

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

Ганбаатарын Одгаваа

**ПРОГРАМ ХАНГАМЖИЙН ИНЖЕНЕР
ХӨТӨЛБӨРИЙН ҮЙЛДВЭРЛЭЛИЙН
ДАДЛАГЫН ТАЙЛАН
(Internship Report)**

Програм хангамж (D061302)
Үйлдвэрлэлийн дадлага

Улаанбаатар

2020 оны 03 сар 13 өдөр

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

ПРОГРАМ ХАНГАМЖИЙН ИНЖЕНЕР ХӨТӨЛБӨРИЙН
ҮЙЛДВЭРЛЭЛИЙН ДАДЛАГЫН ТАЙЛАН

(Internship Report)

Програм хангамж (D061302)
Үйлдвэрлэлийн дадлага

Удирдагч:	_____	Б.Эрдэнэ-Очир
Хамтран удирдагч:	_____	Ч.Оюунгэрэл
Гүйцэтгэсэн:	_____	Г.Одгаваа (17B1NUM1464)

Улаанбаатар

2020 оны 03 сар 13 өдөр

ГАРЧИГ

УДИРТГАЛ.....	1
1. СИСТЕМИЙН ТАНИЛЦУУЛГА	2
1.1 Байгууллагын танилцуулга	2
1.2 Байгууллагын үйл ажиллагаа, үйлчилгээ	2
1.3 Тухайн систем дээр өөрийн ажилласан хэсэг	2
2. ИЖИЛ СИСТЕМИЙН СУДАЛГАА	3
2.1 Ижил төстэй системүүдийн жишээ	3
2.2 Системийг бүтээхэд ашигласан технологи, хэрэгсэл	3
3. СИСТЕМИЙН ШААРДЛАГА	5
3.1 Функционал ба функционал бус хэрэглэгчийн шаардлага	5
3.2 Use-case диаграм	6
4. СИСТЕМИЙН ШИНЖИЛГЭЭ	9
4.1 Класс диаграмын ерөнхий бүтэц:	9
5. ХЭРЭГЖҮҮЛЭЛТ	13
5.1 Системийг бүтээхэд ашигласан технологи	13
5.2 Use-case бүрийн хэрэглэгчийн интерфэйс	18
5.3 Хэрэгжүүлсэн байдал	23
6. ҮР ДҮН	29
6.1 Сүүлийн үр дүн буюу production server дээр ажиллуулсан байдал ..	30
ДҮГНЭЛТ	32
ХАВСРАЛТ	34
6.2 Main component-н кодын хэсэг	34
6.3 Main component дээр Unit Test бичсэн байдал	41

УДИРТГАЛ

Үйлвэрлэлийн дадлага нь оюутны танхимд эзэмшсэн онолын мэдлэг, чадварыг практик дадлаар баяжуулах зорилгоор аж ахуй нэгж, байгууллага дээр зохион байгуулагддаг.

Миний бие Одгаваа нь энэхүү үйлдвэрлэлийн дадлагын хүрээнд танхимын сургалтаар эзэмшсэн онолын мэдлэг, чадвараа програм хангамжийн үйлдвэрлэлийн түвшний үйл ажиллагаатай холбон, бодит практик дээр гүнзгийрүүлэн судалж, ажиглалт, боловсруулалт хийж шинээр сурсан мэдлэг туршлагадаа тулгуурлан дүн шинжилгээ хийн үйлдвэрлэлийн дадлагын тайлангаа бэлтгэсэн билээ.

Сонгосон байгууллага болох Эрин Системс ХХК дээр дадлага хийх хугацаандаа гүйцэтгэсэн ажил, даалгаваруудыг онолын мэдлэг дээр тулгуурлан мөн удирдагчийн зөвлөмж чиглүүлгийн дагуу олон талын эх сурвалжаас мэдээлэл цуглуулж, түүн дээрээ боловсруулалт хийж, шинээр сурч мэдсэн олон шинэ зүйлийн талаарх ойлголтуудыг тайландаа тусган оруулсан билээ. Дадлагын хугацаандаа өөрийн хүсэл сонирхлын дагуу байгууллагынхаа хэрэгжүүлж буй төслийн front-end талын хөгжүүлэлтэнд гар бие оролцон ажилласан бөгөөд ингэхдээ TypeScript болон Angular програмчлалын талаар анхан шатнаас суралцаж, түүн дээрээ тулгуурлан даалгаварууд гүйцэтгэж, даалгавар бүр дээрээ Unit Test бичих, UI Test-ээр баталгаажуулах гэх мэт шинээр олж мэдсэн мэдлэг, туршлага бүрээ тайлагнан хүргэж байна .

1. СИСТЕМИЙН ТАНИЛЦУУЛГА

1.1 Байгууллагын танилцуулга

ЭРИН СИСТЕМС ХХК нь 2016 оны 2 сарын сард байгуулагдсан бөгөөд тухайн оны 4 сараас эхлэн мэдээллийн технологийн үйлчилгээ, зөвлөгөө, программ хангамжийн хөгжүүлэлтийн чиглэлээр тасралтгүй, идэвхтэй үйл ажиллагаа явуулж буй компани юм.

1.2 Байгууллагын үйл ажиллагаа, үйлчилгээ

ЭРИН СИСТЕМС нь одоогоор "Программ хангамжийн үйл ажиллагаа" чиглэлд дагнан үйл ажиллагаа явуулж байна. Программ хангамжийн үйл ажиллагаанд захиалагчийн шаардлагын дагуу программ хангамжийн бүтээгдэхүүн хөгжүүлэх, өөрийн нэрийн программ хангамжийн бүтээгдэхүүн хөгжүүлэх, үйлчлүүлэгчид нэвтрүүлэх, нэвтрүүлсэн бүтээгдэхүүнд дагалдах үйлчилгээг үзүүлэх гэх мэт багтана.

ЭРИН СИСТЕМС нь одоогийн байдлаар Сургалт Менежментийн Систем (LMS), Бизнес Процесс Менежментийн Систем(BPMS) болон Баримт Бичиг Менежментийн Систем(DMS) гэх мэт төслүүд дээр голлон төвлөрч ажиллаж байна.

1.3 Тухайн систем дээр өөрийн ажилласан хэсэг

Миний хувьд ЭРИН СИСТЕМС компанид дадлага хийх хугацаандаа Юнител Группын дотоод үйл ажиллагаанд зориулагдсан Сургалт Менежментийн Системийн Групп менежмент хэсгийн front-end талын зарим даалгаварууд дээр төвлөрч ажилласан. Энэхүү компанийн бүх төсөл Скрам менежментийн дагуу явагддаг учир скрам менежмент гэж юу болох, хэрхэн явагддаг талаарх ойлголт мэдлэгтэй болж, спринт бүрт тодорхой нэг даалгавар дээр ажиллаж түүнийхээ явцыг спринтийн төгсөлд тайлагнах журмаар ажилласан.

2. ИЖИЛ СИСТЕМИЙН СУДАЛГАА

Үйлдвэрлэлийн дадлагын хүрээнд Сургалт Менежмент Хяналтын Системийн нэгээхэн хэсэг модуль дээр ажилласан учир үүнтэй төстэй бусад сургалт менежментийн системүүдийн талаар жишээ авлаа.

2.1 Ижил төстэй системүүдийн жишээ

- Динамик сургалтын систем /onoo.mn/

Энэхүү Динамик Сургалтын Систем нь Их дээд сургууль, Бизнесийн байгууллага, Сургалтын төв, Институт, ТББ гэх мэт олон төрлийн хэрэглэгчид хэрэглэх боломжтой сургалтын үйл ажиллагаа явуулахаас гадна илүү олон төрлийн сургалтын төрөл, модулиудтай бөгөөд мөн онлайн төлбөр тооцооны системийг нэвтрүүлсэн байдал гэх мэт талаараа манай системээс ялгаатай. Ижил тал нь хэрэглэгчидэд холын зайнаас сургалтын үйл ажиллагаагаар мэдлэг олгох юм.

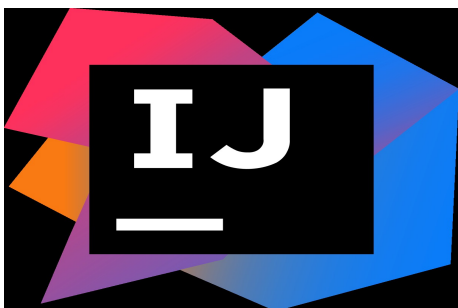
- Сургалт Менежментийн Систем /piazza.com/

Piazza систем нь мөн адил их сургуулийн оюутан, сурагчид гэх мэт тодорхой групп хэрэглэгчидэд зориулагдсан онлайн сургалтын систем ба ерөнхийдөө тухайн ямар нэг группыг удирдан сургалт явуулдаг. Багш хичээлийн тэмдэглэл, даалгавар оруулах, багш оюутнууд хоорондоо хэлэлцүүлэг явуулах боломжуудыг хангасан байдаг. Мөн хэрэглэгчдийн идэвх, хандалтыг хянах харах боломжтой систем юм. Групп үүсгэх, удирдах, багш сургалтын материал нийтлэх, суралцагчдын идэвх, хандалтыг статистикаар үзүүлэх гэх мэт олон талаар манай системтэй төстэй үйл ажиллагаатай билээ.

2.2 Системийг бүтээхэд ашигласан технологи, хэрэгсэл

Дадлагын хугацаанд ажилласан Сургалт Менежментийн Системийн (LMS Project) хувьд төслийг үйл ажиллагаа бүхэлдээ Agile Development-ийн дагуу Скрам Менежментийг барьж

ажилладаг байсан. Технологийн хувьд front-end талын хөгжүүлэлт TypeScript, Angular framework дээр, харин back-end талын хөгжүүлэлт Java хэл дээр хөгжүүлэгддэг. Мөн MongoDB-г үндсэн өгөгдлийн сангийн удирдах систем болгон ашигладаг.



3. СИСТЕМИЙН ШААРДЛАГА

Миний ажилласан систем нь Админ, Супервайзор, Ажилтан гэсэн гурван үндсэн хэрэглэгчтэй байна. Группын тохиргоо буюу Админ тохиргоо хийх модульд групп удирдах, ашиглах үүргийн хүрээнд зөвхөн админ болон супервайзорууд Группын тохиргоо хуудсанд өөрт хамаарах алба хэлтсийн ажилчдын бүртгэлийг харж, хайж, группэд хэрэглэгч нэмж, хасаж, зөөж чадна. Мөн бусад модулиудтай холбон Админ өөрт хамаарах группын хэрэглэгчидэд урамшуулал нийтлэх, хэрэглэгчидийн урамшуулалтай танилцсан эсэх статистикийг хянах зэрэг үйлдлийг групп модулийн тусламжтай хийж чадна.

3.1 Функционал ба функционал бус хэрэглэгчийн шаардлага

3.1.1 Хэрэглэгчид

- Админ
- Супер
- Ажилтан

3.1.2 Функционал хэрэглэгчийн шаардлагууд

- Админ - Системийн үндсэн цэсийг удирдах - Админ “Админ тохиргоо” цэсэнд нэвтэрч системийн үндсэн цэсийг харуулах, нуух, нэр засварлах үйлдэл хийнэ.
- Админ - Системийн дэд цэсийг удирдах - Админ “Админ тохиргоо” цэсэнд нэвтэрч системийн дэд цэсийг харуулах, нуух, нэр засварлах, шинээр нэмэх үйлдэл хийнэ.
- Админ - Системийн нэвтрэх хуудасны зураг оруулах - Админ “Админ тохиргоо” цэснээс системийн нэвтрэх хуудасны зураг сольдог байна.
- - Өөрт хамаарах группд урамшуулал нийтлэх

- Админ, Супервайзор - Өөрт хамаарах хэлтэсийн ажилчдын мэдээллийг харна.
- Админ, Супервайзор - Өөрт хамаарах хэлтэсийн ажилчтдийг группээс хайна.
- Админ, Супервайзор - Шинэ групп нэмэх, групп устгана.
- Админ, Супервайзор - Хэрэглэгч группд нэмэх, хэрэглэгч группээс хасна.
- Админ, Супервайзор - Хэрэглэгч групп хооронд зөөж, шилжүүлнэ.
- Админ, Супервайзор - Хэрэглэгчид роль онооно.

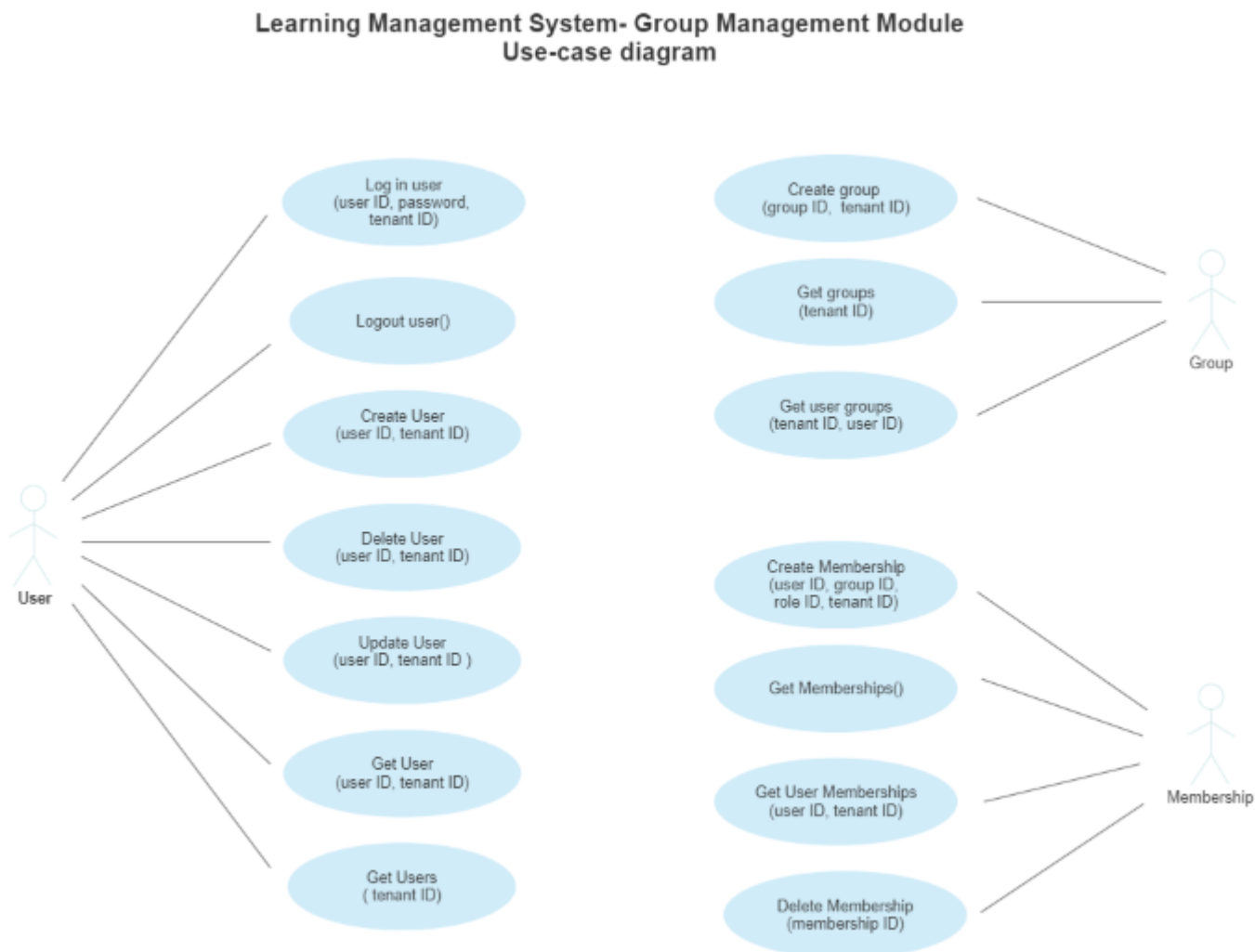
3.1.3 Функционал хэрэглэгчийн шаардлагууд

- Админ, Супервайзор - Хэрвээ групп хасах тохиолдолд бататгах диалог гарч ирэх ба зөвшөөрсөн тохиолдолд устгана.
- Админ, Супервайзор - Хэрэглэгч групп устгах үед бататгах диалог гарч ирэх ба зөвшөөрсөн тохиолдолд хасна.
- Админ, Супервайзор - Хэрэглэгч группд нэмэхдээ заавал хэрэглэгчийг ролийнх нь хүрээнд хайж олж нэмнэ.
- Админ, Супервайзор - Хэрэглэгчийг нэг түвшний групп хооронд зөөх бол өмнөх группд нь үлдээж эсвэл өмнөх группээс хасаж болно.
- Админ - Админ зөвхөн өөрт хамаарах хэрэглэгчидэд урамшуулал нийтэлж, хэрэглэгчидийн идэвхийг харах боломжтой.
- Админ, Супервайзор, Ажилтан - Нэг хэрэглэгч нэг түвшний 2 группд байх боломжгүй.

3.2 Use-case диаграм

Use-case диаграм гэдэг бол тухайн систем эсвэл системийн аль нэг модуль дээр ямар ямар үйлдлүүд хийгдэх, ямар хэрэглэгч юу юуг хийх вэ гэдгийг харуулах буюу товчхондоо

систем юу хийж чадах вэ гэдгийг бүтэцлэн харуулсан диаграм юм. Use-case диаграм гэдэг бол тухайн систем эсвэл системийн аль нэг модуль дээр ямар ямар үйлдлүүд хийгдэх, ямар хэрэглэгч юу юуг хийх вэ гэдгийг харуулах буюу товчхондоо систем юу хийж чадах вэ гэдгийг бүтэцлэн харуулсан диаграм юм.

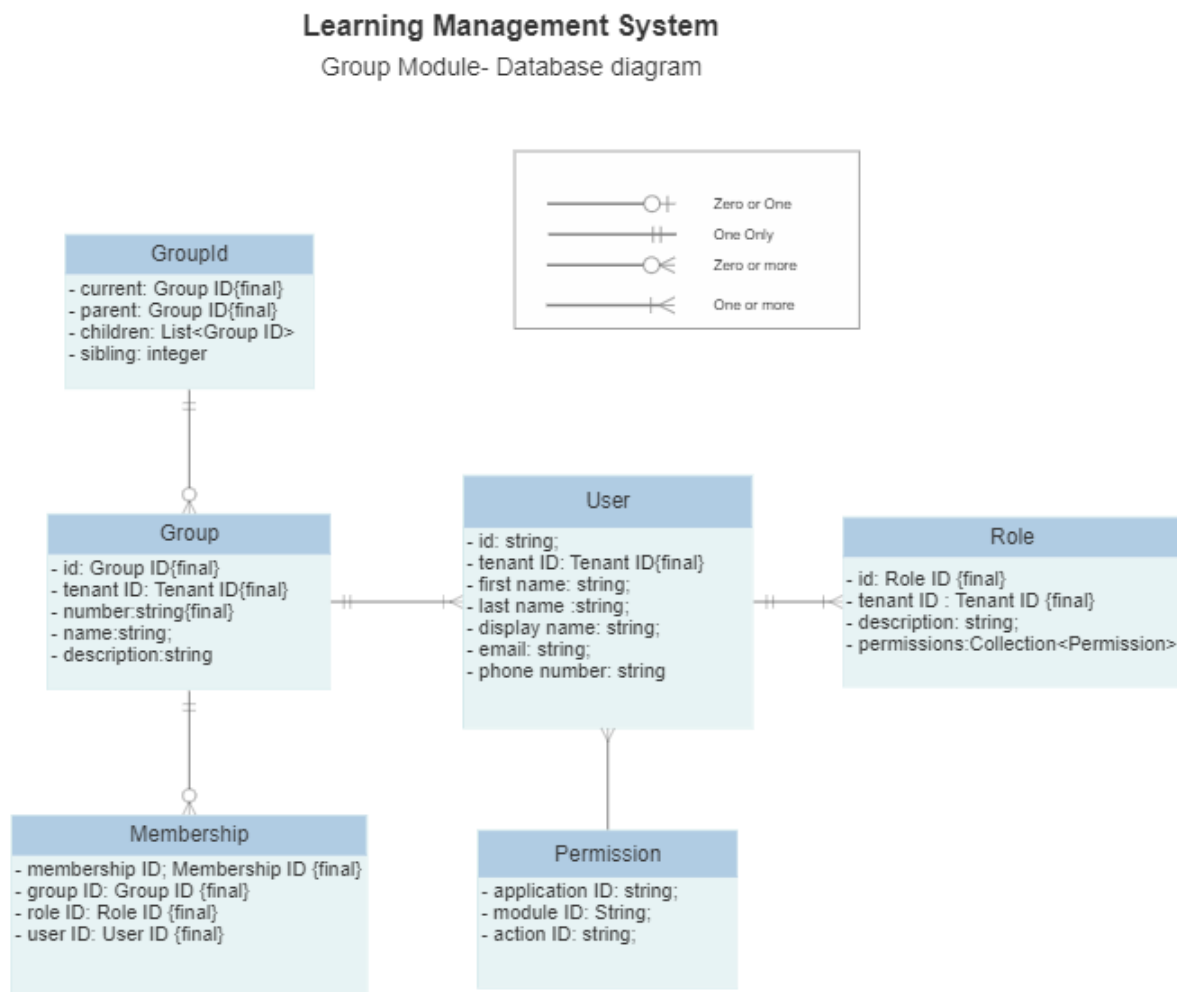


Зураг 3.1: Use-case diagram



3ypar 3.2: Use-case diagram

4. СИСТЕМИЙН ШИНЖИЛГЭЭ



Зураг 4.1: Database diagram

4.1 Класс диаграммын ерөнхий бүтэц:

- Классын нэр

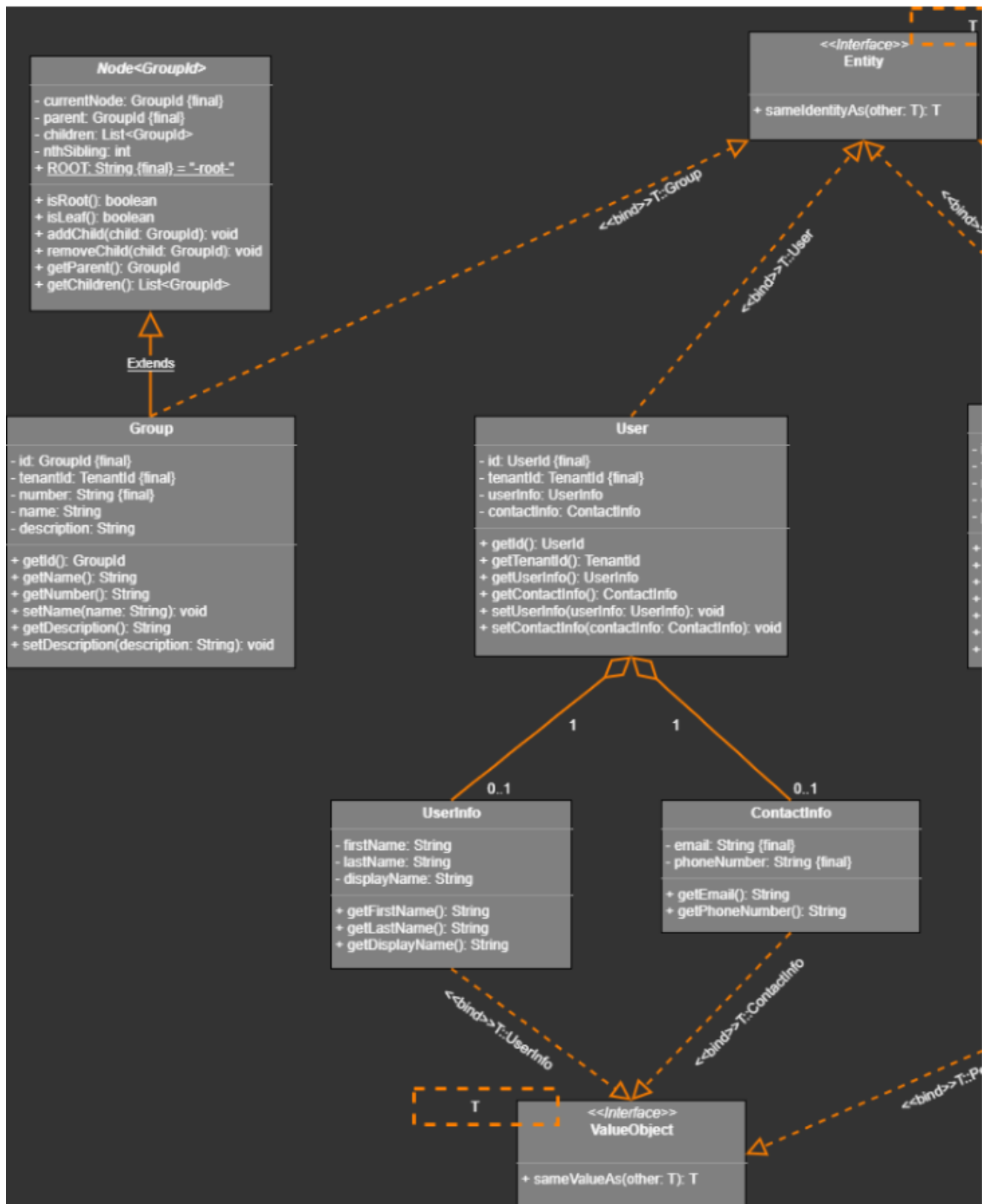
- Атрибут
- Функц

Харин класс дахь гишүүн өгөгдөл болон функцуудын хандалтын түвшинг дүрслэхдээ:

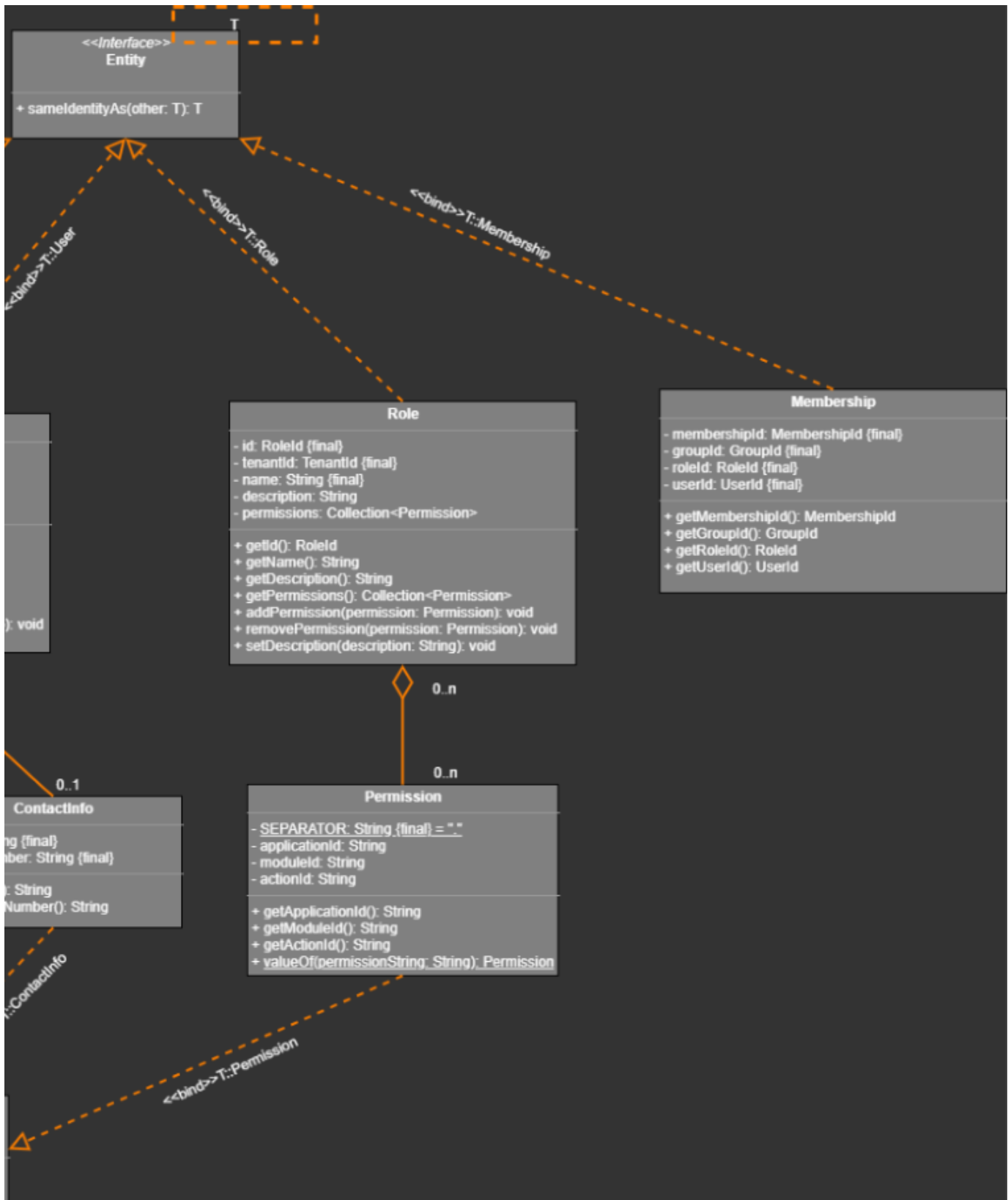
- (-) private
- (+) public

Класс хоорондын хамаарлын төрөл буюу хамаарлыг сумаар ялгаж тэмдэглэнэ.

- Dependency - нэг класс нөгө классаас хамаарч болзошгүй
- Association - Хоорондоо шууд хамааралтай классууд
- Aggregation - 'тай' холбоосоор холбогдох буюу (one to many) хамаарал. Жишээ нь групп гишүүдтэй байна.
- Composition - Aggregation хамааралтай адил боловч (one to one) хамаарал юм.
- Inheritance - нэг классын шинжүүдийг удамшуулан мөн өөрийн шинжүүдийг нэмэн шинэ класс үүсгэснийг удамшил гэнэ.



3ypar 4.2: Class diagram



Зыпар 4.3: Class diagram

5. ХЭРЭГЖҮҮЛЭЛТ

Эрин Системс компани дээр дадлага хийх хугацаандаа Сургалт Менежментийн цогц системийн Групп Менежмент модулийн front-end талын хөгжүүлэлт дээр ажиллахын тулд **TypeScript** болон **Angular CLI**-г анхан шатнаас эхлэн суралцсан. Мөн компанийн дүрэм журам, удирдлага менежмент, үйл ажиллагаатай танилцаж, хөгжүүлэлтийн үйл явцад бодитоор оролцож эхлээд Angular фраймворкыг гүнзгийрүүлэн суралцаж, ашиглахын зэрэгцээ бусад технологи болон удирдлага менежменттэй холбоотой олон шинэ зүйлсийг суралцсан билээ.

5.1 Системийг бүтээхэд ашигласан технологи

Сургалт Менежментийн Системийг (LMS Project) хөгжүүлэхэд ашигладаг технологийн хувьд front-end талын хэрэглэгчийн ашиглах интерфэйсийг TypeScript болон Angular фраймворк буюу вэб апп хөгжүүлэлтийн цогц платформыг ашигладаг. Харин back-end буюу сервер талын хөгжүүлэлт Жава програмчлал, түүний зарим фраймворкууд дээр хийгддэг. Хөгжүүлэлтын орчинд IntelliJ IDEA Community Framework-ийг ашигладаг.

Түүнчлэн манай компанийн гол давуу тал нь Agile Development буюу хурдацтай шаламгай хөгжүүлэлтийн зарчмыг барьж ажилладаг явдал байсан. Үүний хүрээнд Скрам Менежмент (Scrum Management)-ийн дагуу үйл ажиллагаагаа явуулдаг.

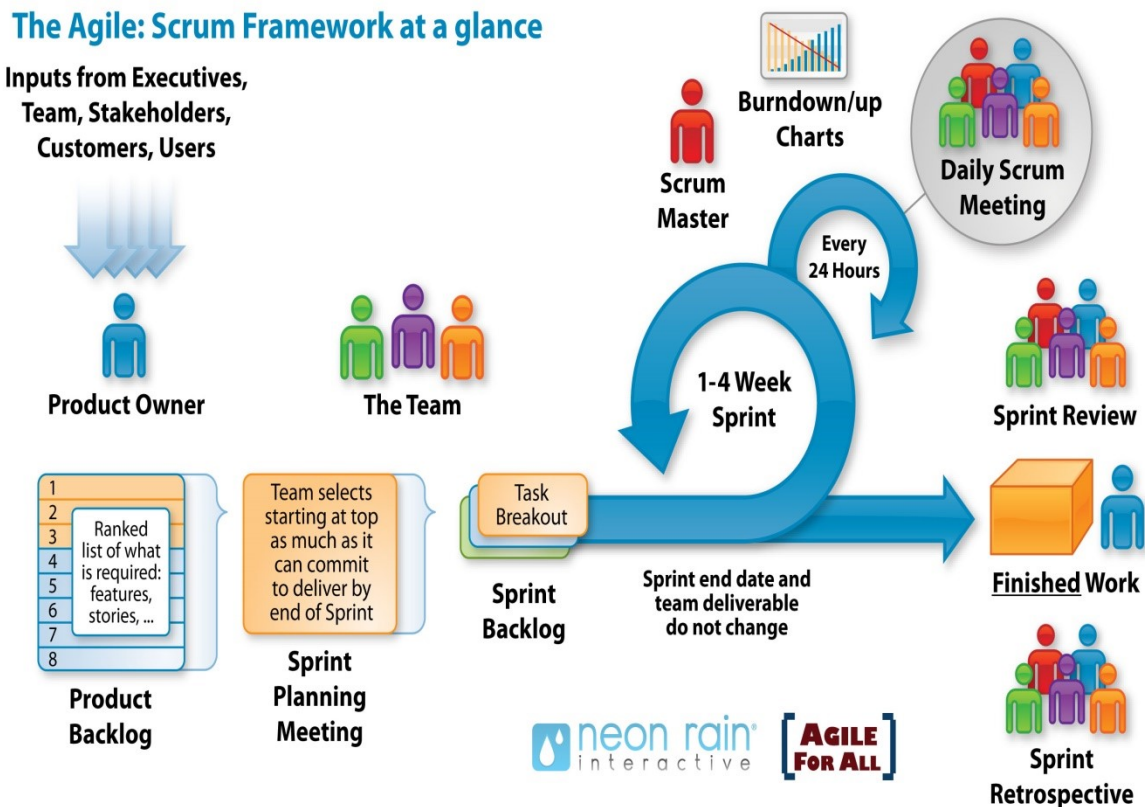
Agile Development-н арга бол програмыг маш богино хугацааны мөчлөгтэйгөөр хөгжүүлдэг учраас алдаа тутагдалыг эрт илрүүлж засах өөрчлөх боломжтойгоороо уламжлалт аргаар хөгжүүлэхээс давуу талтай. Энэхүү аргын гол цөм нь уян хатан нэгдмэл систем зах зээлд худалдаанд гаргахын тулд програм хөгжүүлэгчид багаараа нэг зорилт зорилгыг өмнөө тавьж түүндээ хүрэхэд оршдог ба уламжлалт хүрхрээ хэлбэрийн арга техникийн яг урвуу хэлбэр юм .

Скрам менежмент(Scrum Management)-ийн хэд хэдэн чухал ойлголтуудаас дурьдвал:

- Product Owner (Бүтээгдэхүүн эзэмшигч) гэдэг нь төсөл болон төслийн багийн ерөнхий төлөвлөгөний дагуу удирдах төслийн менежерийг хэлнэ.

- Scrum Master (Скрам мастер) нь төслийн хэрэгжүүлэх явцад багийн гишүүд, бүтээгдэхүүний эзэмшигч нарт Скрамын үнэ цэнэ, зарчим үйл ажиллагааг ойлгуулах, тэдгээрийг гадны нөлөө болон саадаас хамгаалах, үйл явцыг дэмжих багийн гишүүдийг тус тусын ажилдаа хариуцлага хүлээхэд тусалж ажиллах хүн юм.
- Team -Төслийг хэрэгжүүлэх ажлын баг
- Sprint - Төслийн даалгаваруудыг хэрэгжүүлэх тодорхой давтамжит хугацааны мөчлөг буюу 14-21 хоногийн хугацааны хийх ажлуудыг төлөвлөж Sprint дуусахад хийсэн ажлаа дүгнэх журмаар явагддаг.
- Product backlog - Төслийн хүрээнд хийгдэх ажлууд буюу бүтээгдэхүүний шаардлага гэж хэлж болно.
- Backlog grooming meeting - Төсөл буюу бүтээгдэхүүний хүрээнд хийгдэх ажлуудыг эрэмбэлэн үнэлэх уулзалт
- Sprint planning meeting - Спринтэд хийгдэх ажлын төлөвлөгөө гаргах уулзалт
- Sprint review meeting - Төлөвлөгөөт ажлын гүйцэтгэлийг дүгнэх уулзалт
- Sprint retrospective - Төлөвлөгөөт ажлын гүйцэтгэлийн үйл явцад багийн гишүүд бүр дүн шинжилгээ хийж хэлэлцэх
- Weekly scrum meeting - Долоо хоног үйл явцаа бүрийн уулзалт
- Daily scrum meeting - Өдөр бүрийн үйл явцаа дүгнэх уулзалт буюу Daily Stand up гэж хэлж болно.
- Sprint retrospective - Төлөвлөгөөт ажлын гүйцэтгэлийн үйл явцад багийн гишүүд бүр дүн шинжилгээ хийж хэлэлцэх ба үүнд сайн хийгдсэн үйл явц муу хийгдсэн үйл явц, цаашид сайжруулах үйл явцыг тодорхойлж JIRA-д бүртгэнэ. Мөн төлөвлөсөн ажлыг

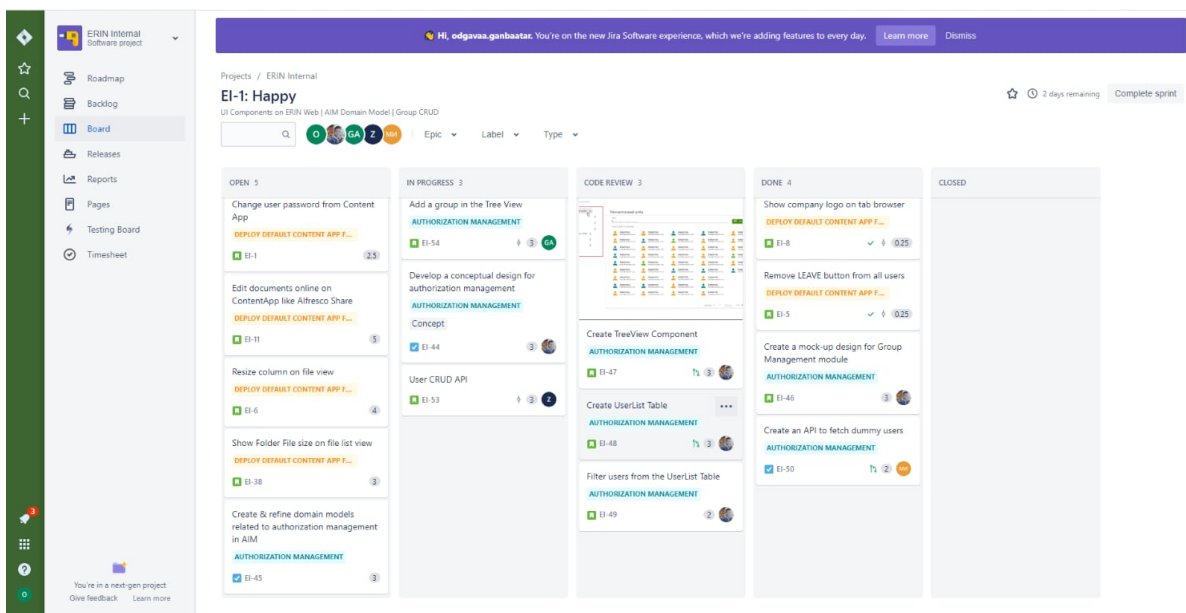
гэрээний болон тухайн Спринтийн хугацаанд дуусгаагүй бол түүний учир шалтгааныг тогтоон цаашид сайжруулах үйл явцад бүртгэн шийдвэрлэнэ.



Зураг 5.1: Scrum Process

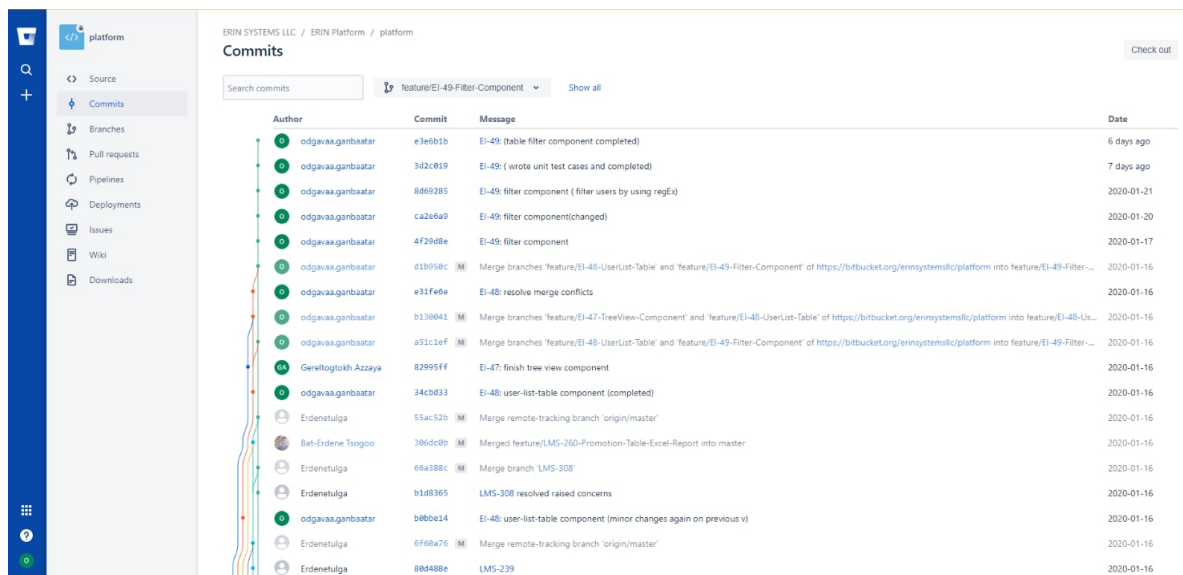
Програм хангамж хөгжүүлэлч компанийн хувьд зайлшгүй шаардлагатай програм хангамж бол **Version Control System** буюу **Git** юм. Манай компани Version Control System буюу ”Эх хувилбар хянах системийн” хувьд **BitBucket**-ийг ашигладаг. Мөн үйл ажиллагаандаа өдөр тутам хэрэглэдэг бусад програм хангамжуудаас дурьдвал:

- Jira - Jira бол төслийн үйл явцын удирдан явуулах хамгийн гол хэрэгсэл бөгөөд төслийн хүрээнд хийгдэх ажлуудыг ангилагдсан талбаруудад хувааж байрлуулснаар ажил тус бүрийг хэдий хугацаанд хэрхэн гүйцэтгэх, ямар үйл явцтай явж байгааг харуулдаг. Jira бол Скрам менежментийн гол хэрэглүүр юм.



Зураг 5.2: Jira board жишээ

- Confluence - Confluence бол (Content Collaboration Software) буюу байгууллагын баримт бичиг, үйл ажиллагаатай холбоотой хэрэгцээт баримт бичгүүдийг хадгалдаг хамтын үйл ажиллагааны програм хангамж юм.
- BitBucket - Version Control System буюу дээр дурьдсан эх хувилбар удирдах систем. VCS-г ашиглахын давуу тал нэг хамт олон нэг төсөл програмын хөгжүүлэлт дээр ажиллаж байгаа үед нэгэн зэрэг хөгжүүлэлт хийх болон хөгжүүлэлтийн явцад оруулсан өөрчлөлт бүрээ Git-н тусламжтай нэгтгэх боломжтой юм. VCS буюу Git-н хамгийн чухал командууд бол
 - git clone - төслийн эх кодыг бүхэлд нь локал дээр суулгах
 - commit - Өөрийн локал компьютер дээрээ өөрчлөлтөө хадгална.
 - push - VCS буюу Git- рүүгээ өөрчлөгдсөн хувилбараа хадгалж өгнө.
 - pull - бусад хөгжүүлэгчидийн оруулсан өөрчлөлтийг өөрийн локал компьютер дээрээ хадгалж авах



Зураг 5.3: Bitbucket VCS жишээ

Скрамын хувьд даалгавар таскуудыг дараа байдлаар ангилдаг.Үүнд:

- Task – refactor буюу кодон дээр сайжруулалт хийх болон back-end талын ажлууд байна.
- Story – UI буюу front-end талын хөгжүүлэлтийн ажлууд
- Bug – Software Bug гэдэг бол програм хангамжийн кодын алдаа доголдол юм. Bug нь major болон minor гэж ангилагддаг.
- TeamCity - TeamCity нь програмын чанарыг илтгэх тестүүдээ шалгах боломжтой програм хангамж буюу Continuous Integration (CI) –ийг хэрэгжүүлж буй програм хангамж гэж болно. Томоохон төсөл дээр алдаа доголдол , bug гарсныг Continuous Integration –ээр мэднэ. Жишээ нь Unit test, Module Test, UI test – үүдээр шалгана.

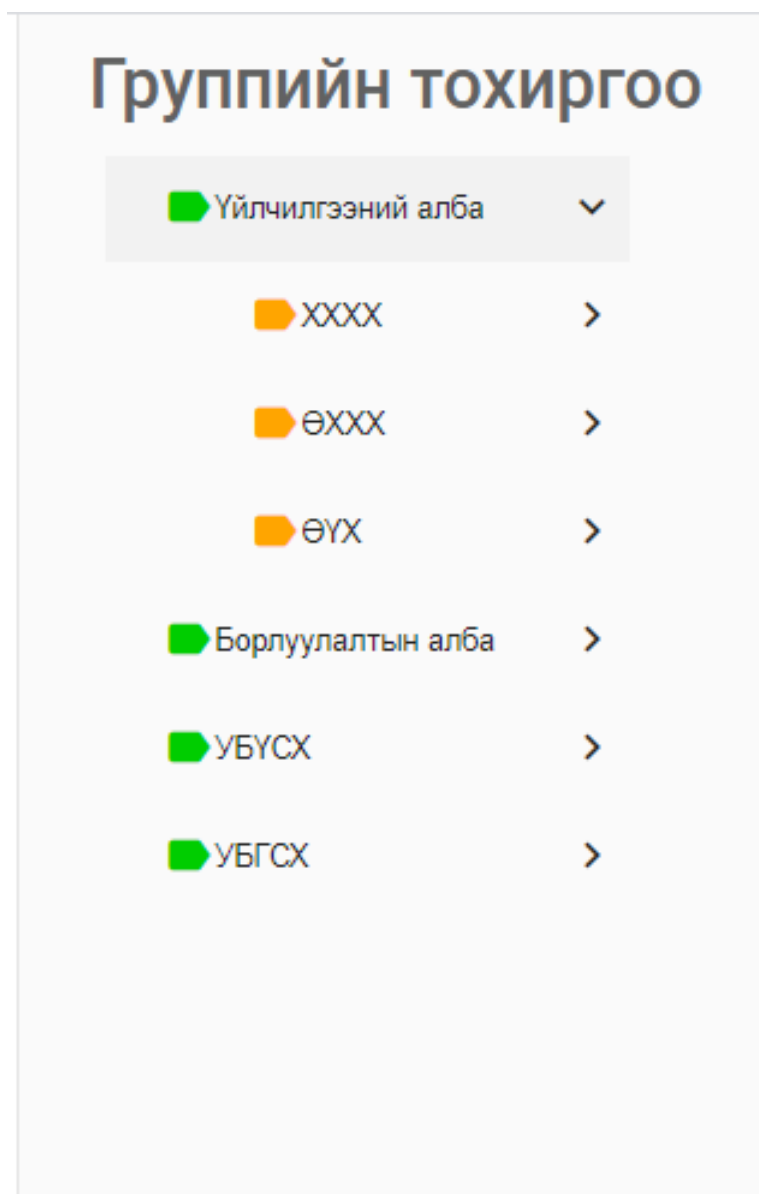
Эдгээр нь Atlassian Network компаний бүтээгдэхүүнүүд бөгөөд програм хангамжийн төсөл хэрэгжүүлэх, төсөл удирдан явуулахад зайлшгүй шаардлагатай програм хангамжийн бүтээгдэхүүнүүд юм.

5.2 Use-case бүрийн хэрэглэгчийн интерфэйс

5.2.1 Групптэй холбоотой хэрэглэгчийн интерфэйс

- Бүх группууд харагдах хэсэг бөгөөд дэд групп бүр нь доош мод бүтэцтэй харагдана.

Хэрэглэгчийн групп



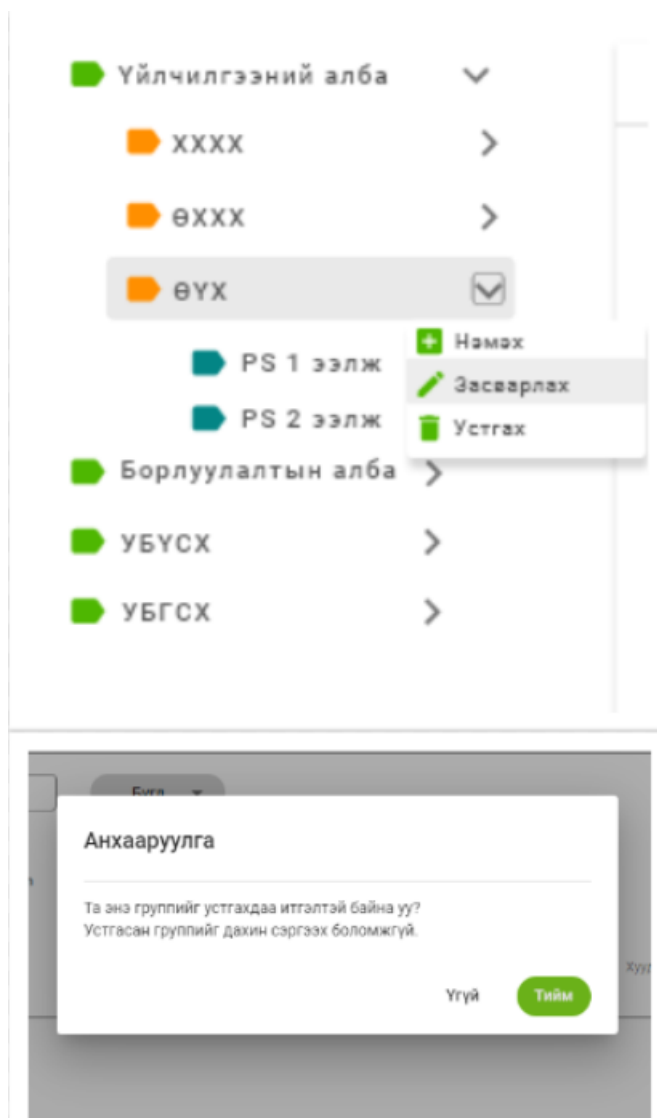
Зураг 5.4: Groups as tree view structure

- Группэд нэмэх хэсэг - Группын нэрийг оруулах хэсэгт нэрээ оруулаад Enter дарснаар шинэ групп үүснэ.



Зураг 5.5: Create group

- Групп устгах хэсэг - Групп устгах товч дарахад анхааруулга диалог гарч ирнэ.



Зураг 5.6: Delete group

5.2.2 Хэрэглэгчтэй холбоотой use - case бүрийн интерфэйс

- Группын тус бүрийн хэрэглэгчдийг харуулах хүснэгт

Групп доторх хэрэглэгчдийг харуулах үндсэн талбар нь мод бүтэцтэй группын талбартай

зэрэгцээ байрлана. Хэрэглэгчийн жагсаалтанд хэрэглэгчийн нэр болон имейл хаягийг харуулсан байна. Мөн доод хуудас гүйлгэх pagination хэсэгтэй байна.

Үйлчилгээний алба

Группд байгаа ажилчид

Хайхны нэрээр шүүх

Бүгд

Kevin kevin@gmail.com	Ken od@gmail.com	John john@gmail.com	Bold bold@gmail.com	Jason kevin@gmail.com	Anu anu@gmail.com
Bat kevin@gmail.com	Dorj bold@gmail.com	Onon bold@gmail.com	Ben bold@gmail.com	Saraa bold@gmail.com	Nina kevin@gmail.com
Nomin bold@gmail.com	Hanu kevin@gmail.com	Bob bold@gmail.com	Saraa bold@gmail.com	Nina kevin@gmail.com	Nomin bold@gmail.com
Hanu kevin@gmail.com	Bob bold@gmail.com	Kevin kevin@gmail.com	Ken od@gmail.com	John john@gmail.com	Bold bold@gmail.com
Jason kevin@gmail.com	Anu anu@gmail.com	Bat kevin@gmail.com	Dorj bold@gmail.com	Onon bold@gmail.com	Ben bold@gmail.com
Anu anu@gmail.com	Bat kevin@gmail.com	Dorj bold@gmail.com	Onon bold@gmail.com	Ben bold@gmail.com	Saraa bold@gmail.com
Nina kevin@gmail.com	Nomin kevin@gmail.com	Hanu kevin@gmail.com	Bob bold@gmail.com	Saraa bold@gmail.com	Nina kevin@gmail.com
Nomin bold@gmail.com	Hanu kevin@gmail.com	Bob bold@gmail.com	Saraa bold@gmail.com	Nina kevin@gmail.com	Nomin bold@gmail.com
Hanu kevin@gmail.com	Bob bold@gmail.com				

Хуудас 1 / 2 Хуудсанд 50 1 - 50 / 85 |< < > >|

Зураг 5.7: User-list table

- Группэд хэрэглэгч нэмэх хэсэг

Хэрэглэгч нэмэх группээ сонгоод нэмэх товчийг дарахад системд бүртгэлтэй хэрэглэгчийг сонгосон группэд нэмнэ.

PS 1 ээлж

Хайх

Хайх ажилчны нэрийг оруулна уу.

Группд байгаа ажилчид

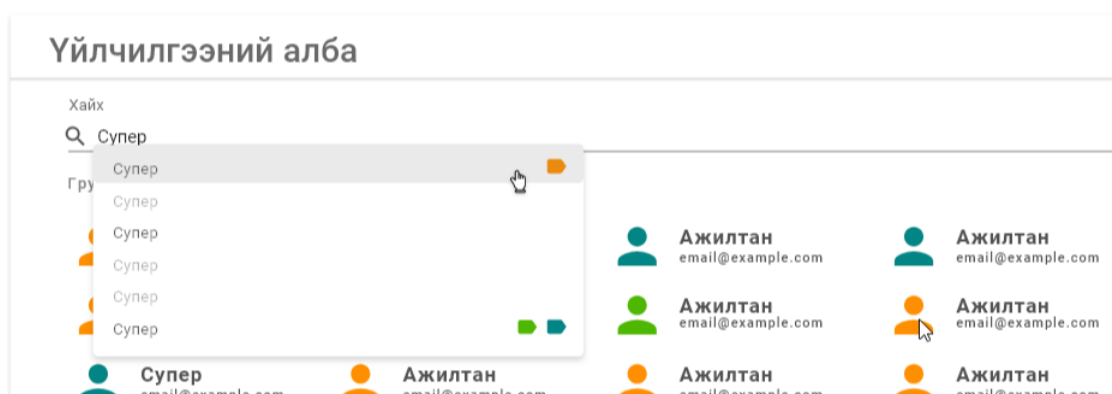
НЭМЭХ

Зураг 5.8: Add user to group

- Группээс хэрэглэгч хайх хэсэг

Хэрэглэгч нэмэх группээ сонгоод нэмэх товчийг дарахад системд бүртгэлтэй хэрэглэгчийг

сонгосон группэд нэмнэ.



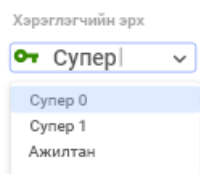
Зураг 5.9: Search user

5.2.3 Хэрэглэгчийн рольтой холбоотой интерфэйс

- Хэрэглэгчид роль оноох



Зураг 5.10: User role



Зураг 5.11: User role

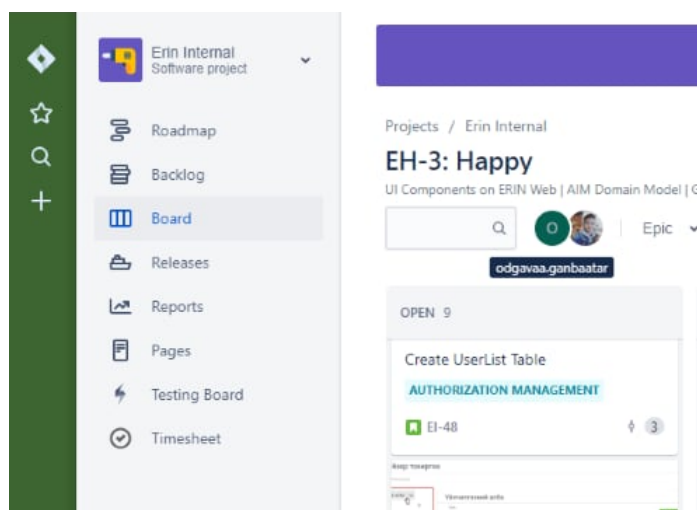
5.3 Хэрэгжүүлсэн байдал

Миний хувьд дадлагын хугацаанд LMS- Групп модуль дээр ажиллахдаа 3 төрлийн стори буюу хэрэглэгчийн интерфэйс гаргах даалгавар авч ажилласан. Эдгээр нь групп доторх хэрэглэгчдийг харуулах, хэрэглэгчийн талбараас хэрэглэгч шүүх болон хэрэглэгчдийг эрхээр нь ангилан харуулах даалгаваруудын front-end талын хөгжүүлэлт дээр ажилласан. Хамгийн сүүлд эдгээр даалгаваруудыг гүйцэтгэж дуусаад Sprint Meeting дээр танилцуулж дутагдалтай орхигдсон зүйлүүдийг засах, нэмэлт өөрчлөлт оруулах нэгтгэсэн нэг стори авч ажилласан. Хөгжүүлэлт хийсэн стори даалгавар дээрээ ажиллагааг шалгахын тулд Unit Test бичсэн.

5.3.1 Анхны авч ажилласан даалгавар (Create User list Table story)

Энэхүү даалгавар нь Админ болон Супервайзор сонгосон группын хэрэглэгчийн мэдээлэл, тухайн групп дахь хэрэглэгчийн тоог харах боломжийг олгодог.

- Jira board дээр энэхүү даалгаварын сториг үүсгэсэн байдал.



Зураг 5.12: Create User Table story







Front-end хөгжүүлэлт Angular Framework дээр хийгддэг учир энэхүү сториг гүйцэтгэхдээ Angular Material.io санг түлхүү ашигласан. Хэрэглэгчдийг хүснэгтэн дотор жагсаалт

байлдаар харуулахын тулд Mat-Card хэмээх материалын компонентийг ашигласан.

Стори, таск бүр өөр өөрсдийн DoD(Definition of Done) буюу энэ компонент юу юуг хийж чаддаг байх вэ гэдгийг ажлын талбар дээр оруулахдаа тодорхойлсон байдаг.

Группын хэрэглэгчдийг харуулах энэхүү даалгаварын хувьд DoD бол:

Create UserList Table



Description

As Admin, I want to see the list of users in a table so that I can see who is in which group.

DoD:

- Each user will contain an icon, display name and email address. Data can be static/fixed
- The user icons are not required to be in color (just yet)
- The table must have a pagination

Зураг 5.13: DoD of UserList Table Implementation

- Гол хийх зүйл нь тухайн групп ямар ямар хэрэглэгчид байгааг харуулах бөгөөд user-icon нь группын лабелийн өнгөөс хамаараад өөр өөр байх боловч энэ даалгаварын хүрээнд хийгдэхгүй. Энэ даалгавар групп модулийн хамгийн эхний даалгаваруудын нэг учир анхлан хөгжүүлэлт хийгдэх явцад production server дээр суулгаагүй байгаа хувилбар юм.









Хэрэгжүүлэлтийн явц:

Хэрэгжүүлэлтийн явц:

- Бүх хэрэглэгчдээс шүүх

Үйлчилгээний алба

Группд байгаа ажилчид









<input type="text" value="ч"/>	Бүгд ▾				
 Kevin kevin@gmail.com	 Ken od@gmail.com	 Kevin kevin@gmail.com	 Ken od@gmail.com	 Kevin kevin@gmail.com	 Ken od@gmail.com
 Kevin kevin@gmail.com	 Ken od@gmail.com				

Хуудас 1 / 1 Хуудсанд 50 ▾ 1 - 8 / 8 |< < > >|

Зураг 5.16: Filter users from the table

- Ажилчидаас шүүх

Үйлчилгээний алба

<input type="text" value="н"/>	Ажилтан ▾				
 Nina kevin@gmail.com	 Nina kevin@gmail.com	 Nina kevin@gmail.com	 Nina kevin@gmail.com	 Nina kevin@gmail.com	 Nina kevin@gmail.com
 Nina kevin@gmail.com	 Nina kevin@gmail.com				

Хуудас 1 / 1 Хуудсанд 50 ▾ 1 - 8 / 8 |< < > >|

Зураг 5.17: Filter users from the table










- Админуудаас шүүх

Үйлчилгээний алба

Группд байгаа ажилчид

Ажилчны нэрээр шүүх

Админ

 John john@gmail.com	 Onon bold@gmail.com	 John john@gmail.com	 Onon bold@gmail.com	 Onon bold@gmail.com	 John john@gmail.com
 Onon bold@gmail.com	 John john@gmail.com	 Onon bold@gmail.com			

Хуудас 1 / 1 Хуудсанд 50 1 - 9 / 9 |< < > >|

Зураг 5.18: Filter users from the table









- Супервайзоруудаас шүүх

Үйлчилгээний алба

Группд байгаа ажилчид

s

Супер

 Saraa bold@gmail.com	 Saraa bold@gmail.com	 Saraa bold@gmail.com	 Saraa bold@gmail.com	 Saraa bold@gmail.com	 Saraa bold@gmail.com
 Saraa bold@gmail.com	 Saraa bold@gmail.com				

Хуудас 1 / 1 Хуудсанд 50 1 - 8 / 8 |< < > >|

Зураг 5.19: Filter users from the table

5.3.3 Хөгжүүлэлтийн явцад Unit Test бичих үе шат

Хөгжүүлэгч өөрийн бичсэн кодын үйл ажиллагааг шалгахын тулд заавал хамгийн анхан шатны Unit Test бичих хэрэгтэй. Програм хангамжийн тест гэдэг бол тодорхой шалгуурын биелэлтээр програмын чанар болон шаардлагын биелэлтийг хянадаг процесс юм. Тестийг автоматаар болон гараар ажиллуулдаг. Мөн тест нь тухайн програм хангамжийн чанарыг тодорхойлох гол үзүүлэлт болдог. Бидний бичсэн код үнэн зөв ажиллаж байгааг тестгүйгээр үнэн зөв ажиллах эсэх нь тодорхойгүй байдаг.. Иймд бичсэн код үнэн зөв, алдаа доголдолгүй ажиллаж байна уу ? гэдгийг хамгийн доод түвшинд Unit test – ээр шалгана. Тухайн класс болгон Unit test байх хэрэгтэй бөгөөд Positive unit test , Negative unit test – ийг бичсэнээр програмын хөгжүүлэлтийн үед алдаа бага гарах , гарахаас эрсдэлээс сэргийлээд зогсохгүй програмын чанарыг тодорхойлно. Front -end талын хөгжүүлэлтэд заавал бүх компонент бүр дээр тест бичих шаардлагагүй. Харин логик үйлдэлүүд хийгдэж байгаа компонентүүд дээр заавал Unit test бичнэ.




- Unit test – Нэг функц дээр бичигдэх хамгийн доод түвшний харьцангуй амархан хийгдэх тест ч үр дүнтэй unit test – үүд бичсэнээр цаашдын тестлэх процесс харьцангуй амар байх болно.
- Module Test – Хоёроос гурван класс, функцийн хооронд хийгдэх тест юм.
- UI Test – Selenium системийг ашиглаж, тухайн систем, програм хангамж дээр хүн ажиллаж буй мэтээр тестлэж үзэхийг хэлнэ. Selenium нь нэг төрлийн web browser–ийн автоматжуулалт юм.

6. ҮР ДҮН

Эдгээр сторины хөгжүүлэлт дуусахад тус бүрээр code review тус бүр дээр хийгдэж дараа нь UI - тестээр шалгаж ямар нэг алдаа гараагүй бол сториг хаана.

Projects / ERIN Internal / Authorization Management / EI-48

Create UserList Table


Description
As Admin, I want to see the list of users in a table so that I can see who is in which group.

DoD:

- Each user will contain an icon, display name and email address. Data can be static/fixed
- The user icons are not required to be in color (just yet)
- The table must have a pagination

Closed ✓ Done

Product name
Platform

Assignee
 odgavaa.ganbaatar






Labels
None

Зураг 6.1: Closed story 1

•

Projects / ERIN Internal / Authorization Management / EI-49

Filter users from the UserList Table


Description
As Admin, I want to filter a user from UserList table so that I can easily find a user.
This issue is blocked by EI-48.

DoD:

- Create an input field for filter
- Typing a value should dynamically show the corresponding users. (For example, typing "B" will display all the users whose names start with the letter "B")

Closed ✓ Done

Product name
Platform

Assignee
 odgavaa.ganbaatar

Labels
None

Sprint

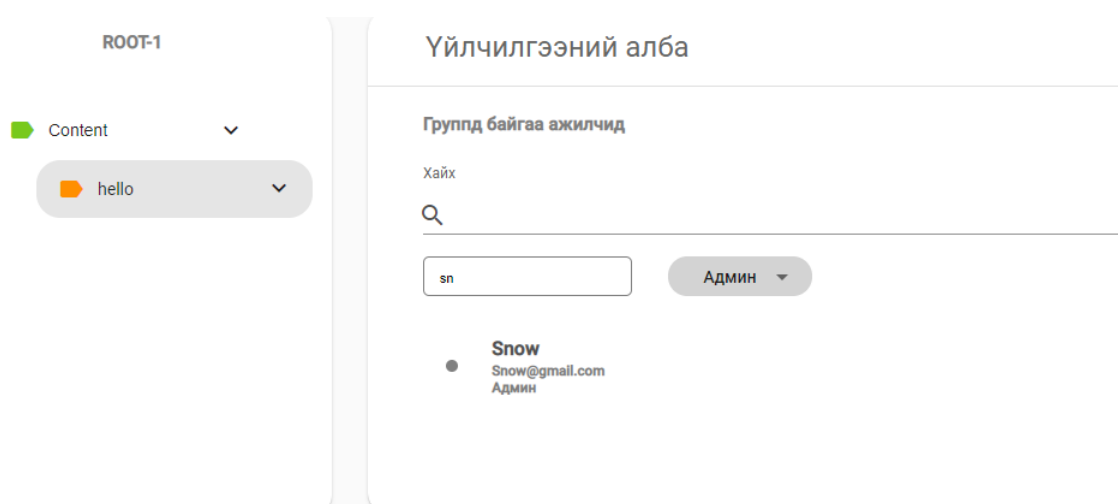
Зураг 6.2: Closed story 2

•

6.1 Сүүлийн үр дүн буюу production server дээр ажиллуулсан байдал

Өмнөх даалгавараар эдгээр сторинуудын front-end хөгжүүлэлтийг хийж гүйцэтгэсэн бол түүн дээрээ нэмэлт өөрчлөлт оруулах болон back-end тэй холбож production server дээр ажиллуулах даалгаварыг нэмэлт сторигеор гүйцэтгэсэн. Мөн хэрэглэгч нэмэх болон хайх хэсгийн хөгжүүлэлт дээр ажилласан. Өгөгдлийн хувьд жишээ өгөгдлүүд оруулсан.

6.1.1 Хэрэглэгчдийг харуулах, шүүх



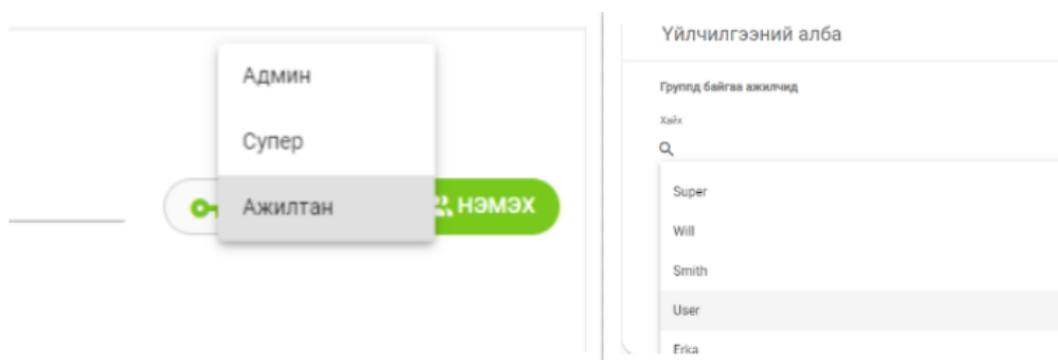
Зураг 6.3: Үр дүн 1



Зураг 6.4: Үр дүн 2

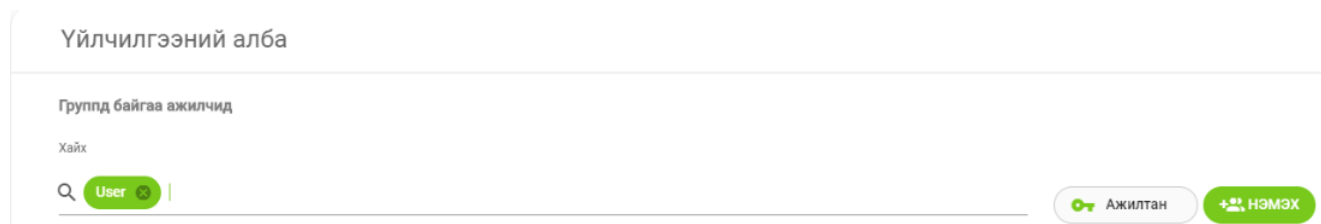
6.1.2 Хэрэглэгч хайх нэмэх даалгаварын хэрэгжүүлэлтийн үр дүн

- Хэрэглэгч хайх - Энэ нь хүснэгтээс хэрэглэгч хайхаас ялгаатай тал нь нийт систем дээрх бүртгэлтэй группэд харьяалагдаагүй хэрэглэгчдийг роль тус бүрээс сонгон хайж, группд нэмнэ.



Зураг 6.5: Хэрэглэгч хайх - үр дүн

- Хэрэглэгч нэмэх - Хайлтаар гарч ирсэн хэрэглэгчийн нэмэх



Зураг 6.6: Хэрэглэгч нэмэх - үр дүн

Дүгнэлт

Миний бие үйлдвэрлэлийн дадлагыг Эрин Системс ХХК-д 2020.01.03 - 2020.01.25 ний хооронд 3 долоо хоногийн хугацаанд хийж цаашид үргэлжлүүлэн дадлагажигчаар ажиллаж байгаа билээ. Дадлагын эхний өдрүүдэд байгууллага, хамт олон, үйл ажиллагаатай танилцах, ашигладаг технологиудын талаар суралцах болон хэрэгжүүлж буй төслүүдтэй танилцаж Скрамын үйл ажиллагааны хүрээнд Daily Stand-up meeting болон Sprint Change meeting- д оролцон дотоод үйл ажиллагаатай танилцсан. Миний хувьд front-end талын хөгжүүлэлт дээр ажиллах илүү сонирхолтой байсан учир удирдагчийн зөвлөмжийн дагуу TypeScript болон Angular Framework-г суралцаж эхэлсэн. Анх дадлага хийж эхлэхдээ Angular-н талаар ямар ч мэдлэггүй байсан ба Angular-н үндсэн ойлголтууд болох модуль, компонент болон service- тэй ажиллах, Unit test бичих талаар сурсан. Тест бол програм хангамжийн чанарыг тодорхойлох томоохон хүчин зүйл бөгөөд Unit test бол хамгийн бага нэгж дээр шалгалт хийж байгаа хялбар юм шиг боловч хамгийн хариуцлагатай тест гэдгийг ойлгож авсан. Цаашлан Scalable Angular Architecture- н талаар илүү дэлгэрэнгүй судалж хөгжүүлтийн явцад Scalable Architecture-н дагуу ажилласан. Мөн дан ганц технологийн талаар суралцаад зогсохгүй програм хангамж хөгжүүлэлтийн чухал ойлголт болох Domain Driven Design, Solid програмчлалын талаар судалж мэдсэн билээ. Хөгжүүлэлт хийж байхдаа Clean code болон Refactoring -н хэрэгжүүлж ажилласан. Энэ нь бусад хөгжүүлэгчид миний бичсэн кодыг уншихад хялбар, цэгцтэй болгож өгдөг. Clean code бичихийн тулд IntelliJ IDEA дээр ажилладаг учир SonarLint - ийг ашигладаг байсан. Код бичсэний дараа SonarLint-ийг ажиллуулснаар кодын зүй зохисгүй , засаж сайжруулах ойлгомжгүй хэсгийг олж өгдөг. Энэ нь нэг хөгжүүлэгчийн бичсэн кодыг нөгөө хөгжүүлэгч code review хийхэд маш хэрэгтэй гэдгийг ойлгож авсан. Мөн програм хангамжийн төслийн менежмент хичээлээр судалж мэдсэн Agile(хурдацтай хөгжүүлэлт) болон Скрам менежмент байгууллагын хүрээнд бодит төсөл дээр хэрхэн явагддаг талаар практик дээрээс суралцаж

танхимын сургалтаар олж авсан мэдлэгээ илүү бататгаж чадсан. Миний хувьд манай компанийн чухал төслүүдийн нэг болох Сургалт Менежментийн Системийн нэг хэсэг дээр гар бие оролцож өөрийн хувь нэмрээ оруулахын зэрэгцээ олон шинэ зүйлүүд суралцсандаа баяртай байна. Үйлдвэрлэлийн дадлага хийснээр оюутан өөрийн мэргэжлийн онцлог шинжийг илүү таньж мэдэх, програм хангамжийн инженер хүн ажлын байранд гарахдаа юу юуг анхаарах, цаашид юу юуг суралцах, ажлын байранд гараад ажил хэрхэн явагддаг гэх мэт хувь хүний өөрөө өөрийгөө ямар түвшинд явж байгаа, би юу юу чаддаг юм гэдгээ таньж мэдэх нь чухал юм байна гэж дүгнэлээ.

Хавсралт

6.2 Main component-н кодын хэсэг

Групп доторх хэрэглэгчдийг харуулах, ролиор ангилах болон нэрээр шүүх даалгаварыг хэрэгжүүлсэн Main code буюу user-list-view.ts файл. Энэхүү main component дээр user-search-field болон user-card-filter дэд компонентүүдийг дуудан ажиллуулж байгаа. Тэдгээр компонент дээр логих үйлдэл хийгдээгүй бөгөөд parent component- руугаа data binding хийнэ.

```
1 import {AfterViewInit, ChangeDetectorRef, Component, EventEmitter,
    Input, OnChanges, Output, SimpleChanges, ViewChild} from '@angular/
    core';
2 import {AddUserModel, DetailedUserInfo, Role, UserInfo} from '../..//
    models/group-management.model';
3 import {MatTableDataSource} from '@angular/material/table';
4 import {MatPaginator} from '@angular/material/paginator';
5 import {Observable} from 'rxjs';
6 import {UserSearchFieldComponent} from '../search/user-search-field.
    component';
7 import {DropdownComponent} from '../..//common/components/dropdown/
    dropdown.component';
8
9 @Component({
10   selector: 'user-card-view',
11   template: `
12     <mat-card class="card">
```

```

13     <mat-card-header><h1 class="mat-h1">{{nameOfGroup}}</h1></mat-
        card-header>
14 </mat-divider></mat-divider>
15 <mat-card-content>
16     <h4 class="mat-h4">{{title}}</h4>
17     <user-search-field #searchField
18         [allUsers]="allUsers"
19         [roles]="roles"
20         (addUsers)="onUserAdded($event)">
21 </user-search-field>
22 <user-card-filter (change)="receivedValue($event)"></user-card-
        filter>
23 <dropdown
24     *ngIf="roleNames.length > 2"
25     #dropdown [values]="roleNames"
26     [style]=" 'role-filter ' "
27     (selectionChange)="onChangeRole($event)"></dropdown>
28 <div class="user-card"
29     cdkDropList
30     [cdkDropListData]="users">
31 <div cdkDrag
32     *ngFor="let user of users | async"
33     [ngClass]="user.moreOption ? OPTION_GRAY : OPTION_WHITE"
34     class="each-user"
35     (mouseenter)="mouseenter(user)"
36     (mouseleave)="mouseleave(user)">
37 <div class="card-icon"></div>
38 <div class="card-detail">

```

```

39         <h3 class="mat-h3">{{user.username}}</h3>
40         <h5 class="mat-h5">{{user.email}}</h5>
41         <h5 class="mat-h5">{{user.role}}</h5>
42     </div>
43     <button mat-icon-button [matMenuTriggerFor]="addOption"
44         class="user-more-option">
45         <mat-icon>
46             more_vert
47         </mat-icon>
48         <mat-menu #addOption='matMenu'>
49             <button mat-menu-item (click)="onUserRemoved($event,
50                 user)">
51                 <mat-icon style="line-height: 0.8; margin-right: 10px
52                     ">remove_circle</mat-icon>
53                 <span    ></span>
54             </button>
55         </mat-menu>
56     </button>
57 </div>
58 </div>
59 </mat-card-content>
60 <erin-pagination [dataSource]="dataSource" [defaultPageSize
61     ]="50"></erin-pagination>
62 </mat-card>
63 ` ,
64 styleUrls: ['./user-card-view.component.scss']
65 })

```

```

62 export class UserCardViewComponent implements OnChanges, AfterViewInit
    {
63     @Input() allUsers: DetailedUserInfo[];
64     @Input() data: UserInfo[];
65     @Input() roles: Role[];
66
67     @Output() addUser: EventEmitter<AddUserModel> = new EventEmitter<
        AddUserModel>();
68     @Output() removeUser: EventEmitter<UserInfo> = new EventEmitter<
        UserInfo>();
69
70     @ViewChild(MatPaginator) paginator: MatPaginator;
71     @ViewChild('searchField') searchField: UserSearchFieldComponent;
72     @ViewChild('dropdown') dropdown: DropdownComponent;
73
74     nameOfGroup =      '    ';
75     title =      '    ';
76     OPTION_GRAY = 'optionGray';
77     OPTION_WHITE = 'optionWhite';
78     dataSource = new MatTableDataSource<UserInfo>();
79     users: Observable<any>;
80     searchValue: string;
81     selectedRole: string;
82
83     userInfos: UserInfo[];
84     originalUserData: UserInfo[];
85     categorizedData: UserInfo[];
86

```



```

87     roleNames: string[] = [];
88
89     constructor(private cd: ChangeDetectorRef) {
90     }
91
92     ngOnChanges(changes: SimpleChanges): void {
93         for (const prop in changes) {
94             if (prop === 'data' && this.data !== null) {
95                 this.userInfos = this.data;
96                 this.originalUserData = this.data;
97                 this.updateTable();
98             }
99             if (prop === 'roles' && this.roles !== null) {
100                 this.roleNames = this.roles.map(role => role.roleName);
101                 this.roleNames.unshift('');
102             }
103         }
104     }
105
106     ngAfterViewInit(): void {
107         this.cd.detectChanges();
108     }
109
110     onChangeRole(value: string) {
111         this.selectedRole = value;
112         if (this.selectedRole === '') {
113             this.userInfos = this.originalUserData;
114             this.updateTable();

```

```

115         return;
116     }
117     this.userInfos = this.originalUserData.filter(user => user.role ===
        this.selectedRole);
118     this.categorizedData = this.userInfos;
119     this.updateTable();
120 }
121
122 onUserAdded(data: AddUserModel) {
123     this.addUser.emit(data);
124     this.dropdown.reset();
125 }
126
127 onUserRemoved(event, user: UserInfo) {
128     this.removeUser.emit(user);
129     this.dropdown.reset();
130 }
131
132 receivedValue(searchValue: string) {
133     this.searchValue = searchValue;
134     if (searchValue === null) {
135         return;
136     }
137     if (this.searchValue !== undefined && this.searchValue.trim().
        length === 0) {
138         if (this.selectedRole === undefined) {
139             this.userInfos = this.originalUserData;
140             this.updateTable();

```

```

141         return;
142     }
143     if (this.selectedRole === '') {
144         this.userInfos = this.originalUserData;
145     } else {
146         this.userInfos = this.categorizedData;
147     }
148     this.updateTable();
149 }
150 this.userInfos = this.originalUserData.filter(user => new RegExp(`^
    ${this.searchValue}`, 'gi').test(user.username) ||
151     user.username.toLocaleLowerCase().includes(this.searchValue.
        toLocaleLowerCase())));
152 this.updateTable();
153 }
154
155 updateTable() {
156     this.dataSource = new MatTableDataSource<UserInfo>(this.userInfos);
157     this.users = this.dataSource.connect();
158     this.cd.detectChanges();
159 }
160
161 reset(): void {
162     this.userInfos = this.data;
163     this.originalUserData = this.data;
164     this.updateTable();
165     this.searchField.reset();
166 }

```

```

167
168     mouseEnter(user) {
169         user.moreOption = true;
170     }
171
172     mouseLeave(user) {
173         setTimeout(() => {
174             user.moreOption = false;
175         }, 100);
176     }
177 }

```

6.3 Main component дээр Unit Test бичсэн байдал

```

1 import {async, ComponentFixture, fakeAsync, TestBed, tick} from '
    @angular/core/testing';
2 import {UserCardViewComponent} from './user-card-view.component';
3 import {By} from '@angular/platform-browser';
4 import {DetailedUserInfo, Role, UserInfo} from '../models/group-
    management.model';
5 import {UserCardFilter} from './filter/user-card-filter.component';
6 import {MaterialModule} from '../material.module';
7 import {FormsModule, ReactiveFormsModule} from '@angular/forms';
8 import {UserSearchFieldComponent} from './search/user-search-field.
    component';
9 import {ErinCommonModule} from '../common/common.module';
10 import {SimpleChange} from '@angular/core';
11 import {DragDropModule} from "@angular/cdk/drag-drop";

```

```

12
13 describe('UserCardViewComponent', () => {
14     let component: UserCardViewComponent;
15     let fixture: ComponentFixture<UserCardViewComponent>;
16
17     const USER_INFOS: UserInfo[] = [
18         {username: 'Kevin', email: 'kevin@gmail.com', role:    '',
19           phoneNumber: '123123'},
20         {username: 'Ken', email: 'od@gmail.com', role:      '', phoneNumber:
21           '123123'},
22         {username: 'John', email: 'john@gmail.com', role:   '', phoneNumber
23           : '123123'},
24         {username: 'Bold', email: 'bold@gmail.com', role:   '', phoneNumber
25           : '123123'},
26         {username: 'Jason', email: 'kevin@gmail.com', role:  '',
27           phoneNumber: '123123'},
28     ];
29
30     const DETAILED_USER_INFOS: DetailedUserInfo[] = [
31         {username: 'Kevin', email: 'kevin@gmail.com', phoneNumber:
32           '123123',
33         membership: {userId: 'Kevin', roleId: '', groupId: 'Group',
34           membershipId: '123'}}},
35         {username: 'Ken', email: 'od@gmail.com', phoneNumber: '123123',
36           membership: null},
37         {username: 'John', email: 'john@gmail.com', phoneNumber: '123123',
38           membership: null},
39         {username: 'Bold', email: 'bold@gmail.com', phoneNumber: '123123',

```

```

        membership: null},
31 {username: 'Jason', email: 'kevin@gmail.com', phoneNumber:
    '123123', membership: null},
32 {username: 'Anu', email: 'anu@gmail.com', phoneNumber: '123123',
    membership: null},
33 {username: 'Bat', email: 'kevin@gmail.com', phoneNumber: '123123',
    membership: null},
34 {username: 'Dorj', email: 'bold@gmail.com', phoneNumber: '123123',
    membership: null},
35 {username: 'Onon', email: 'bold@gmail.com', phoneNumber: '123123',
    membership: null},
36 {username: 'Ben', email: 'bold@gmail.com', phoneNumber: '123123',
    membership: null},
37 {username: 'Saraa', email: 'bold@gmail.com', phoneNumber: '123123',
    membership: null},
38 {username: 'Nina', email: 'kevin@gmail.com', phoneNumber: '123123',
    membership: null},
39 {username: 'Nomin', email: 'bold@gmail.com', phoneNumber: '123123',
    membership: null},
40 {username: 'Hanu', email: 'kevin@gmail.com', phoneNumber: '123123',
    membership: null},
41 {username: 'Bob', email: 'bold@gmail.com', phoneNumber: '123123',
    membership: null}
42 ];
43
44 const ROLES: Role[] = [
45     {roleId: '-admin-', roleName: ''},
46     {roleId: '-super-', roleName: ''},

```

```

47     {roleId: '-employee-', roleName: ''}
48 ];
49
50 beforeEach(async(() => {
51     TestBed.configureTestingModule({
52         imports: [
53             MaterialModule,
54             ErinCommonModule,
55             FormsModule,
56             ReactiveFormsModule,
57             DragDropModule
58         ],
59         declarations: [UserCardViewComponent, UserCardFilter,
60             UserSearchFieldComponent]
61     })
62     .compileComponents();
63
64     beforeEach(() => {
65         fixture = TestBed.createComponent(UserCardViewComponent);
66         component = fixture.componentInstance;
67         component.allUsers = DETAILED_USER_INFOS;
68         component.data = USER_INFOS;
69         component.roles = ROLES;
70         component.ngOnChanges({
71             data: new SimpleChange(null, component.data, true),
72             roles: new SimpleChange(null, component.roles, true)
73         });

```

```

74     });
75
76     afterEach(() => {
77         fixture.destroy();
78     });
79
80     it('should create', () => {
81         expect(component).toBeTruthy();
82     });
83
84     it('should have users', () => {
85         const user = fixture.debugElement.queryAll(By.css('.each-user'));
86         expect(user.length).toEqual(5);
87     });
88
89     it('user name should be equal', () => {
90         const username = fixture.debugElement.queryAll((By.css('.mat-h3')))
91             ;
92         expect(username[0].nativeElement.innerText).toEqual('Kevin');
93     });
94
95     it('each user should have icon', () => {
96         const icon = fixture.debugElement.queryAll((By.css('.card-icon')));
97         expect(icon.length).toEqual(5);
98     });
99
100    it('after filter user should be equal', () => {
101        const filter = fixture.debugElement.query((By.css('user-card-filter
102            '))));
103        filter.triggerEventHandler('change', 'k');

```



```

100     const user = fixture.debugElement.queryAll((By.css('.mat-h3')));
101     expect(user.length).toEqual(2);
102     expect(user[1].nativeElement.innerText).toEqual('Ken');
103   });
104
105   it('after filter there will be no user', () => {
106     const filter = fixture.debugElement.query((By.css('user-card-filter
107       '))));
108
109     filter.triggerEventHandler('change', 'q');
110
111     const user = fixture.debugElement.queryAll((By.css('.mat-h3')));
112     expect(user.length).toEqual(0);
113   });
114
115   it('after filter there will be all user', () => {
116     const filter = fixture.debugElement.query((By.css('user-card-filter
117       '))));
118
119     filter.triggerEventHandler('change', '');
120
121     fixture.detectChanges();
122
123     const user = fixture.debugElement.queryAll((By.css('.mat-h3')));
124     expect(user.length).toEqual(5);
125   });
126
127   it('only staffs should be displayed', () => {
128     component.onChangeRole('');
129     fixture.detectChanges();

```

```

126     const user = fixture.debugElement.queryAll(By.css('.mat-h3'));
127     expect(user.length).toEqual(1);
128   });
129   it('all users should be displayed', () => {
130     component.onChangeRole ('');
131
132     fixture.detectChanges();
133
134     const user = fixture.debugElement.queryAll(By.css('.mat-h3'));
135     expect(user.length).toEqual(5);
136   });
137   it('should display users in the search field', () => {
138     generateInput('B');
139
140     const matOptions = fixture.debugElement.queryAll(By.css('.mat-
141       option'));
142     expect(matOptions.length).toEqual(4);
143   });
144   it('should disable users already in the view', () => {
145     generateInput('Kevin');
146
147     const matOptions = fixture.debugElement.queryAll(By.css('.mat-
148       option'));
149     expect(matOptions.length).toEqual(1);
150     expect(matOptions[0].nativeElement.classList.contains('mat-option-
151       disabled')).toBeTruthy();
152   });
153   it('should display a user as a chip', () => {

```

```

151     generateInput('Bob');
152     const option = fixture.debugElement.query(By.css('.mat-option'));
153     option.nativeElement.click();
154     fixture.detectChanges();
155     const matChip = fixture.debugElement.query(By.css('.mat-chip'));
156     expect(matChip).toBeDefined();
157   });
158
159   function generateInput(value: string) {
160     let inputElement: HTMLInputElement;
161     inputElement = fixture.nativeElement.querySelector('input');
162     inputElement.focus();
163     inputElement.dispatchEvent(new Event('focusing'));
164     inputElement.value = value;
165     inputElement.dispatchEvent(new Event('input'));
166
167     fixture.detectChanges();
168   }
169 });

```