi;
47         int days = (int) ((diff / (maxRdiDif / 1000)) * 7);
48         Calendar targetDate1 = Calendar.getInstance();
49         targetDate1.add(Calendar.DAY_OF_MONTH, days);
50         result.put((double) minRdi, targetDate1);
51     }
52 }
53
54 Iterator myVeryOwnIterator = result.keySet().iterator();
55 while(myVeryOwnIterator.hasNext()) {
56     Double rd = (Double)myVeryOwnIterator.next();
57     Calendar date = (Calendar) result.get(rd);
58     SimpleDateFormat format = new SimpleDateFormat("d/MMMM/
59             yyyy");
60     String targetDate = format.format(date.getTime());
61     Log.i("result", "" + rd + " " + targetDate);
62 }
63 return result;
64 }
```

Код В.6: Хэрэглэгчийн зорилгод үндэслэн боломжтой хугацаа болон илчлэгийн сонголтуудыг тооцож буй код

```

1 public View onCreateView(@NonNull LayoutInflater inflater,
2                         ViewGroup container, Bundle
3                         savedInstanceState) {
4     BottomNavigationView view = getActivity().findViewById(R.id
5             .nav_view);
6     actionBar = getActivity().findViewById(R.id.
7             action_button);
8     actionBar.setVisibility(View.VISIBLE);
9     if (view.getVisibility() == View.GONE)
10        view.setVisibility(View.VISIBLE);
11     dashboardViewModel =
12         new ViewModelProvider(requireActivity()).get(
13             DashboardViewModel.class);
14     View root = inflater.inflate(R.layout.fragment_dashboard,
15         container, false);
```

```

11     profile = Profile.getInstance();
12     mAuth = FirebaseAuth.getInstance();
13     currentUser = mAuth.getCurrentUser();
14     database = FirebaseDatabase.getInstance();
15     myRef = database.getReference("profiles").child(currentUser
16         .getUid()).child("history");
16     myWeightRef = database.getReference("profiles").child(
17         currentUser.getUid());
17     sdf = new SimpleDateFormat("yyyy-MM-dd");
18     initView(root);
19     today = Calendar.getInstance();
20
21     setValues(today, today);
22
23     mCardViewBreakfast.setOnClickListener(this);
24     mCardViewLunch.setOnClickListener(this);
25     mCardViewDinner.setOnClickListener(this);
26     mCardViewSnack.setOnClickListener(this);
27     mImgViewLeft.setOnClickListener(this);
28     mImgViewRight.setOnClickListener(this);
29
30     myWeightRef.addValueEventListener(new ValueEventListener()
31     {
32         @Override
33         public void onDataChange(@NonNull DataSnapshot
34             dataSnapshot) {
35             mTxtWeight.setText(String.valueOf(profile.
36                 getHistory().getDate(today).getWeight().
37                 getAmount()));
38         }
39
40         @Override
41         public void onCancelled(@NonNull DatabaseError
42             databaseError) {
43     });
44
45     return root;
46 }
47
48
49
50
51
52
53
54

```

```

55         e.printStackTrace();
56     }
57 }
58
59     @Override
60     public void onResume() {
61         super.onResume();
62         if (dashboardViewModel.getDate().getValue() != null) {
63             if (profile.getHistory().getDate(dashboardViewModel.
64                 getDate().getValue()) != null)
65                 today = dashboardViewModel.getDate().getValue();
66             else
67                 today = dashboardViewModel.getLastDate().getValue()
68                 ;
69             setValues(today, dashboardViewModel.getDate().getValue
70                 ());
71         }
72     }
73
74     private void initView(View root) {
75         mConstraintLayout = getActivity().findViewById(R.id.layout)
76             ;
77         mTxtTotalConsumedKcal = root.findViewById(R.id.
78             txt_total_consumed_kcal);
79         mTxtRemainingKcal = root.findViewById(R.id.
80             txt_remaining_kcal);
81         mTxtWeight = root.findViewById(R.id.txt_weight);
82         mTxtCarbs = root.findViewById(R.id.txt_carbs);
83         mTxtProtein = root.findViewById(R.id.txt_protein);
84         mTxtFat = root.findViewById(R.id.txt_fat);
85         mTxtCarbsState = root.findViewById(R.id.txt_carbs_state);
86         mTxtProteinState = root.findViewById(R.id.txt_protein_state
87             );
88         mTxtFatState = root.findViewById(R.id.txt_fat_state);
89         mTxtDate = root.findViewById(R.id.txt_date);
90         mTxtbreakfastFoods = root.findViewById(R.id.
91             txt_breakfast_foods);
92         mTxtLunchFoods = root.findViewById(R.id.txt_lunch_foods);
93         mTxtDinnerFoods = root.findViewById(R.id.txt_dinner_foods);
94         mTxtSnackFoods = root.findViewById(R.id.txt_snack_foods);
95         mTxtBreakfastConsumedKcal = root.findViewById(R.id.
96             txt_breakfast_consumed_kcal);
97         mTxtBreakfastRecKcal = root.findViewById(R.id.
98             txt_breakfast_recommended_kcal);
99         mTxtLunchConsumedKcal = root.findViewById(R.id.
100             txt_lunch_consumed_kcal);
101         mTxtLunchRecKcal = root.findViewById(R.id.
102             txt_lunch_recommended_kcal);
103         mTxtDinnerConsumedKcal = root.findViewById(R.id.
104             txt_dinner_consumed_kcal);
105         mTxtDinnerRecKcal = root.findViewById(R.id.
106             txt_dinner_recommended_kcal);

```

```

93     mTxtSnackConsumedKcal = root.findViewById(R.id.
94         txt_snack_consumed_kcal);
95     mTxtSnackRecKcal = root.findViewById(R.id.
96         txt_snack_recommended_kcal);
97     mProgressBarCarbs = root.findViewById(R.id.
98         progressBar_carbs);
99     mProgressBarProtein = root.findViewById(R.id.
100        progressBar_protein);
101    mProgressBarFat = root.findViewById(R.id.progressBar_fat);
102    mImgViewLeft = root.findViewById(R.id.imageView_left);
103    mImgViewRight = root.findViewById(R.id.imageView_right);
104    mImgViewBreakfastAdd = root.findViewById(R.id.
105        imageView_breakfast_add);
106    mImgViewLunchAdd = root.findViewById(R.id.
107        imageView_lunch_add);
108    mImgViewDinnerAdd = root.findViewById(R.id.
109        imageView_dinner_add);
110    mImgViewSnackAdd = root.findViewById(R.id.
111        imageView_snack_add);
112    mProgressBarRing = root.findViewById(R.id.progressBar_ring
113 );
114    mCardViewDinner = root.findViewById(R.id.cardView_dinner);
115    mCardViewSnack = root.findViewById(R.id.cardView_snack);
116    mCardViewBreakfast = root.findViewById(R.id.
117        cardView_breakfast);
118    mCardViewLunch = root.findViewById(R.id.cardView_lunch);
119    mProgressBar = getActivity().findViewById(R.id.
120        progress_bar_read_history);
121    mTxtRemKcalState = root.findViewById(R.id.
122        txt_remaining_kcal_state);
123 }
124
125     private boolean isSameDay(Calendar calendar, Calendar calendar1
126 ) {
127     return calendar.get(Calendar.DAY_OF_YEAR) == calendar1.get(
128         Calendar.DAY_OF_YEAR) && calendar.get(Calendar.YEAR) ==
129         calendar1.get(Calendar.YEAR);
130 }
131
132     private void setValues(Calendar calendar, Calendar calendar1) {
133     mTxtDate.setText(sdf.format(calendar1.getTime()));
134     mTxtWeight.setText(String.valueOf(profile.getHistory().
135         getDate(calendar).getWeight().getAmount()));
136     mTxtBreakfastRecKcal.setText(String.valueOf((int) profile.
137         getCalorieGoal() * 20 / 100));
138     mTxtLunchRecKcal.setText(String.valueOf((int) profile.
139         getCalorieGoal() * 25 / 100));
140     mTxtDinnerRecKcal.setText(String.valueOf((int) profile.
141         getCalorieGoal() * 35 / 100));
142     mTxtSnackRecKcal.setText(String.valueOf((int) profile.
143         getCalorieGoal() * 20 / 100));
144     if (!isSameDay(calendar, calendar1)) {

```

```

125     mTxtRemainingKcal.setText(String.valueOf((int) profile.
126         getCalorieGoal()));
127     mTxtTotalConsumedKcal.setText("0");
128     mTxtCarbs.setText(String.valueOf((int) (profile.
129         getCalorieGoal() * 50 / 100) / 4));
130     mTxtCarbsState.setText("    ");
131     mTxtProtein.setText(String.valueOf((int) (profile.
132         getCalorieGoal() * 20 / 100) / 4));
133     mTxtProteinState.setText("    ");
134     mTxtFat.setText(String.valueOf((int) (profile.
135         getCalorieGoal() * 30 / 100) / 9));
136     mTxtFatState.setText("    ");
137     mProgressBarCarbs.setProgress(0);
138     mProgressBarProtein.setProgress(0);
139     mProgressBarFat.setProgress(0);
140     mProgressBarRing.setProgress(0);
141     mTxtbreakfastFoods.setText(" ");
142     mTxtLunchFoods.setText(" ");
143     mTxtDinnerFoods.setText(" ");
144     mTxtSnackFoods.setText(" ");
145 } else {
146     mTxtTotalConsumedKcal.setText(String.valueOf((int)
147         profile.getHistory().getDate(calendar).getCalorie().
148             getConsumed()));
149     mTxtWeight.setText(String.valueOf(profile.getHistory().
150         getDate(calendar).getWeight().getAmount()));
151     double percentage = (profile.getHistory().getDate(
152         calendar).getCalorie().getConsumed() / profile.
153             getCalorieGoal()) * 100;
154     if (percentage > 100) {
155         mProgressBarRing.setProgress((int) percentage -
156             100, true);
157         mProgressBarRing.getProgressDrawable().
158             setColorFilter(ContextCompat.getColor(
159                 requireContext(), R.color.colorOrange),
160                     android.graphics.PorterDuff.Mode.MULTIPLY);
161         mTxtRemainingKcal.setText(String.valueOf(Math.abs((
162             int) profile.getHistory().getDate(calendar).
163                 getCalorie().getRemaining())));
164         mTxtRemKcalState.setText("    ");
165     } else {
166         mProgressBarRing.setProgress((int) percentage, true
167             );
168         mProgressBarRing.getProgressDrawable().
169             setColorFilter(ContextCompat.getColor(
170                 requireContext(), R.color.colorWhite),
171                     android.graphics.PorterDuff.Mode.MULTIPLY);
172         mTxtRemainingKcal.setText(String.valueOf((int)

```

```

        profile.getHistory().getDate(calendar).
            getCalorie().getRemaining()));
        mTxtRemKcalState.setText("   ");
    }
    List<MealFood> mealFoods = profile.getHistory().getDate(
        calendar).getMealPlan().getBreakfast().getMealFoods
    ();
    String mealString = "";
    for (int i = 0; i < mealFoods.size(); i++) {
        if (i + 1 == mealFoods.size())
            mealString += mealFoods.get(i).getName();
        else
            mealString += mealFoods.get(i).getName() + ", "
                ;
    }
    mTxtbreakfastFoods.setText(mealString);
    mealFoods = profile.getHistory().getDate(calendar).
        getMealPlan().getLunch().getMealFoods();
    mealString = "";
    for (int i = 0; i < mealFoods.size(); i++) {
        if (i + 1 == mealFoods.size())
            mealString += mealFoods.get(i).getName();
        else
            mealString += mealFoods.get(i).getName() + ", "
                ;
    }
    mTxtLunchFoods.setText(mealString);
    mealFoods = profile.getHistory().getDate(calendar).
        getMealPlan().getDinner().getMealFoods();
    mealString = "";
    for (int i = 0; i < mealFoods.size(); i++) {
        if (i + 1 == mealFoods.size())
            mealString += mealFoods.get(i).getName();
        else
            mealString += mealFoods.get(i).getName() + ", "
                ;
    }
    mTxtDinnerFoods.setText(mealString);
    mealFoods = profile.getHistory().getDate(calendar).
        getMealPlan().getSnack().getMealFoods();
    mealString = "";
    for (int i = 0; i < mealFoods.size(); i++) {
        if (i + 1 == mealFoods.size())
            mealString += mealFoods.get(i).getName();
        else
            mealString += mealFoods.get(i).getName() + ", "
                ;
    }
    mTxtSnackFoods.setText(mealString);

    if (profile.getHistory().getDate(today).getMealPlan().
        isRead()) {

```

```

200         setNutrition();
201
202     } else {
203         setBlurry();
204         FoodService foodService = new FoodService();
205         MealServingTask mealServingTask = foodService.
206             getMealFoods(today);
207         mealServingTask.setOnPostExecuteListener(() -> {
208             setNutrition();
209             removeBlurry();
210         });
211     }
212 }
213
214 private void removeBlurry() {
215     Blurry.delete(mConstraintLayout);
216     mProgressBar.setVisibility(View.GONE);
217     mCardViewSnack.setClickable(true);
218     mCardViewDinner.setClickable(true);
219     mCardViewLunch.setClickable(true);
220     mCardViewBreakfast.setClickable(true);
221     mImgViewRight.setClickable(true);
222     mImgViewLeft.setClickable(true);
223 }
224
225 private void setBlurry() {
226     mCardViewSnack.setClickable(false);
227     mCardViewDinner.setClickable(false);
228     mCardViewLunch.setClickable(false);
229     mCardViewBreakfast.setClickable(false);
230     mImgViewRight.setClickable(false);
231     mImgViewLeft.setClickable(false);
232     Blurry.with(getContext())
233         .radius(10)
234         .sampling(5)
235         .animate(500)
236         .onto(mConstraintLayout);
237     mProgressBar.setVisibility(View.VISIBLE);
238 }
239
240 private void setNutrition() {
241     mTxtBreakfastConsumedKcal.setText(String.valueOf((int)
242         profile.getHistory().getDate(today).getMealPlan().
243         getBreakfast().calcTotalCalories())));
244     mTxtLunchConsumedKcal.setText(String.valueOf((int) profile.
245         getHistory().getDate(today).getMealPlan().getLunch().
246         calcTotalCalories()));
247     mTxtDinnerConsumedKcal.setText(String.valueOf((int) profile
248         .getHistory().getDate(today).getMealPlan().getDinner().
249         calcTotalCalories()));
250     mTxtSnackConsumedKcal.setText(String.valueOf((int) profile.

```

```

        getHistory().getDate(today).getMealPlan().getSnack() .
        calcTotalCalories())));
245    double recCarb = (profile.getCalorieGoal() * 50 / 100) / 4;
246    double recProtein = (profile.getCalorieGoal() * 20 / 100) /
        4;
247    double recFat = (profile.getCalorieGoal() * 30 / 100) / 9;
248    double consumedCarb = profile.getHistory().getDate(today).
        getMealPlan().calcTotalCarb();
249    double consumedProtein = profile.getHistory().getDate(today)
        .getMealPlan().calcTotalProtein();
250    double consumedFat = profile.getHistory().getDate(today).
        getMealPlan().calcTotalFat();
251    if (recCarb - consumedCarb > 0) {
252        mTxtCarbs.setText(String.valueOf((int) (recCarb -
            consumedCarb)));
253        mTxtCarbsState.setText("      ");
254    } else {
255        mTxtCarbs.setText(String.valueOf((int) (consumedCarb -
            recCarb)));
256
            mTxtCarbsState.setText("      ");
257    }
258    if (recProtein - consumedProtein > 0) {
259        mTxtProtein.setText(String.valueOf((int) (recProtein -
            consumedProtein)));
260        mTxtProteinState.setText("      ");
261    } else {
262        mTxtProtein.setText(String.valueOf((int) (
            consumedProtein - recProtein)));
263        mTxtProteinState.setText("      ");
264    }
265    if (recFat - consumedFat > 0) {
266        mTxtFat.setText(String.valueOf((int) (recFat -
            consumedFat)));
267        mTxtFatState.setText("      ");
268    } else {
269        mTxtFat.setText(String.valueOf((int) (consumedFat -
            recFat)));
270        mTxtFatState.setText("      ");
271    }
272    int carbPer = (int) ((consumedCarb * 100) / recCarb);
273    int proteinPer = (int) ((consumedProtein * 100) /
        recProtein);
274    int fatPer = (int) ((consumedFat * 100) / recFat);
275
276    if (carbPer > 100) {
277        LayerDrawable progressBarDrawable = (LayerDrawable)
            mProgressBarCarbs.getProgressDrawable();
278        progressBarDrawable.setColorFilter(ContextCompat.
            getColor(getApplicationContext(), R.color.colorOrange), android
            .graphics.PorterDuff.Mode.MULTIPLY);
279        progressBarDrawable.getDrawable(0).setColorFilter(

```

```

        ContextCompat.getColor(getApplicationContext(), R.color.
        colorWhite), android.graphics.PorterDuff.Mode.
        MULTIPLY);
    mProgressBarCarbs.setProgress(carbPer - 100, true);
} else {
    mProgressBarCarbs.getProgressDrawable().setColorFilter(
        ContextCompat.getColor(getApplicationContext(), R.color.
        colorWhite),
        android.graphics.PorterDuff.Mode.MULTIPLY);
    mProgressBarCarbs.setProgress(carbPer, true);
}

if (carbPer > 100) {
    LayerDrawable progressBarDrawable = (LayerDrawable)
        mProgressBarProtein.getProgressDrawable();
    progressBarDrawable.setColorFilter(ContextCompat.
        getColor(getApplicationContext(), R.color.colorOrange), android
        .graphics.PorterDuff.Mode.MULTIPLY);
    progressBarDrawable.getDrawable(0).setColorFilter(
        ContextCompat.getColor(getApplicationContext(), R.color.
        colorWhite), android.graphics.PorterDuff.Mode.
        MULTIPLY);
    mProgressBarProtein.setProgress(carbPer - 100, true);
} else {
    mProgressBarProtein.getProgressDrawable().
        setColorFilter(ContextCompat.getColor(getApplicationContext(),
        R.color.colorWhite),
        android.graphics.PorterDuff.Mode.MULTIPLY);
    mProgressBarProtein.setProgress(carbPer, true);
}

if (carbPer > 100) {
    LayerDrawable progressBarDrawable = (LayerDrawable)
        mProgressBarFat.getProgressDrawable();
    progressBarDrawable.setColorFilter(ContextCompat.
        getColor(getApplicationContext(), R.color.colorOrange), android
        .graphics.PorterDuff.Mode.MULTIPLY);
    progressBarDrawable.getDrawable(0).setColorFilter(
        ContextCompat.getColor(getApplicationContext(), R.color.
        colorWhite), android.graphics.PorterDuff.Mode.
        MULTIPLY);
    mProgressBarFat.setProgress(carbPer - 100, true);
} else {
    mProgressBarFat.getProgressDrawable().setColorFilter(
        ContextCompat.getColor(getApplicationContext(), R.color.
        colorWhite),
        android.graphics.PorterDuff.Mode.MULTIPLY);
    mProgressBarFat.setProgress(carbPer, true);
}

@Override

```

```

312     public void onClick(View v) {
313
314         switch (v.getId()) {
315             case R.id.imageView_left:
316                 onLeftClick();
317                 break;
318             case R.id.imageView_right:
319                 onRightClick();
320                 break;
321             case R.id.cardView_breakfast:
322                 onAddClick("mealName", " ");
323                 break;
324             case R.id.cardView_lunch:
325                 onAddClick("mealName", " ");
326                 break;
327             case R.id.cardView_dinner:
328                 onAddClick("mealName", " ");
329                 break;
330             case R.id.cardView_snack:
331                 onAddClick("mealName", " ");
332                 break;
333         }
334     }
335
336     private void onLeftClick() {
337         Calendar calendar = Calendar.getInstance();
338         Calendar calendar1 = Calendar.getInstance();
339         try {
340             calendar.setTime(sdf.parse(mTxtDate.getText().toString()));
341             calendar.add(Calendar.DATE, -1);
342             calendar1.setTime(calendar.getTime());
343         } catch (ParseException e) {
344             e.printStackTrace();
345         }
346         myRef.addValueEventListener(new ValueEventListener() {
347             @Override
348             public void onDataChange(@NonNull DataSnapshot
349                 dataSnapshot) {
350                 if (!after(profile.getStartDate(), calendar)) {
351                     while (after(profile.getHistory().getDate(today)
352                         .getTime(), calendar)) {
353                         if (profile.getHistory().getDate(calendar)
354                             != null) {
355                             today.setTime(calendar.getTime());
356                         } else if (dataSnapshot.child(sdf.format(
357                             calendar.getTime())).exists()) {
358                             getHistory(dataSnapshot, sdf.format(
359                                 calendar.getTime()));
360                             today.setTime(calendar.getTime());
361                         } else {
362                             calendar.add(Calendar.DATE, -1);

```

```

358         }
359     }
360     setValues(today, calendar1);
361 }
362
363     @Override
364     public void onCancelled(@NonNull DatabaseError
365         databaseError) {
366     }
367 });
368 }
369
370
371     public boolean before(Calendar yesterday, Calendar today) {
372         if(yesterday == today) return false;
373         if(yesterday == null || today == null) return false;
374         return yesterday.get(Calendar.YEAR) < today.get(Calendar.
375             YEAR) ? true :
376             yesterday.get(Calendar.YEAR) == today.get(Calendar.
377                 YEAR) && yesterday.get(Calendar.DAY_OF_YEAR) <
378                 today.get(Calendar.DAY_OF_YEAR);
379     }
380
381     public boolean after(Calendar yesterday, Calendar today) {
382         if(yesterday == today) return false;
383         if(yesterday == null || today == null) return false;
384         return yesterday.get(Calendar.YEAR) > today.get(Calendar.
385             YEAR) ? true :
386                 yesterday.get(Calendar.YEAR) == today.get(Calendar.
387                     YEAR) && yesterday.get(Calendar.DAY_OF_YEAR) >
388                     today.get(Calendar.DAY_OF_YEAR);
389     }
390
391     private void onRightClick() {
392         Calendar calendar = Calendar.getInstance();
393         try {
394             calendar.setTime(sdf.parse(mTxtDate.getText().toString
395                 ()));
396             calendar.add(Calendar.DATE, 1);
397         } catch (ParseException e) {
398             e.printStackTrace();
399         }
400         myRef.addValueEventListener(new ValueEventListener() {
401             @Override
402             public void onDataChange(@NonNull DataSnapshot
403                 dataSnapshot) {
404
405                 if (profile.getHistory().getDate(calendar) != null)
406                 {
407                     today = calendar;
408                     setValues(today, calendar);
409                 }
410             }
411         });
412     }

```

```

400         }
401         } else if (dataSnapshot.child(sdf.format(calendar.
402             getTime())).exists()) {
403             getHistory(dataSnapshot, sdf.format(calendar.
404                 getTime()));
405             today = calendar;
406             setValues(today, calendar);
407         }
408     }
409
410     @Override
411     public void onCancelled(@NonNull DatabaseError
412         databaseError) {
413
414     }
415 }
416
417 private void onAddClick(String key, String value) {
418     Bundle bundle = new Bundle();
419     bundle.putString(key, value);
420     navigate(bundle);
421 }
422
423 private void navigate(Bundle bundle) {
424     bundle.putString("mealDate", mTxtDate.getText().toString())
425         ;
426     Calendar calendar = Calendar.getInstance();
427     try {
428         calendar.setTime(sdf.parse(mTxtDate.getText().toString
429             ()));
430     } catch (ParseException e) {
431         e.printStackTrace();
432     }
433     bundle.putString("valueDate", sdf.format(today.getTime()));
434     if (!profile.getHistory().getDate(today).getMealPlan().
435         getMealByName(bundle.getString("mealName")).getMealFoods
436         ().isEmpty() && isSameDay(today, calendar))
437         Navigation.findNavController(this.requireView()).
438             navigate(R.id.
439                 action_navigation_dashboard_to_navigation_meal_detail
440                 , bundle);
441     else {
442         Navigation.findNavController(this.requireView()).
443             navigate(R.id.
444                 action_navigation_dashboard_to_navigation_search,
445                 bundle);
446     }
447 }
448
449 private void getHistory(DataSnapshot dataSnapshot, String today

```

```

        )
439     double bmi = dataSnapshot.child(today).child("bmi").
440         getValue(Double.class);
440     double bodyFat = dataSnapshot.child(today).child("bodyFat")
441         .getValue(Double.class);
441     double weight = dataSnapshot.child(today).child("weight").
442         child("amount").getValue(Double.class);
442     double consumed = dataSnapshot.child(today).child("calorie"
443         ).child("consumed").getValue(Double.class);
443     double remaining = dataSnapshot.child(today).child("calorie"
444         ).child("remaining").getValue(Double.class);
444     double drunk = dataSnapshot.child(today).child("water").
445         child("drunk").getValue(Double.class);

446     Calendar calendar = Calendar.getInstance();
447     try {
448         calendar.setTime(sdf.parse(today));
449     } catch (ParseException e) {
450         e.printStackTrace();
451     }
452     MealPlan mealPlan = new MealPlan();
453     if (profile.getHistory().getDate(calendar) != null)
454         mealPlan = profile.getHistory().getDate(calendar).
455             getMealPlan();
455     if (dataSnapshot.child(today).child("mealPlan").child(""
456         ).exists())
456         addMealFood(mealPlan, dataSnapshot.child(today).child(""
457             "mealPlan").child("").getChildren(), "");
457     if (dataSnapshot.child(today).child("mealPlan").child(""
458         ).exists())
458         addMealFood(mealPlan, dataSnapshot.child(today).child(""
459             "mealPlan").child("").getChildren(), "");
459     if (dataSnapshot.child(today).child("mealPlan").child(""
460         ).exists())
460         addMealFood(mealPlan, dataSnapshot.child(today).child(""
461             "mealPlan").child("").getChildren(), "");
461     if (dataSnapshot.child(today).child("mealPlan").child(""
462         ).exists())
462         addMealFood(mealPlan, dataSnapshot.child(today).child(""
463             "mealPlan").child("").getChildren(), "");

464     profile.getHistory().addDate(new com.example.myhealth.model
465         .Date(calendar, bmi, bodyFat, new Weight(weight), new
466         Calorie(consumed, remaining), new Water(drunk), mealPlan
467         ));
467 }

468     private void addMealFood(MealPlan mealPlan, Iterable<
469         DataSnapshot> children, String mealName) {
470         for (DataSnapshot mealSnapShot : children)
471             if (mealSnapShot.child("brandName").exists())
472                 mealPlan.getMealByName(mealName).addFood(new

```

```

        MealFood(mealSnapShot.child("foodId").getValue(
    Long.class), mealSnapShot.getKey(), mealSnapShot
    .child("type").getValue(String.class),
    mealSnapShot.child("description").getValue(
    String.class), mealSnapShot.child("brandName").
    getValue(String.class), mealSnapShot.child(""
    servingAmount").getValue(Double.class),
    mealSnapShot.child("servingId").getValue(Long.
    class)));
471
472     else
        mealPlan.getMealByName(mealName).addFood(new
            MealFood(mealSnapShot.child("foodId").getValue(
    Long.class), mealSnapShot.getKey(), mealSnapShot
    .child("type").getValue(String.class),
    mealSnapShot.child("description").getValue(
    String.class), mealSnapShot.child("servingAmount
    ").getValue(Double.class), mealSnapShot.child(""
    servingId").getValue(Long.class)));
473
    }

```

Код В.7: Хоолны төлөвлөгөө болон түүхийн хэсгийн код

```

1  public class FoodsTask extends AsyncTask {
2      private String url;
3      private FoodContainer foodContainer;
4      private List<CompactFood> foods;
5      private OnUpdateFoodsListener mListener;
6
7      public FoodsTask(String url) {
8          this.url = url;
9      }
10
11     public interface OnUpdateFoodsListener {
12         void onUpdateFoods();
13     }
14
15     public void setOnUpdateFoodsListener(OnUpdateFoodsListener
16         listener) {
17         this.mListener = listener;
18     }
19
20     @Override
21     protected Object doInBackground(Object[] objects) {
22         foodContainer = FoodContainer.getInstance();
23         foods = foodContainer.getFoods();
24         foods.clear();
25         HttpHandler handler = new HttpHandler();
26         String jsonStr = handler.makeServiceCall(url);
27         if (jsonStr != null) {
28             try {
29                 JSONObject jsonObject = new JSONObject(jsonStr);

```

```

30         getJSONObject("foods");
31         JSONArray jsonFoodArray = jsonFoodObject.
32             getJSONArray("food");
33
34         for (int i = 0; i < jsonFoodArray.length(); i++) {
35             JSONObject jsonFoodArrayItem = jsonFoodArray.
36                 getJSONObject(i);
37             if (!jsonFoodArrayItem.has("brand_name"))
38                 foods.add(new CompactFood(jsonFoodArrayItem
39                     .getLong("food_id"), jsonFoodArrayItem.
40                     getString("food_name"),
41                     jsonFoodArrayItem.getString("food_type")
42                     , jsonFoodArrayItem.getString(""
43                     food_description")));
44             else
45                 foods.add(new CompactFood(jsonFoodArrayItem
46                     .getLong("food_id"), jsonFoodArrayItem.
47                     getString("food_name"),
48                     jsonFoodArrayItem.getString("food_type")
49                     , jsonFoodArrayItem.getString(""
50                     food_description), jsonFoodArrayItem.
51                     getString("brand_name")));
52
53         }
54     } catch (JSONException e) {
55         e.printStackTrace();
56     }
57 }
58
59     @Override
60     protected void onPostExecute(Object o) {
61         super.onPostExecute(o);
62         mListener.onUpdateFoods();
63     }
64 }
```

Код В.8: Хэрэглэгчийн хайсан хүнсний жагсаалтыг fatsecret -ийн өгөгдлийн сангаас авч хэсгийн код

```

1 while (i < count) {
2     calendar.add(Calendar.DATE, -1);
3     if (dataSnapshot.child("history").child(sdf.format(calendar.
4         getTime())).exists()) {
5         try {
6             getHistory(dataSnapshot, sdf.format(calendar.getTime()))
7                 );
8         } catch (ParseException e) {
9             e.printStackTrace();
10        }
11    }
12 }
```

```

9     i++;
10    }
11 }

```

Код В.9: Хэрэглэгчийн өмнөх түүхийг унших хэсгийн код

```

1 public void update() {
2     Serving serving = (Serving) mSpinnerSerDesc.getSelectedItem();
3     mealFood.setServing(serving);
4     mealFood.setServingId(serving.getServingId());
5     mealFood.setServingAmount(Double.parseDouble(mETxtSerAmount.
6         getText().toString()));
7     myRef.child(food.getName()).setValue(new MealTemp(mealFood.
8         getId(), mealFood.getDescription(), mealFood.
9         getServingAmount(), mealFood.getServingId(), mealFood.
10        getType()), (databaseError, databaseReference) -> {
11            if (databaseError == null)
12                Navigation.findNavController(this.requireView()).
13                    popBackStack();
14        });
15    }

```

Код В.10: Хүнсний хэмжээг шинэчлэх хэсгийн код

```

1 public void remove() {
2     Profile.getInstance().getHistory().getDate(mealDate).
3         getMealPlan().getMealByName(mealName).removeMealFood(
4             mealFood);
5     myRef.child(mealFood.getName()).removeValue((databaseError,
6         databaseReference) -> {
7         if (databaseError == null)
8             Navigation.findNavController(this.requireView()).
9                 popBackStack();
10    });
11 }

```

Код В.11: Хүнсийг цэснээс хасах хэсгийн код

```

1 public void add() {
2     Serving serving = (Serving) mSpinnerSerDesc.getSelectedItem();
3     MealFood mealFood = new MealFood(food.getId(), food.getName(),
4         food.getType(), food.getDescription(), food.getBrandName(),
5         Double.parseDouble(mETxtSerAmount.getText().toString()),
6         serving);
7     Profile.getInstance().getHistory().getDate(mealDate).
8         getMealPlan().getMealByName(mealName).addFood(mealFood);
9     Navigation.findNavController(this.requireView()).popBackStack()
10    ;
11 }

```

Код В.12: Хүнсийг цэсэд нэмэх хэсгийн код

```

1 public String buildFoodsSearchUrl(String query, int pageNumber)
2     throws Exception {
3     List<String> params = new ArrayList<String>(Arrays.asList(
4         generateOauthParams()));
5     String[] template = new String[1];
6     params.add("method=foods.search");
7     params.add("max_results=20");
8     params.add("page_number=" + pageNumber);
9     params.add("search_expression=" + encode(query));
10    params.add("oauth_signature=" + sign(HTTP_METHOD, APP_URL,
11        params.toArray(template)));
12
13    return APP_URL + "?" + paramify(params.toArray(template));
14 }
```

Код В.13: Хайх хүнсний url бэлдэх хэсгийн код

```

1 public String buildFoodGetUrl(Long id) throws Exception {
2     List<String> params = new ArrayList<String>(Arrays.asList(
3         generateOauthParams()));
4     String[] template = new String[1];
5     params.add("method=food.get");
6     params.add("food_id=" + id);
7     params.add("oauth_signature=" + sign(HTTP_METHOD, APP_URL,
8         params.toArray(template)));
9
9 } 
```

Код В.14: Хүнсний дэлгэрэнгүйг авах url бэлдэх хэсгийн код

```

1 private void startAlarm(Calendar c, int id) {
2     AlarmManager alarmManager = (AlarmManager) requireActivity().
3         getSystemService(Context.ALARM_SERVICE);
4     Intent intent = new Intent(getContext(), AlertReceiver.class);
5     PendingIntent pendingIntent = PendingIntent.getBroadcast(
6         getContext(), id, intent, 0);
7     assert alarmManager != null;
8     alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP, c.
9         getTimeInMillis(),
10            AlarmManager.INTERVAL_DAY, pendingIntent);
11 }
```

Код В.15: Сануулга эхлүүлэх хэсгийн код

```

1 private void cancelAlarm(int id) {
2     AlarmManager alarmManager = (AlarmManager) requireActivity().
3         getSystemService(Context.ALARM_SERVICE);
4     Intent intent = new Intent(getContext(), AlertReceiver.class);
5     PendingIntent pendingIntent = PendingIntent.getBroadcast(
6         getContext(), id, intent, 0);
7     assert alarmManager != null;
```

```
6     alarmManager.cancel(pendingIntent);  
7 }
```

Код В.16: Сануулга зогсоох хэсгийн код