

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

Баасансүрэн Мөнх-Ундрал

**ДАДЛАГЫН АЖЛЫН ТАЙЛАН
(Internship report)**

Програм Хангамж (D061302)
Бакалавр

Улаанбаатар

2020 оны 03 сар

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

ДАДЛАГЫН АЖЛЫН ТАЙЛАН
(Internship report)

Програм Хангамж (D061302)
Бакалавр

Удирдагч: _____ Б. Хуягбаатар

Гүйцэтгэсэн: _____ Б.Мөнх-Ундрал (17B1NUM1824)

Улаанбаатар

2020 оны 03 сар

Зохиогчийн баталгаа

Миний бие Баасансүрэн Мөнх-Ундрал ”ДАДЛАГЫН АЖЛЫН ТАЙЛАН” сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- 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.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

УДИРТГАЛ.....	1
1. СИСТЕМИЙН ТАНИЛЦУУЛГА	2
1.1 Компанийн танилцуулга	2
1.2 Системийн танилцуулга	2
2. ИЖИЛ ТӨСТЭЙ СИСТЕМИЙН СУДАЛГАА	3
2.1 Ижил төстэй систем	3
2.2 Системүүдийн шинж чанарыг харьцуулах	3
2.3 Судалгаа	4
3. СИСТЕМИЙН ШААРДЛАГА	6
3.1 Функциональ шаардлага	6
3.2 Функциональ бус шаардлага	7
3.3 Use-case диаграм	9
4. СИСТЕМИЙН ШИНЖИЛГЭЭ ЗОХИОМЖ	12
4.1 Database diagram	12
4.2 Class diagram	21
4.3 Sequence diagram	22
5. ХЭРЭГЖҮҮЛЭЛТ	24
5.1 Системийг бий болгоход ашигладаг технологи	25
6. ҮР ДҮН	26
7. КОДЫН ХЭРЭГЖҮҮЛЭЛТ	33
ДҮГНЭЛТ	34
НОМ ЗҮЙ	34
ХАВСРАЛТ	37
А. ӨГӨГДӨЛ ХАДГАЛАХ	38

ЗУРГИЙН ЖАГСААЛТ

2.1	Фэйсбүүк дээрх хэрэглэгчдийн одоогийн байдал	4
3.1	Use Case	10
4.1	ER Diagram	12
4.2	Database Diagram	13
4.3	Class Diagram	21
4.4	Хэрэглэгч бүртгүүлэх дарааллын диаграм	22
4.5	Ном нэмэх дарааллын диаграм	23
6.1	Энэхүү системийг хийхдээ Vue.js , vuetify ашиглан front-end- ийг хийсэн	26
6.2	Loopback4 дээр project үүсгэн backend-ийг хийж гүйцэтгэсэн	27
6.3	get, post - оор дамжуулан ном нэмэх харах үйл ажиллагаа	27
6.4	Хэрэглэгч бүртгүүлэх хэсэг	28
6.5	Хэрэглэгч нэвтрэх хэсэг	29
6.6	Хэрэглэгч нэвтэрч орсоны дараах үндсэн цонх	30
6.7	Ном нэмэх хэсэг	30
6.8	Book Exchange хуудас	31
6.9	Хайлтын хэсэг	32
7.1	Ном бүртгүүлэх Back-end талын код	33
7.2	Ном бүртгүүлэх Front-End талын код	34
7.3	Хэрэглэгчийн бүртгэл Back-end	35
7.4	Хэрэглэгчийн бүртгэл Front-end	36
7.5	User model	36
A.1	Base дээрх бүртгүүлсэн хэрэглэгчийн мэдээлэл хадгалсан байдал	38
A.2	Base дээрх хэрэглэгчийн солихыг хүссэн номны ID болон хэрэглэгчийн ID хадгалсан нь	38
A.3	Base дээрх бүртгүүлсэн номны мэдээлэл хадгалсан нь	38

ХҮСНЭГТИЙН ЖАГСААЛТ

4.1	User.....	14
4.2	Book	15
4.3	Exchange	16
4.4	Register (Бүртгэл).....	17
4.5	Book Add	18
4.6	Book Rating (номны үнэлгээ)	19
4.7	Book Category	20
5.1	Work Plan	24

УДИРТГАЛ

Програм хангамж (D061302) хөтөлбөрийн 3р түвшний оюутан Баасансүрэн овогтой Мөнх-Ундрал нь UNIMEDIA SOLUTIONS компанид 21 хоногийн хугацаанд INTE401 хичээлийн индекстэй үйлдвэрлэлийн дадлагыг удирдамж, дадлагын ажлын төлөвлөгөөний дагуу дадлага хийсэн бөгөөд энэ хугацаандаа loopback болон vue.js ашиглан front-end, back-end дээр ажиллаж онолын мэдлэгээ баталгаажуулж, практик дадлага эзэмшиж өргөн хүрээний мэдлэгтэй болж чадсан мөн судалгаа шинжилгээ хийж түүнийгээ практикаар хэрэгжүүлсэн.

Миний хувьд өгөгдлийн сан талд буюу backend талыг хийж үзэж байгаагүй сурахыг хүсдэг байсан боловч хаанаас яаж эхэлж сурахаа мэдэхгүй байсан. Дадлага хийнэ гэж ирэхдээ өгөгдлийн сан, back талдаа илүү их зүйлийг сурна гэж зорилго тавьсан. Системээ хийж эхлэхэд мэдэхгүй зүйл их байсан ч хийх явцдаа өгөгдлийн санг хэрхэн хийх вэ, frontend- тэй яаж холбох, loopback болон vue.js хэлний бүтэц зэргийг ойлгож давуу талуудыг нь мэдэж авсан.

Энэхүү тайланд дадлага хийсэн байгууллага, мөн дадлага хийх хугацаандаа юуг шинээр мэдэж авсан, түүнийгээ хэрхэн хэрэгжүүлсэн тухай өгүүлэх болно.

1. СИСТЕМИЙН ТАНИЛЦУУЛГА

1.1 Компанийн танилцуулга

Юнимедиа Солюшнс ХХК компани нь Японы Юнимедиа компанийн Төлөөлөгчийн газар нэртэйгээр 2013 оны 6 сард үйл ажиллагаагаа явуулж эхэлсэн. 2016 оны 11 сард Юнимедиа Солюшнс ХХК хэмээн шинэ компаниа үүсгэн байгуулж, 2017 оны 1 сараас бизнесийн үйл ажиллагаагаа явуулж эхэлсэн юм.

Бид Япон улсад хэрэгжиж буй “Crowdsourcing” төслийн цахим хуудас бэлтгэх, зар сурталчилгаа, ухаалаг утасны хөгжүүлэлтийг гардан хийдэг. Цаашид ч гэсэн Япон дахь ажиллагаагаа түшиглэн, Монголд үйл ажиллагаа явуулж, бусад улс орнуудтай хамтран ажилласанаар нийгэмд хувь нэмрээ оруулна.

Бидний үнэт зүйл бол нийгэмд хэрэгтэй ажил хийж, ажиллаж буй ажиллагсаддаа баяр хөөртэйгээр ажлаа хийдэг үнэ цэнэ бүхий нийгмийн хэсэг байхад оршино.

1.2 Системийн танилцуулга

Манай систем нь Ном солилцооны (Book Exchange) систем бөгөөд хэрэглэгч нэвтэрч (солилцохыг хүссэн номоо системд бүртгүүлж болно) ороод солилцож унших боломжтой номнуудыг харах бөгөөд хэрвээ тус номын жагсаалтаас уншихыг хүссэн ном нь байвал тухайн номыг солилцох button дээр дарж эзэнтэй нь холбогдож номоо солилцох журмаар ажилдаг.

1.2.1 Purpose

Таны уншсан номыг хэн нэгэн уншихыг хүсэж байгаа юм биш биз

Таны уншихыг хүссэн ном өөр хэн нэгэнд байгаа юм биш биз

2. ИЖИЛ ТӨСТЭЙ СИСТЕМИЙН СУДАЛГАА

Ном солилцох вэбсайтууд нь хуучин номоо бусад ном эзэмшигчийн ашигласан номтой худалдаа хийх мөн солилцох сонирхолтой ном эзэмшигчдийг холбоход тусална. Хүн бүхэн уншихыг хүссэн номоо уншихад илүү их мөнгө зарцуулахгүй байх боломжтой.

2.1 Ижил төстэй систем

2.1.1 <https://www.paperbackswap.com/index.php> : *Номнуудаа жагсаагаад 1.7 сая номоос сонгоно.*

2.1.2 <https://www.bookcrossing.com/?> : *Номоо бүртгүүлээд паркын сандал дээр эсвэл биеийн тамирын зааланд байрлуулснаар шинэ эзэн олох боломжтой.*

2.1.3 <http://booktooch.com/> : *Номнуудаа хүссэн хүмүүст илгээгээд дараа нь бусад хэрэглэгчдээс ном худалдан авахад оноогоо ашиглаарай.*

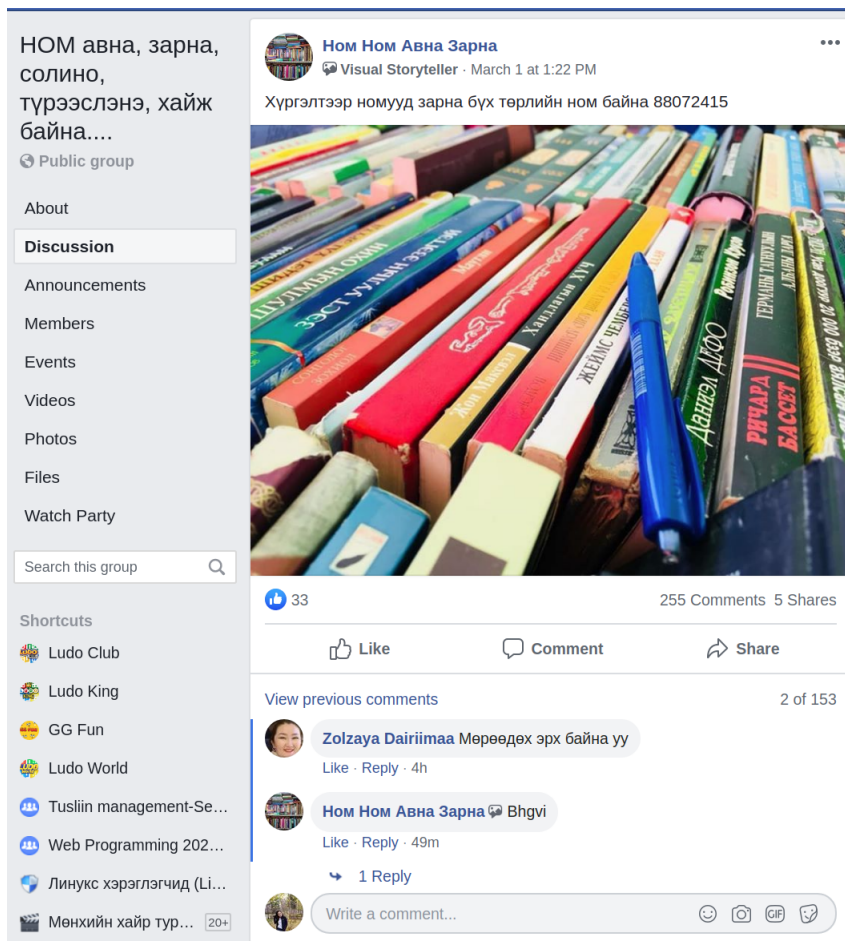
2.1.4 <https://www.booksfreewap.com/> : *Своп шаардлага байхгүй бөгөөд хүлээн авагч шуудангийн төлбөрийн байнга төлдөг.*

2.2 Системүүдийн шинж чанарыг харьцуулах

- Системийн ижил төстэй шинж чанар *[Ижил төстэй талаар харьцуулалт хийх]*
 - Номны хайлтын хэсэг нь title, author, ISBN ашиглан хайх боломжтой байсан.
 - Хэрэглэгч номоо бүртгүүлнэ.
 - Бусад хэрэглэгчийн оруулсан номыг харах боломжтой. гэхдээ тус бүрдээ онцлог шинж чанартай.

Хамгийн чухал нь монголд ийм систем байдаггүй бөгөөд номонд дурлагсад маань фэйсбүүк групп дээр зар тавьдаг бөгөөд бүх хэрэглэгчид хүрэх нь төдийлөн сайн биш эмх замбараагүй

байдаг.



Зураг 2.1: Фэйсбүүк дээрх хэрэглэгчдийн одоогийн байдал

Хэрэглэгчид уншсан болон уншихыг хүссэн номны нэр болон зургийг постоор оруулж хайдаг гэвч 100 гаруй сэтгэгдэл гүйлгэж хайна гэдэг хэцүү асуудал. Энэ нь хэрэглэгчдийг залхаах, цаг хугацаа алдах зэрэг бэрхшээлүүдийг учруулдаг. Тиймээс манай систем нь шууд хайлт хийдгээрээ давуу талтай.

2.3 Судалгаа

Дадлагын ажлаар JavaScript хэлний сангууд болох NodeJS, Vue.js болон өгөгдлийн санг Loorback ашиглан хэрэгжүүлсэн.

1. Node.JS нь Браузараас гадна JavaScript кодыг гүйцэтгэдэг орчин юм. Скриптүүдийг сервер талд ажиллуулж хэрэглэгчийн веб браузерт агууламж (content) очихоос өмнө динамик веб хуудсыг бий болгодог.

2. Loopback4- NODE JS дээр суурилсан ба TypeScript мөн Backend хийхэд зориулагдсан учир илүү хурдан хөгжүүлэлт хийх боломжтой. Base дээр Schema үүсгэхэд хялбар (migrate) санагдсан. Server-лүүгээ орж гүйцэтгэлээ (тестлэл хийх) шалгах боломжтой байсан.

3. Vue.Js- Хэрэглэгчийн интерфейс ба JavaScript. Vuetify- тай хосолсоноор css бичих шаардлагагүй мөн Project үүсгэж ажиллахад хялбар байсан.

4. Axios- хөтөч болон Node.JS-д зориулсан HTTP хэрэглэгч

5. MYSQL workbench- Программын өгөгдлийг загварчлах схемд суурилсан (schema-based) шийдэл

3. СИСТЕМИЙН ШААРДЛАГА

3.1 Функциональ шаардлага

1. Систем нь хэрэглэгчийг бүртгэнэ.
2. Систем нь хэрэглэгчийн нэр, утасны дугаарыг бүртгэх ёстой.
3. Хэрэв хэрэглэгч бүртгэлд шаардлагатай хэсгийг бөглөхгүй орхигдуулсан бол систем сануулна.
4. Хэрэв хэрэглэгчид ойлгомжгүй зүйл байвал систем гарын авлагыг агуулсан байх ёстой
5. Хэрэглэгч өөрийн бүртгэлийн мэдээллийг сольж , шинэчилж болно
6. Хэрэглэгч өөрийн анкетыг бөглөж бэлтгэнэ
7. Хэрэглэгч хэрэв нууц үгээ мартсан бол систем нөөц хаяг руу холбоос явуулж баталгаажуулна
8. Хэрэглэгч номны нэр, зохиогч, солилцох төрлийг харах боломжтой байх
9. Хэрэглэгч өөрт таалагдаж буй номны эзэнтэй холбоо тогтоох боломжтой
10. Сонгосон номоо цуцлах боломжтой байх
11. Хэрэглэгч өөрийн номоо бүртгүүлэх боломжтой байх
12. Хэрэглэгч зар тавих тутамд ирэх мэдэгдлийг өөрийн хүссэнээр тохируулж болно
13. Хэрэглэгч өөрийн бүртгэлийг бүр мөсөн системээс устгаж болно
14. Хэрэглэгч өөрт таалагдсан номоо хадгалах боломжтой байна.
15. Хэрэглэгч асуулт асуух боломжтой байх.

16. Хэрэглэгч нийтэлсэн номны зараа шинэчилж өөрчилж болно
17. Хэрэглэгч өөрийн оруулсан номын жагсаалтыг харах боломжтой байх
18. Хэрэглэгч бүх номын жагсаалтыг харах боломжтой байх

Систем талын

- /ХФ10/ Систем нь номны талаарх мэдээллийг хүлээн авах, сонголт хийх боломжийг хэрэглэгчид олгох хэрэгтэй.

/ФШ-11/ Систем нь хэрэглэгчийн нэр, утасны дугаар зэргийг оруулан бүртгүүлэх ёстой.

/ФШ-12/ систем нь номыг ISBN- ээр бүртгэх боломжтой байх.

/ХФ20/ Систем нь хэрэглэгчийн сонгосон номыг мэдэгдэж тус номны эзэнд сонголтын мэдээллийг харуулна.

/ФШ-21/ Системд эхлээд хэрэглэгч номоо оруулах, сонгох боломжийг олгох хэрэгтэй.

/ФШ-22/ Систем нь баталгаажуулсан мэдээллийг хэрэглэгчид хүргэх ёстой.

/ФШ-23/ Систем нь сонголтын мэдээллийг хадгалах ёстой.

/ХФ30/ Систем нь сонголт хийх явцыг олгох ёстой.

/ФШ-31/ Сонгосон номыг баталгаажуулах ёстой.

/ФШ-32/ Сонголтын мэдээллийг бүртгэх ёстой.

/ХФ40/ Систем нь хэрэв хэрэглэгчийн солихыг хүссэн ном байхгүй тохиолдолд хэрэглэгчийн хүссэн номтой төрлийн хувьд адил өөр номыг санал болгох ёстой.

3.2 Функциональ бус шаардлага

1.1 Availability

- 365 хоногийн 24 цагийн турш тус апп-г ашиглах боломжтой байх.

- Апп-н шинэ хувилбар install хийснээс 2 хүртэлх минутны дотор ашиглах боломжтой байх

1.2 Effivency(Performance)

- Хэрэглэгч 5 хүртэлх секундны хугацаанд sign in хийх боломжтой байх
- Хамгийн ихдээ 2 секундыг хэрэглэгчийн Interface солигдоход зарцуулах
- Хэрэглэгчээс ирсэн report 3 минутны дотор системийн инженерт ирдэг байх
- Нэгэн зэрэг 4000 хүртэлх хэрэглэгч хандах боломжтой байх

1.3 Flexibility

- Англи болон Монгол хэл дээр ашиглаж болдог байх
- Хэрэглэгчдэд харагдаж болзошгүй source code нь hard coded байх

1.4 Integrity(Security)

- Хэрэглэгч бүр нөөц хаягаа бүртгүүлсэн байх
- Сэжигтэй нэвтрэх тохиолдол бүртгэгдсэн бол хэрэглэгчийн утасны дугаар руу мессеж явуулж баталгаажуулах

1.5 Interoperability

- Android , ISO дээр ажилладаг байх
- Хууль эрх зүйн хүрээнд гажуудалгүй байх

1.6 Reliability

- ROCOF(Rate of change of frequency) нь 1/1000 байх

1.7 Robustness(Fault Tolerance)

- Хэрэглэгч хайх түлхүүр үгийг алдаатай оруулсан ч гэсэн боломжит хувилбаруудыг санал болгох
- Хэрэглэгчийн нэр, нууц үг тохирох форматын дагуу биш үед хэргэлэгчид анхааруулга хүргэх.
- “Self-healing” асуудал гарвал өөрөө эхний ээлжний арга хэмжээг авах функцтай байх

1.8 Usability

- 16-60 хүртэлх насны хүмүүс ашиглах боломжтой байх
- Нэг гараар ашиглах боломжтой байх

1.9 Maintainability

- Ямар нэгэн гэмтэл асуудлыг ажлын 3 хүртэлх хоногт шийдэх
- Тест хийх хугацаа ажлын 2 хүртэлх хоног байх

1.10 Reusability

- Ерөнхий загвар , хэрэглэгчийн interface-г веб хуудас хийхэд ашиглах
- Ерөнхий консэфтыг веб хуудас хийхэд ашиглах

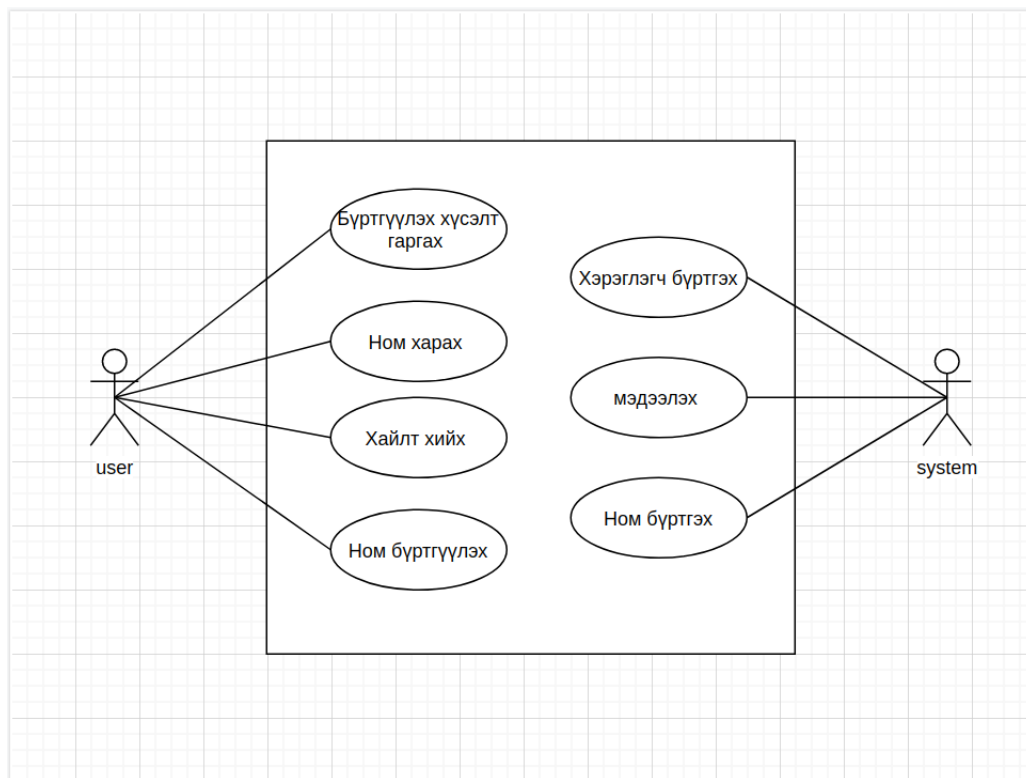
1.11 Testability

- Апп-н хөгжүүлэлтийн хувилбар бүрийг тестлэж үзэх

3.3 Use-case диаграм

Тайлбар

1. Систем хэрэглэгчийг бүртгэнэ, номыг бүртгэнэ, номны мэдээллийг хэрэглэгчид харуулна.



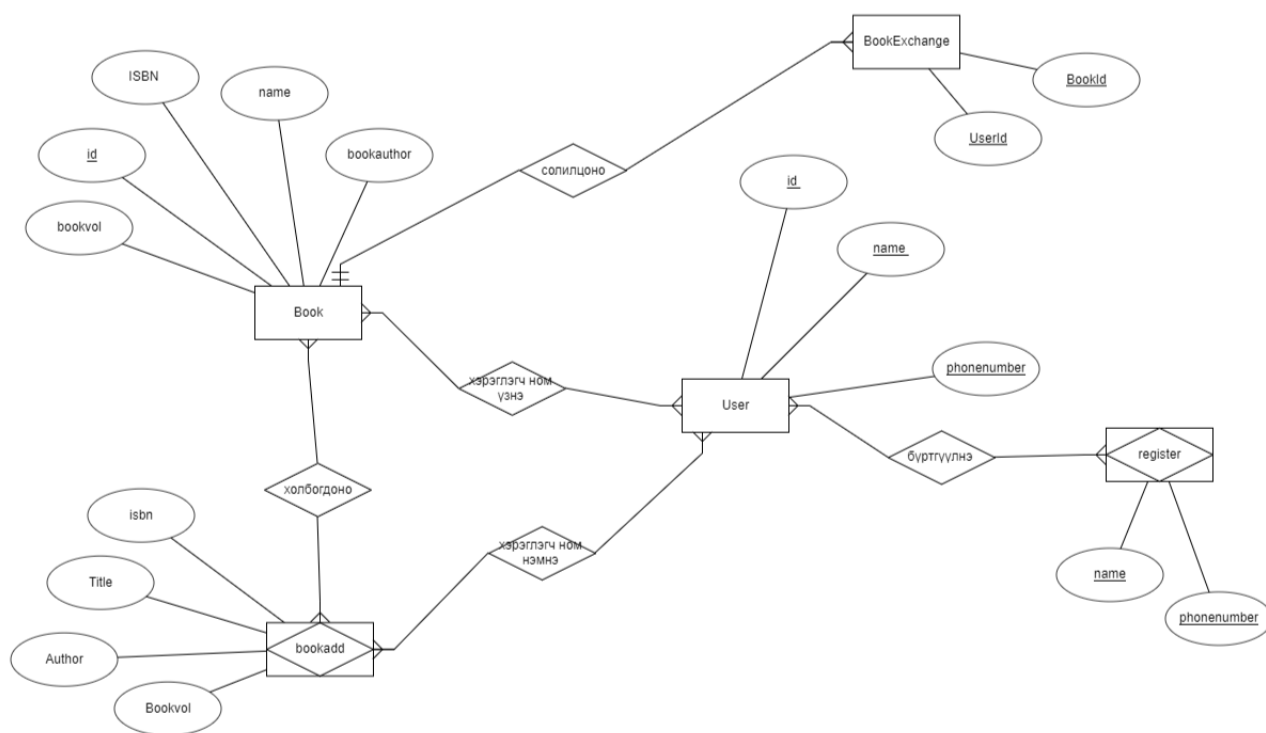
Зураг 3.1: Use Case

2. Хэрэглэгч бүртгүүлнэ, номоо бүртгүүлнэ, номны хайлтыг хийнэ, номын жагсаалтыг харна.
- Бүртгүүлэх хүсэлт гаргах: Системд шинээр хэрэглэгч болохын тулд системд бүртгүүлж орно.
 - Ном харах: Бүртгүүлсний дараа хэрэглэгч системд бүртгэгдсэн байгаа номнуудын жагсаалтыг харах боломжтой.
 - Хайлт хийх: Хэрэглэгч өөрийн хүссэн номоо системээс нэр, зохиолч, ISBN -ээр хайж үр дүнг харах боломжтой.
 - Ном бүртгүүлэх: Хэрэглэгч өөрийн солилцоо хийх номоо системд бүртгүүлэх боломжтой.

- Хэрэглэгч бүртгэх: Хэрэглэгчийн бүртгүүлэх хүсэлтийг систем хүлээн авч хэрэглэгчийг бүртгэх.
- Мэдээлэх: Системд бүртгэгдсэн номны мэдээллийг хэрэглэгчид харуулах.
- Ном бүртгэх: Хэрэглэгчийн бүртгүүлэхийг хүссэн номыг системд бүртгэх.

4. СИСТЕМИЙН ШИНЖИЛГЭЭ ЗОХИОМЖ

4.1 Database diagram

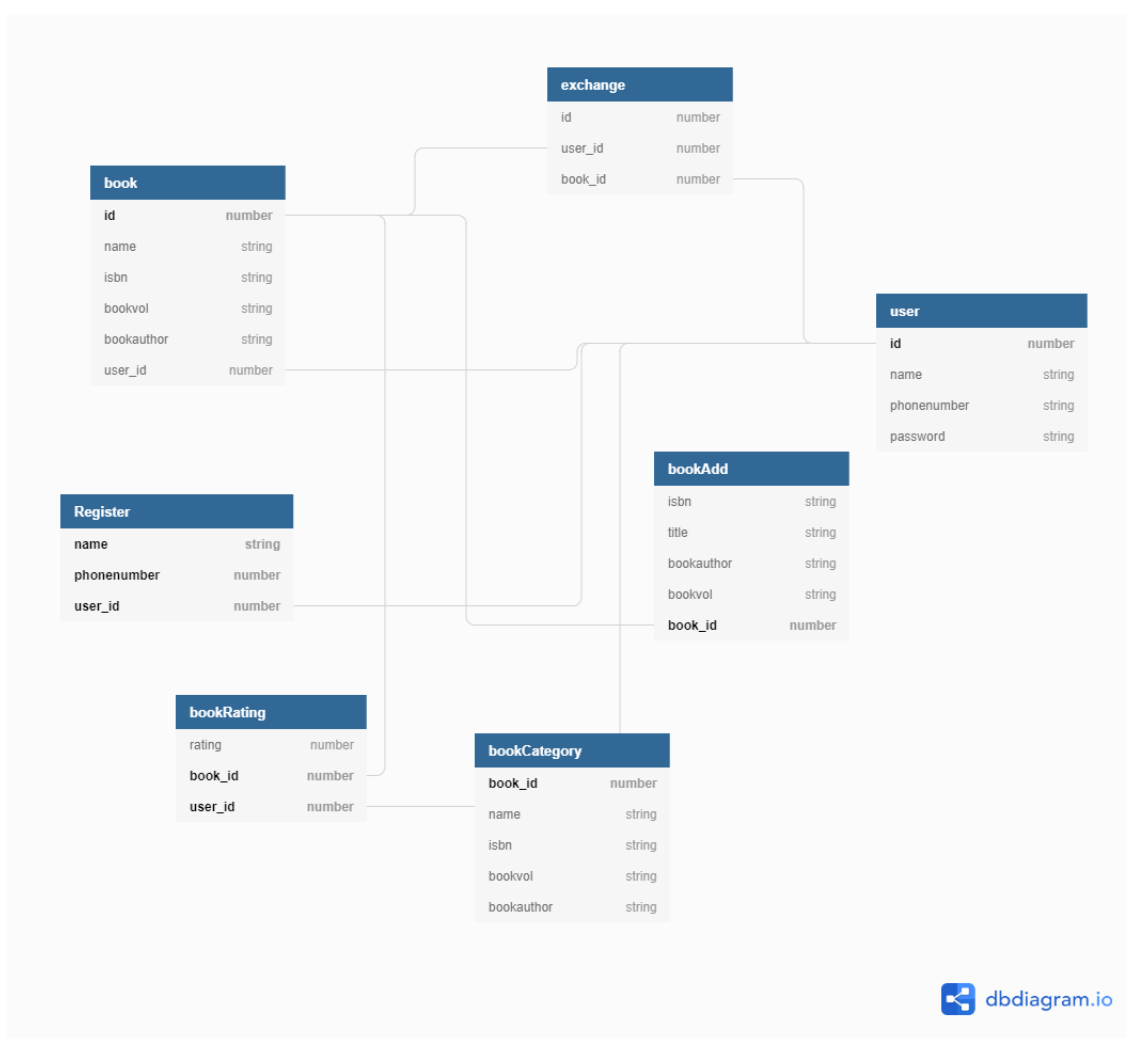


Зураг 4.1: ER Diagram

- Register нь name, phonenumner - тай байна. Хэрэглэгч системд орж номоо солилцохын тулд нэвтэрсэн байх хэрэгтэй. Бүртгүүлэхдээ нэр, утасны дугаараар бүртгүүлнэ. Хэрэглэгчийн утасны дугаар дахин давтагдахгүй.
- User нь id, name, phonenumner- тай байна. Хэрэглэгч бүртгүүлсэнийхээ дараа нэвтэрч орно. Хэрэглэгчийн ID дахин давтагдахгүй.
- Book нь id, isbn, номны нэр, номны боть, номны зохиолчтой. Хэрэглэгч нэвтэрч орсоноор

book хэсэг рүү орно. Үүнд хэрэглэгчдийн оруулсан бүх ном нь харагдана. Номны ID дахин давтагдахгүй.

- Bookadd нь хэрэглэгч өөрийн солихыг хүссэн номоо оруулна. Энэ нь isbn, title, author, bookvol - тай.
- BookExchange нь BookId, UserId - тай байна. Хэрэглэгч солихыг хүссэн номоо сонгосноор тухайн номны эзэн нь гарч ирэх бөгөөд эзэнтэй нь холбогдох боломжтой болно.



Зураг 4.2: Database Diagram

Table 4.1: User

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
id	+		int(20)	Solution ID	No	Хүснэгтийг илэрхийлэх дахин давтагдахгүй түлхүүр
name			varchar(20)	Name	No	Хэрэглэгчийн системд нэвтрэх нэр
phonenumber			int(8)	Phone number	No	Хэрэглэгчийн баталгаажуулах утасны дугаар
password			varchar(20)	password	No	Хэрэглэгчийн системд нэвтрэх нууц үг

Table 4.2: Book

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
id	+		int(20)	Solution ID	No	Хүснэгтийг илэрхийлэх дахин давтагдахгүй түлхүүр
name			varchar(20)	Name	No	Системд бүртгэж буй номны нэр
isbn			int(8)	isbn	No	Номны олон улсын стандартчилагдсан код
bookvol			varchar(20)	bookvol	No	Системд бүртгүүлж буй номны боть
bookauthor			varchar(20)	Book Author	no	Номны нэр
user_id		+	varchar(20)	user ID	No	Хэрэглэгчийн дахин давтагдахгүй түлхүүр

Table 4.3: Exchange

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
id	+		int(20)	Solution ID	No	Хүснэгтийг илэрхийлэх дахин давтагдахгүй түлхүүр
user_id		+	varchar(20)	User ID	No	Хэрэглэгчийн дугаар
book_id		+	varchar(20)	Book ID	No	Номны дугаар

Table 4.4: Register (Бүртгэл)

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
name	+		varchar(20)	Solution ID	No	Хэрэглэгчийн системд нэвтрэх нэр
phonenumner			char(20)	Phone number	No	Хэрэглэгчийн утасны дугаар
user_id		+	varchar(20)	User ID	No	Хэрэглэгчийн дугаар

Table 4.5: Book Add

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
isbn	+		int(20)	Solution ID	No	Номны олон улсын стандартчилагдсан код
title			varchar(20)	Name	No	Системд бүртгэж буй номны нэр
bookauthor			varchar(20)	Book Author	No	Номны зохиолч
bookvol			varchar(10)	Book Vol	No	Системд бүртгүүлж буй номны боть
bookauthor			varchar(20)	Book Author	no	Номны нэр
book_id		+	varchar(20)	book ID	No	Номны дугаар

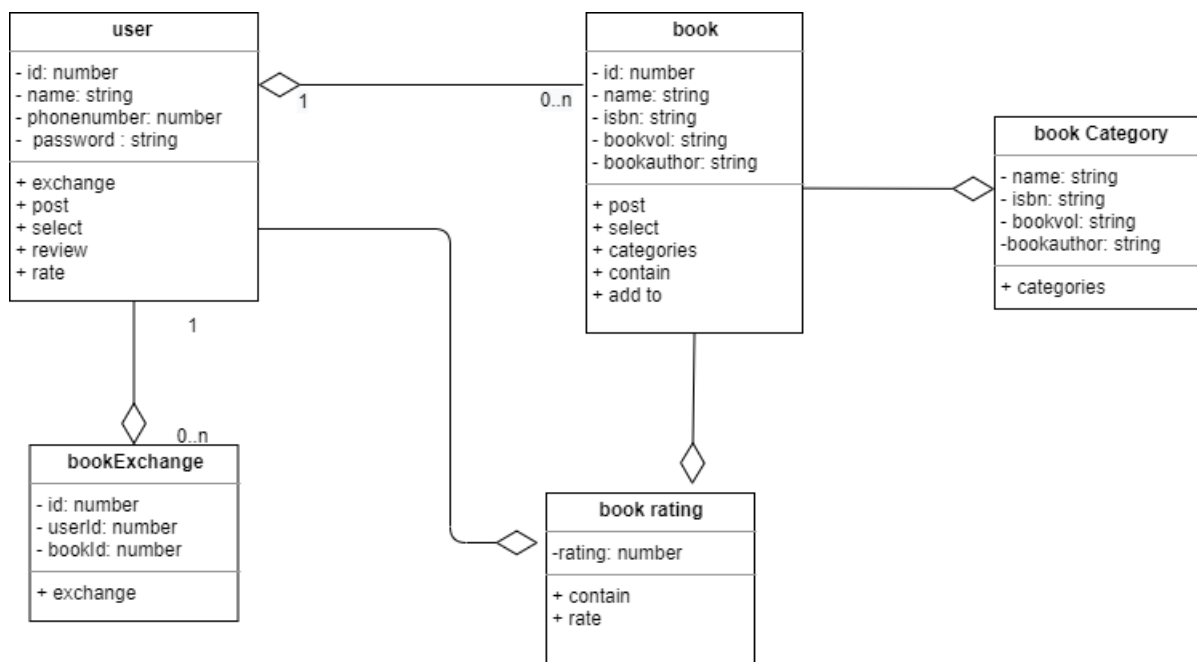
Table 4.6: Book Rating (номны үнэлгээ)

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
rating	+		int(5)	Solution ID	No	Номны үнэлгээ
book_id		+	varchar(20)	Book ID	No	Номны дугаар
user_id		+	varchar(20)	User ID	No	Хэрэглэгчийн дугаар

Table 4.7: Book Category

Баганын нэр	PK	FK	Төрөл ба урт	Тайлбар нэр	Хоосон утга	Тайлбар
book_id	+		varchar(20)	Solution ID	No	Номны дугаар
name			varchar(20)	Book ID	No	Номны дугаар
isbn			varchar(20)	User ID	No	Хэрэглэгчийн дугаар
bookvol			varchar(10)	Book Vol	No	Системд бүртгүүлж буй номны боть
bookauthor			varchar(20)	Book Author	No	Номны зохиолч

4.2 Class diagram



Зураг 4.3: Class Diagram

- User class

User класс нь өөрийн номнуудыг(Book) агуулах ба ном солилцоход BookExchange классын объектыг үүсгэнэ.

- Book class

Book класс нь заавал ямар нэгэн хэрэглэгчид хамааралтай.

- BookExchange class

BookExchange класс нь хэрэглэгчгүйгээр оршин байх боломжгүй учраас нийлмэл харьцаатай байна.

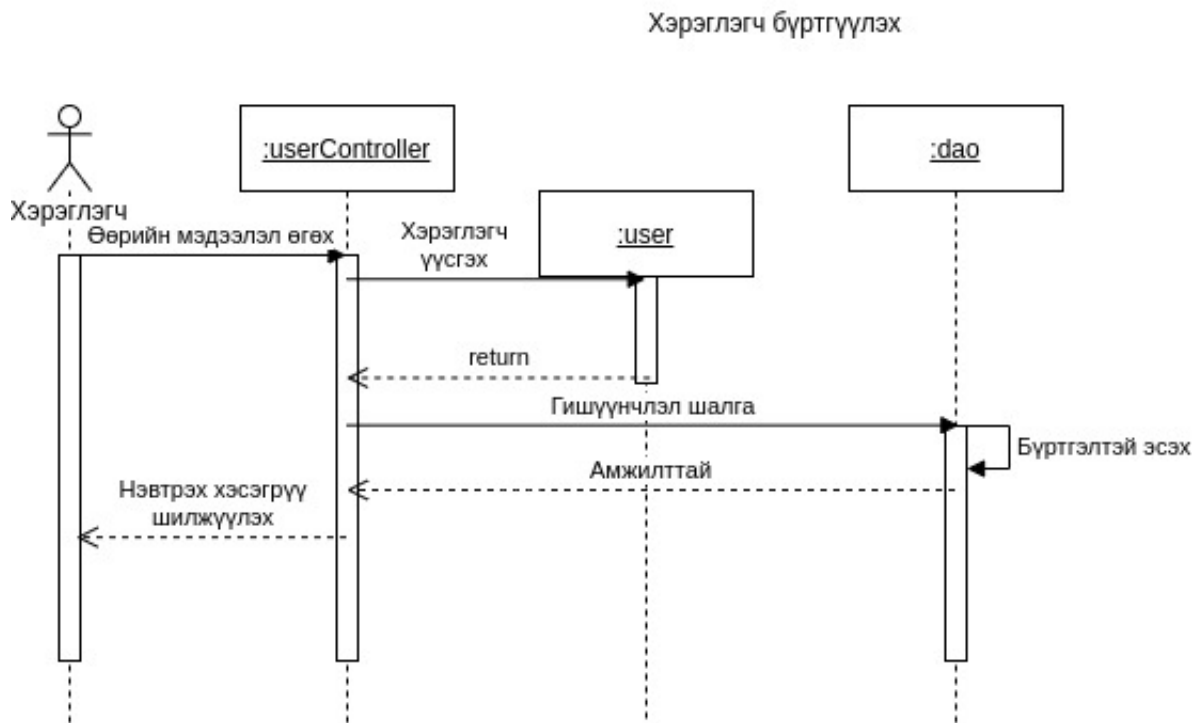
- book rating

Book rating класс нь хүссэн номондоо үнэлгээ өгөх боломжтой. Rating класс нь rate, contain - ийг агуулна.

- book category

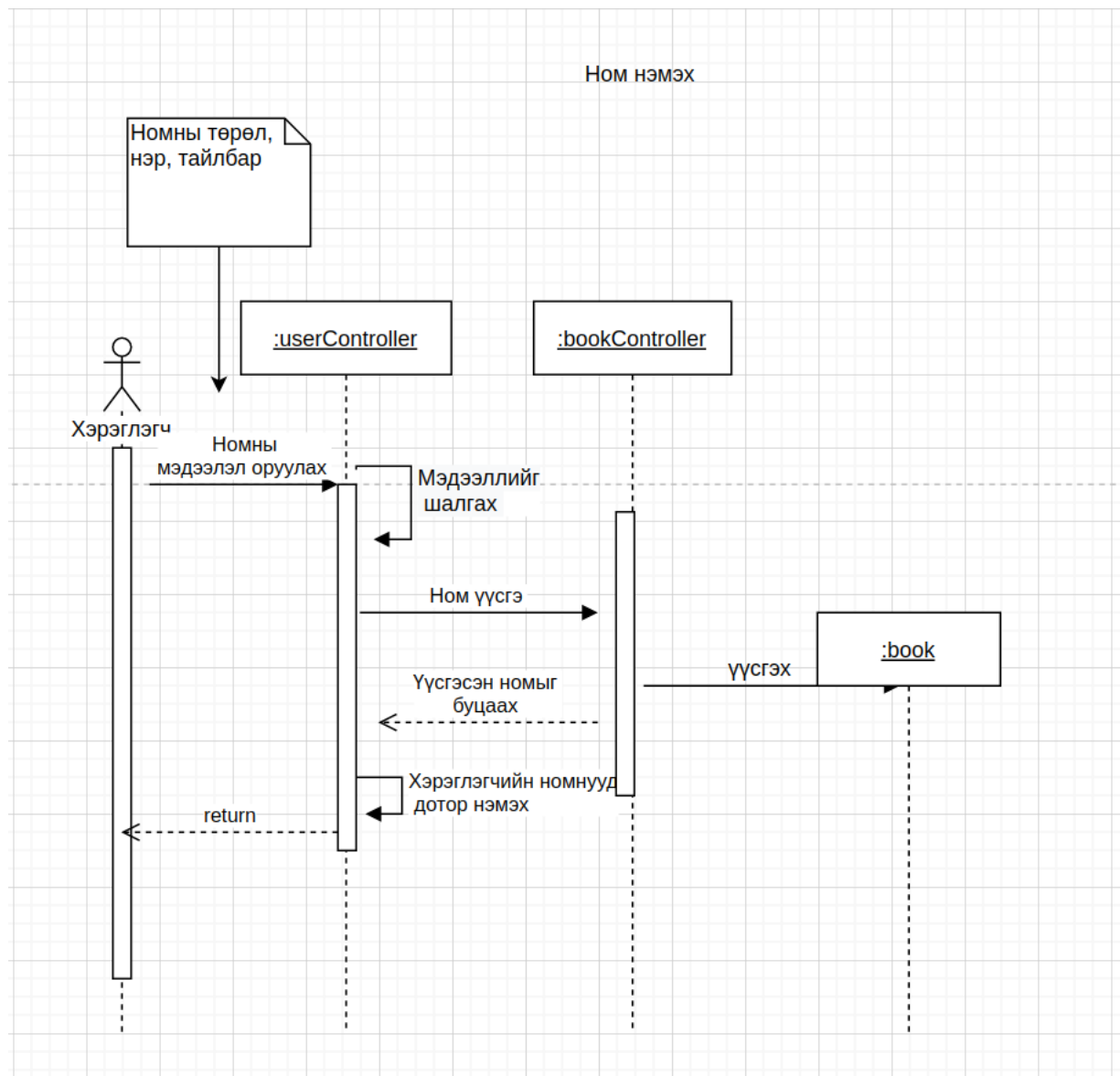
Book category класс нь бүх хэрэглэгчийн оруулсан номыг харах боломжтой. Номны name, isbn, bookvol, bookauthor - ийг агуулна.

4.3 Sequence diagram



Зураг 4.4: Хэрэглэгч бүртгүүлэх дарааллын диаграм

- userController нь хэрэглэгчийг захиран зохицуулах объект.
- user объектыг userController объект үүсгэх ба өгөгдлийн сангийн холбоос объект болох dao-д дамжуулж өгнө.
- dao объект нь ирсэн хэрэглэгчийн объектын мэдээллийг өгөгдлийн санд бүртгэлтэй эсэхийг шалгаад бүртгэлгүй бол хадгална.



Зураг 4.5: Ном нэмэх дарааллын диаграм

- UserController нь хэрэглэгчийн өгсөн номны мэдээллийг bookController-руу дамжуулна.
- bookController нь авсан мэдээллээр номыг үүсгэнэ.

5. ХЭРЭГЖҮҮЛЭЛТ

Table 5.1: Work Plan

Task ID	Work Breakdown Structure	Planned start	Planned Finish	plan	Progress
1	Prototype	2020/01/06	2020/01/06	1 day	100%
2	Database	2020/01/07	2020/01/07	1 day	100%
3	Backend			7 day	94.3%
3_1	Loopback Research	2020/01/08	2020/01/08	1 day	100%
3_2	BooksController Coding	2020/01/09	2020/01/10	2 day	100%
3_3	UserController Coding	2020/01/13	2020/01/15	2 day	100%
3_4	SwapController Coding	2020/01/23	2020/01/24	2 day	80%
4	Frontend			8 day	97.5%
4_1	Vue js Research	2020/01/16	2020/01/16	1 day	100%
4_2	Sign_up	2020/01/17	2020/01/20	2 day	100%
4_3	Sign_In	2020/01/21	2020/01/22	2 day	100%
4_4	Dashboard	2020/01/23	2020/01/23	1 day	100%
4_5	Book create	2020/01/21	2020/01/21	1 day	100%
4_6	Exchange_books	2020/01/24	2020/01/24	1 day	80%
	Validation	2020/01/20	2020/01/20	0	100%
5	test				

5.1 Системийг бий болгоход ашигладаг технологи

5.1.1 Хэрэглэгчийн харицах хэсэг

Хэрэглэгч нэвтрэх хэсэг, ном нэмэх, харах хэсэг, ном солилцох хэсэг, хэрэглэгчийн хэсгийг хэрэгжүүлсэн.

Сервертэй харилцах хэсгийг axios 2 санг ашиглан хэрэгжүүлсэн.

5.1.2 Серверийн хэсэг

Өгөгдлийн сантай холбогдох, загварыг MySQL Workbench ашиглан base дээрээ өгөгдөл хадгалж хэрэглэгч бүртгэх хүсэлтийг сервер талд шалган өгөгдлийн санд оруулах түүнийгээ өгөгдлийн сангаас авах зэрэг хүсэлтүүдийг хүлээн авах хэсгүүдийг хэрэгжүүлсэн.

5.1.3 Хэрэглэгчийн интерфейс

Vue.js vueify ашиглан хэрэглэгчийн интерфейсийг хийсэн.

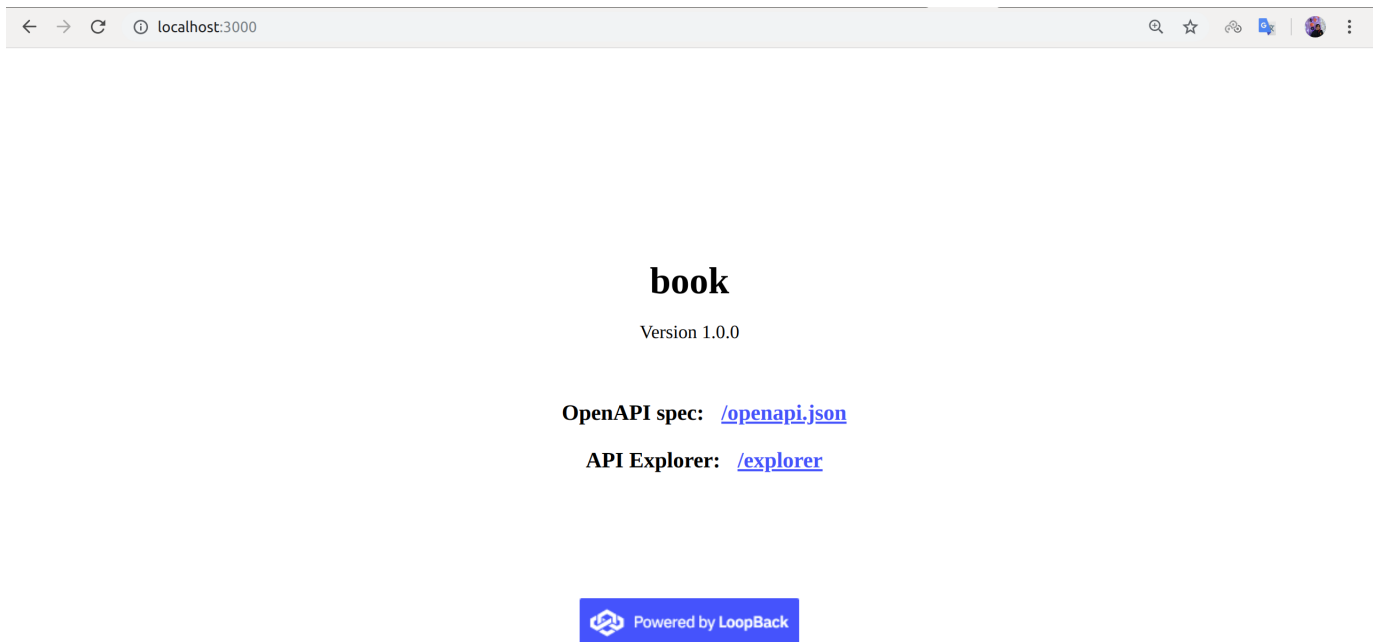
Нэвтрэх хэсэг: энэхүү хэсэг нь хамгийн хүнд байсан бөгөөд хамгийн их цагийг үрсэн. Судалгаа болон суурь мэдлэгийн хүчинд хэцүү байсан ч амжилттай хийж гүйцэтгэсэн. Энэ хэсгийг хийснээр frontend болон backend-ийг хооронд нь холбоход илүү ойлголттой болсон. Ном нэмэх болон солилцоо хийх хэсэгт сервертэй харилцах хэсгийг axios санг ашиглан хэрэгжүүлсэн.

Дадлага хийх хугацаанд хэрэглэгч нэвтрэх, бүртгүүлэх, ном нэмэх, нэмсэн бүх номыг харах мөн солилцохыг хүссэн номны эзнийг харах боломжтой болсон

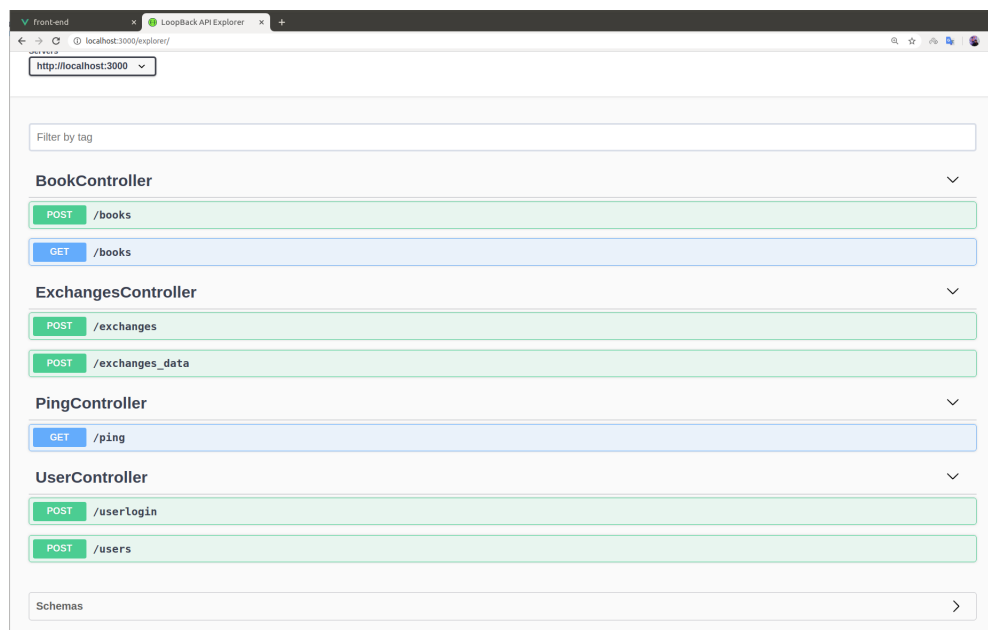
6. ҮР ДҮН



Зураг 6.1: Энэхүү системийг хийхдээ Vue.js , vuetsify ашиглан front-end- ийг хийсэн

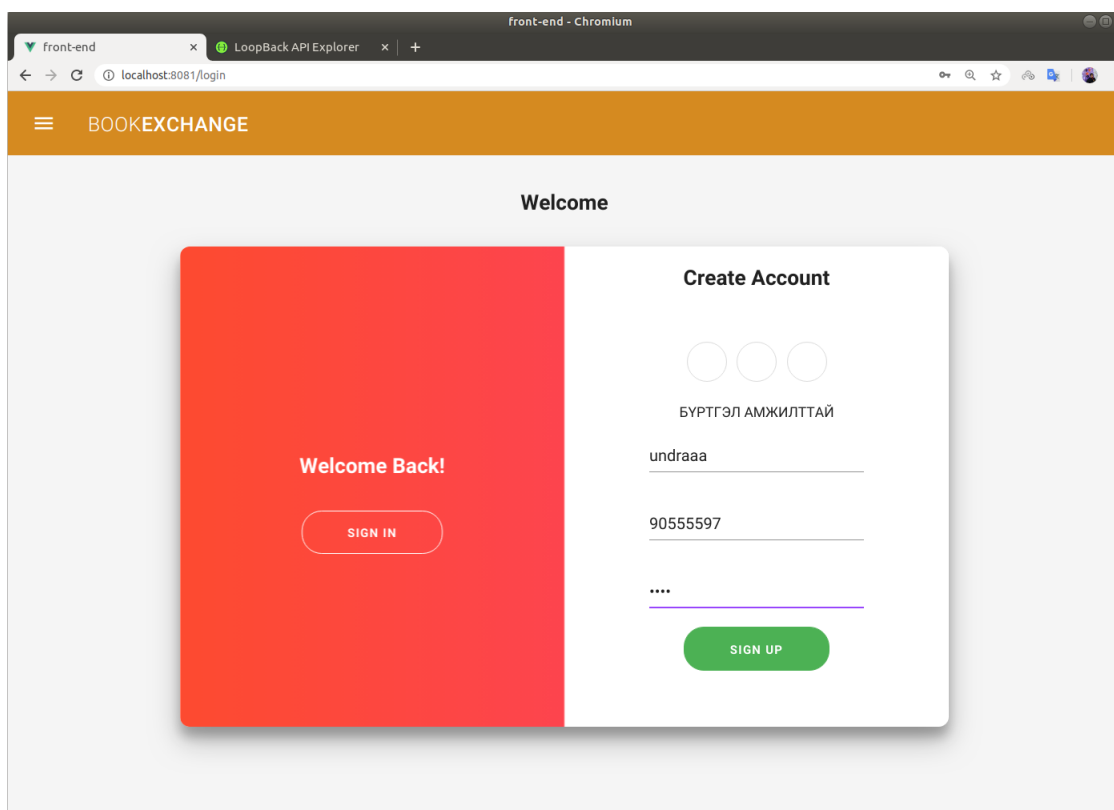


Зураг 6.2: Loopback4 дээр project үүсгэн backend-ийг хийж гүйцэтгэсэн



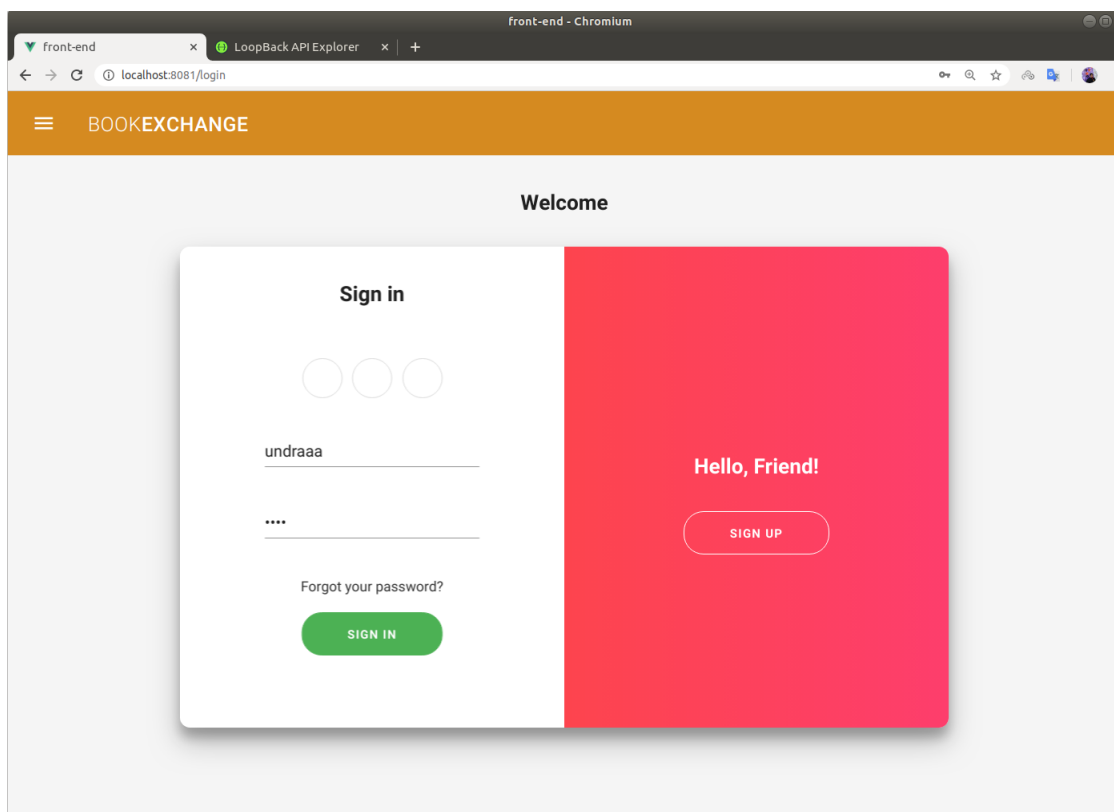
Зураг 6.3: get, post - оор дамжуулан ном нэмэх харах үйл ажиллагаа

Жишээ нь: GET BOOKS - рүү ороод номоо нэмэхэд энэ нь base дээр адил нэмэгдэнэ.



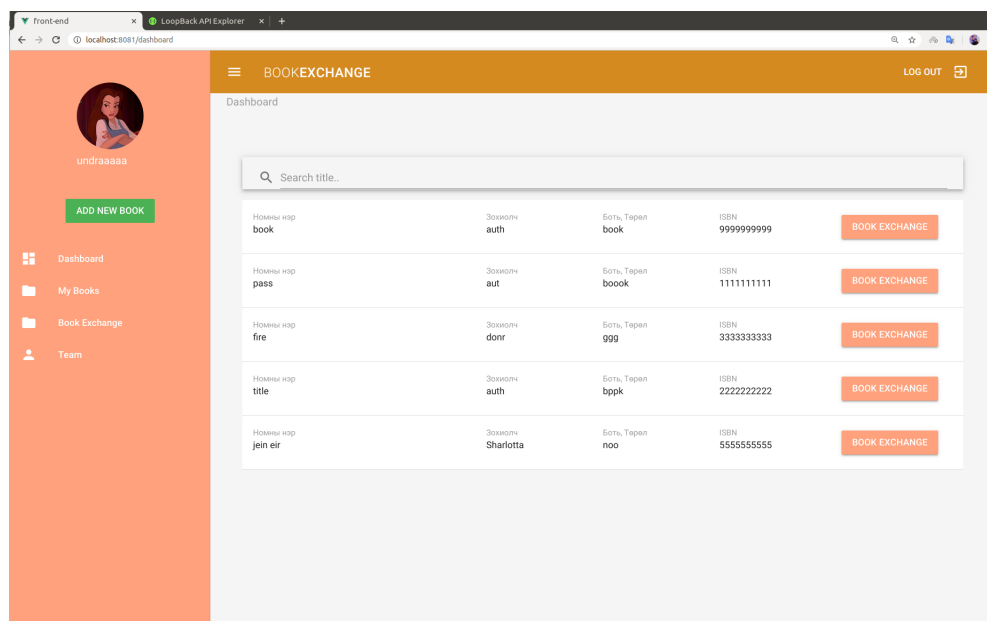
Зураг 6.4: Хэрэглэгч бүртгүүлэх хэсэг

Хэрэглэгч өөрийн нэр утасны дугаараар бүртгүүлнэ. Хэрэв нэр давхардсан бол систем хэрэглэгчид бүртгэлтэй хэрэглэгч гэдгийг сануулна. Бүртгэл амжилттай болсон тохиолдолд нэвтэрч орно.

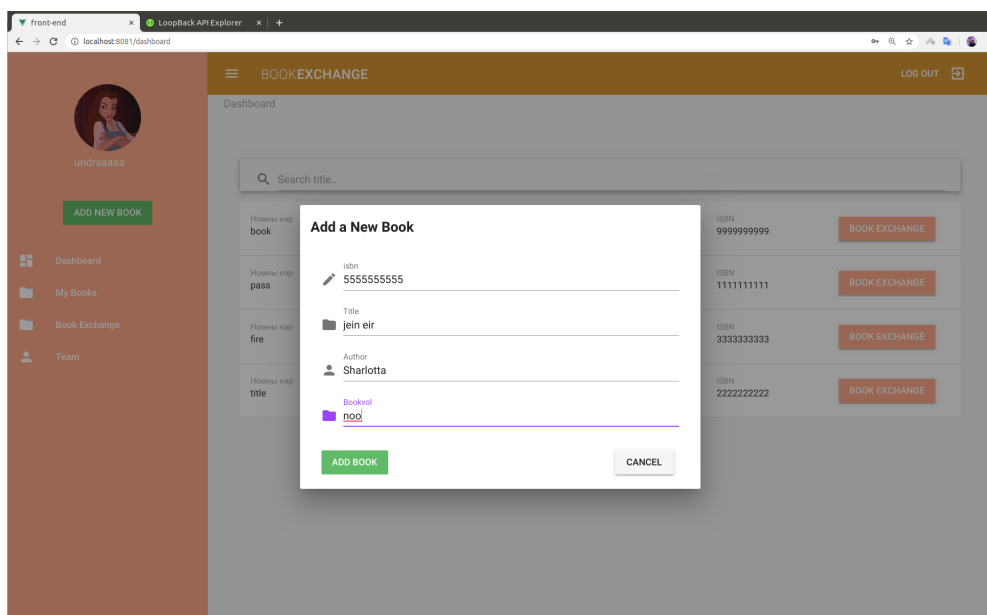


Зураг 6.5: Хэрэглэгч нэвтрэх хэсэг

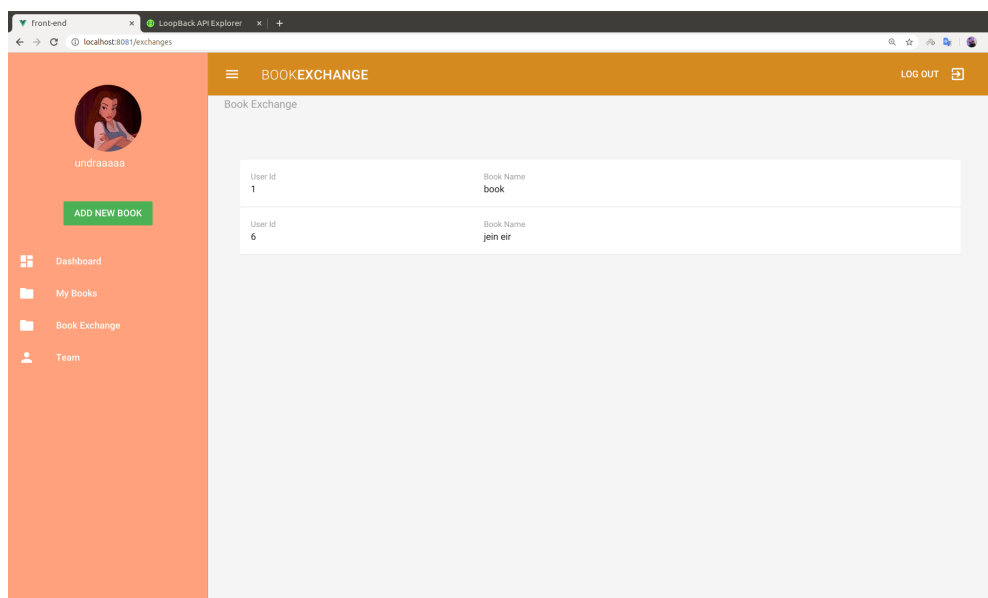
Хэрэглэгч өөрийн нэр password -ийг оруулан нэвтэрнэ. Бүртгүүлээгүй болон нэр, password зөрсөн тохиолдолд хэрэглэгчид сануулна. Амжилттай нэвтэрсэн тохиолдолд үндсэн цонх гарч ирнэ.



Зураг 6.6: Хэрэглэгч нэвтэрч орсоны дараах үндсэн цонх



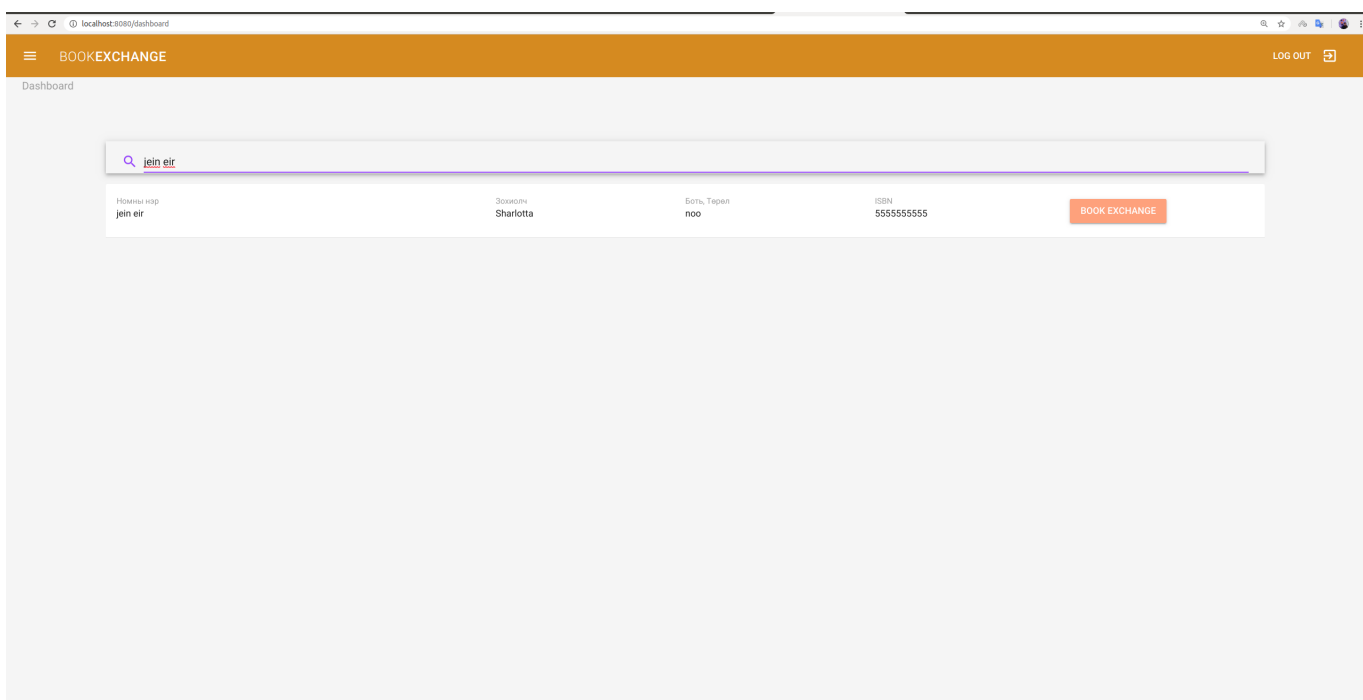
Зураг 6.7: Ном нэмэх хэсэг



Зураг 6.8: Book Exchange хуудас

Ном солих button дээр дарснаар Book Exchange хуудаст таны сонирхсон номыг харуулна.

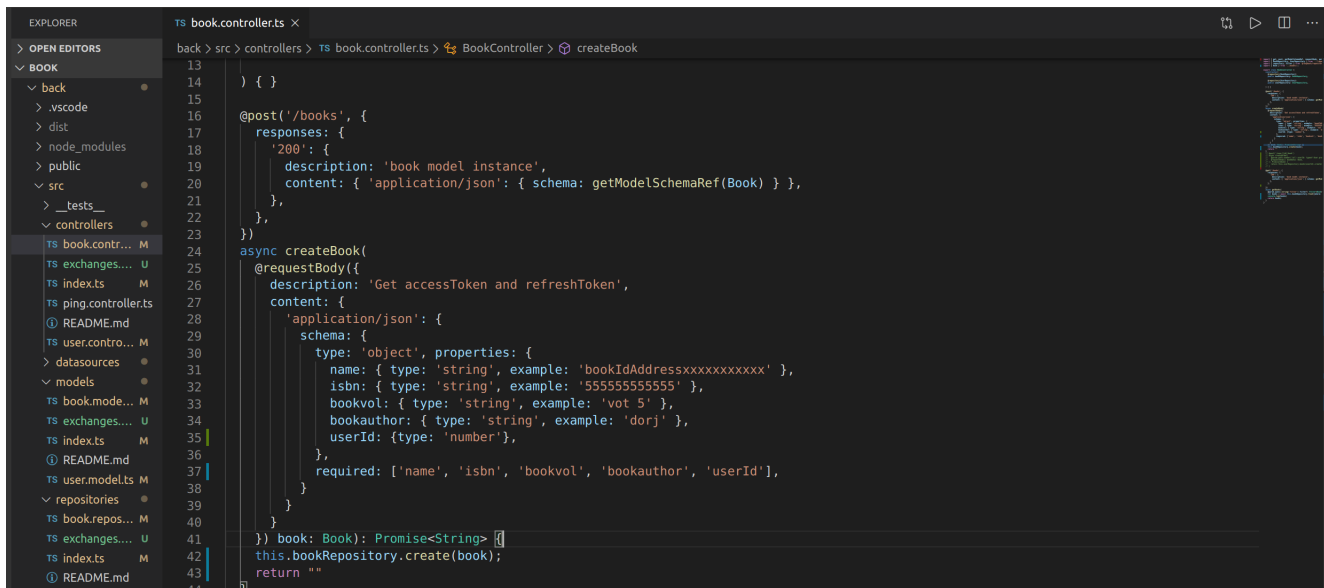
Ингэснээр та тухайн номны жинхэнэ эзэнтэй холбогдон номоо солих боломжтой болно.



Зураг 6.9: Хайлтын хэсэг

Хэрэглэгч цаг хугацаагаа хэмнэж хялбараар хайлт хийн хүссэн номоо хурдан олох боломжтой. Энэхүү зурган дээр номыг номны нэрээр нь хайж байна.

7. КОДЫН ХЭРЭГЖҮҮЛЭЛТ



Зураг 7.1: Ном бүртгүүлэх Back-end талын код

```

50   methods: {
51     submit() {
52       console.log(localStorage.getItem('user'))
53       if(this.$refs.form.validate()) {
54         this.loading = true;
55         const book = {
56           isbn: this.isbn,
57           name: this.title,
58           bookauthor: this.Author,
59           bookvol: this.Bookvol,
60           userId: Number(localStorage.getItem('user'))
61         }
62         console.log(book)
63         this.axios.post('http://localhost:3000/books', book) .then(() => {
64           this.loading = false
65           this.dialog = false
66         })
67       }
68     },
69   },

```

Зураг 7.2: Ном бүртгүүлэх Front-End талын код

Дүгнэлт

Дадлагын ажлыг 2020.01.06-2020.01.24 хүртэл 3-н долоо хоногийн хугацаанд UNIME-DIA SOLUTIONS компанид хийж гүйцэтгэсэн. Дадлагын явцдаа судалгаа шинжилгээ хийх, хичээлээр олж авсан суурь мэдлэгээ ажил хэрэгт хэрхэн ашиглах талаар суралцсан. Судалгаагаар олж авсан мэдлэгээ бие даан туршиж ойлгомжгүй зүйл гарвал Mentor – оосоо асуун туслалцаа авч байсан бөгөөд мэдлэгээ улам бататгаж, практик дээр ашиглах, хэрэгжүүлэхийг сурсан. Ажлыг гүйцэтгэж байх үед нилээд асуудлууд тулгарч байсан нь буцаад судалгааны хэсэг рүү орохоос өөр аргагүйд хүргэж энэ нь судалгаа шинжилгээ ямар нягт холбоотойг ойлгуулсан. Мөн шинжилгээг буруу хийснээр ямар их цаг, хөдөлмөр ашиггүй үрэгдүүлэхийг өөрийн биеэр мэдэрсэн. Гэсэн ч огт шантралгүй үргэлжлүүлэн, хийж буй системээ эхлээд ажилдаг, дараа нь ойлгомжтой хэрэглэхэд хялбар дараа нь дизайн гэсээр сайжруулсаар төгс биш ч тодорхой хэмжээнд хийж гүйцэтгэсэн.


```

@post('/users', {
  responses: {
    '200': {
      description: 'User model instance',
      content: { 'application/json': { schema: getModelSchemaRef(User) } },
    },
  },
})
})
async createUser(
  @requestBody({
    description: 'Get accessToken and refreshToken',
    content: {
      'application/json': {
        schema: {
          type: 'object', properties: {
            name: { type: 'string', example: 'userIdAddress' },
            phonenumber: { type: 'string', example: '12345' },
            password: { type: 'string', example: 'password' },
          },
          required: ['name', 'phonenumber', 'password'],
        }
      }
    }
  }) user: User): Promise<String> {

  try {
    let existsUser = await this.userRepository.findOne({ where: { name: user.name, password: user.password } })

    if (existsUser) {
      throw new HttpErrors.BadRequest('user already registered')
    }

    await this.userRepository.create(user)
  }
  catch (error) {
    throw new HttpErrors.BadRequest('user already registered')
  }

  return ''
}

```

Зураг 7.3: Хэрэглэгчийн бүртгэл Back-end

```

    },
    async signup() {
      if(this.$refs.form.validate()) {
        this.loading = true
        const user = {
          name: this.username,
          phonenumber: this.phonenumber,
          password: this.password,
        }
        try {
          let response = await axios.post('http://localhost:3000/users', user);
          if (response && response.status == 200) {
            this.message = 'БҮРТГЭЛ АМЖИЛТТАЙ'
          }
        } catch(error) {
          this.message = 'Бүргэлтэй хэрэглэгч тул нэрээ солин дахин бүртгүүлнэ үү'
        }
        this.loading = true;
      }
    },
    async signin() {

```

Зураг 7.4: Хэрэглэгчийн бүртгэл Front-end

```

TS book.controller.ts TS user.model.ts x
back > src > models > TS user.model.ts > ...
6   @property({
7     type: 'number',
8     id: true,
9     generated: true,
10  })
11  id?: number;
12
13  @property({
14    type: 'string',
15    required: true,
16    index: {
17      unique: true,
18    }
19  })
20  name: string;
21
22  @property({
23    type: 'string',
24    required: true,
25  })
26  phonenumber: string;
27
28  @property({
29    type: 'string',
30    required: true,
31  })
32  password: string;
33

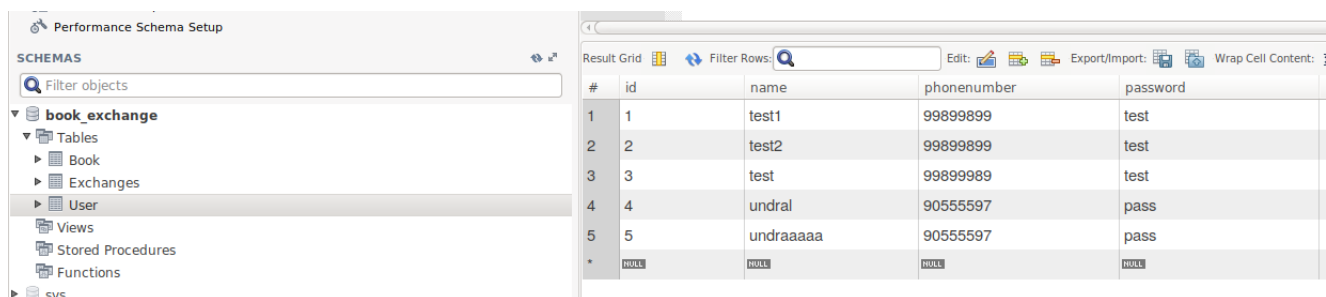
```

Зураг 7.5: User model

Bibliography

- [1] <https://loopback.io/doc/en/lb4/>
- [2] <https://vuejs.org/>
- [3] <https://vuetifyjs.com/en/getting-started/quick-start/>
- [4] <https://nodejs.org/en/docs/>

А. ӨГӨГДӨЛ ХАДГАЛАХ



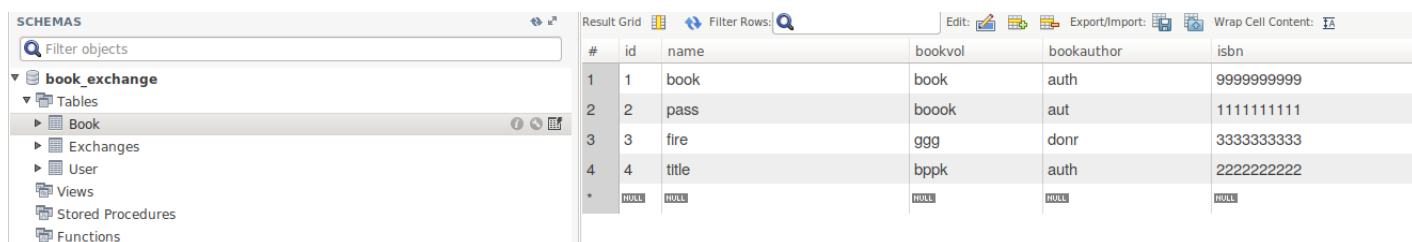
#	id	name	phonenumber	password
1	1	test1	99899899	test
2	2	test2	99899899	test
3	3	test	99899899	test
4	4	undral	90555597	pass
5	5	undraaaaa	90555597	pass
*	NULL	NULL	NULL	NULL

Зураг А.1: Base дээрх бүртгүүлсэн хэрэглэгчийн мэдээлэл хадгалсан байдал



#	id	bookId	userId
1	1	1	1
2	2	1	3
3	3	2	3
4	4	1	5
*	NULL	NULL	NULL

Зураг А.2: Base дээрх хэрэглэгчийн солихыг хүссэн номны ID болон хэрэглэгчийн ID хадгалсан нь



#	id	name	bookvol	bookauthor	isbn
1	1	book	book	auth	9999999999
2	2	pass	boook	aut	1111111111
3	3	fire	ggg	donr	3333333333
4	4	title	bppk	auth	2222222222
*	NULL	NULL	NULL	NULL	NULL

Зураг А.3: Base дээрх бүртгүүлсэн номны мэдээлэл хадгалсан нь