

Төсөл 8: Фото зургийн програм спринт

[Програм хангамжийн шалмаг](#) (agile) хөгжүүлэх арга зүйг ашигладаг удирдлагын шинэ баг таны зургийн аппын хөгжүүлэлтийг удирдаж эхэллээ. Дараагийн [спринтийн](#) удирдлагын баг хувьд та төслийг хэрэгжүүлэх инженерийн хувьд сонгох боломжтой ажлуудыг багтаасан [хэрэглэгчийн түүхийн скрамсамбарыг](#) тодорхойлсон. Түүх бүрт хэд хэдэн стори оноо оноодог бөгөөд спринтийн хувьд танд дор хаяж 12 оноотой бүрэн хэмжээний стори хэрэгтэй болно.

Тохиргоо

Та өөрийн системд MongoDB болон Node.js суулгасан байх ёстой. Үгүй бол одоо суулгах зааврыг дагана уу.

Төсөл 7-ын нэгэн адил та өмнөх төслийн хавтас (**project7**)-ийн хуулбарыг **project8** нэртэй сан болгон хийж эхэлнэ. **project8** хавтас руу [энэ зип](#) файлын агуулгыг задлана. Энэхүү зип файл нь илгээх үйл явцын нэг хэсэг болгон бөглөх маягт агуулсан submission.txt нэртэй файлыг нэмэх болно. Энэ төслийн бүх ажлыг project8 хавтас хэсэгт хийгээрэй.

Төсөл 7-ын нэгэн адил танд MongoDB болон таны вэб сервер ажиллах шаардлагатай болно.

Даалгавар 1: Зарим түүхийг хэрэгжүүлэх (75 оноо)

Төслийн scrumboard-г доор харуулав. Хэрэглэгчийн түүх бүр түүхтэй холбогдож, уг функц юу хийх ёстойг зааж өгдөг. Та хийх

түүхээ сонгохдоо болгоомжтой байх хэрэгтэй. Түүхүүд хоорондоо харилцан уялдаатай байдаг бөгөөд тэдгээрийг хамтдаа хийхэд хялбар эсвэл хэцүү болгодог. Жишээлбэл, хэрэв та зураг дээр харагдах байдлын хяналт нэмэх тухай түүхийг хийхээр сонгосон бол үйл ажиллагааны мэдээллийн (feed) түүх нь хэрэглэгч харах боломжгүй зургийг буцааж өгөх ёсгүй байх ёстой.

Өмнөх даалгавруудаас ялгаатай нь та өгөгдлийн сангийн схем, вэб серверийн API болон ReactJS програмын бүрэлдэхүүн хэсгүүдийг өөрчлөх зэрэг хүссэн өөрчлөлтийг хийх боломжтой. Танд хэрэгцээтэй гэж үзвэл гуравдагч талын програм хангамжийн багцыг ашиглаж болох ба илүүтэйгээр санал болгож байна.

Таны хөгжүүлэх эрх чөлөөг хязгаарлаж байгаа нэг зүйл бол таны зургийн аппликейшн класс-илгээх механизмаар дамжуулж, курсын ажилтнуудад үнэлгээ өгөхөд саадгүй ажиллах ёстой. Мөн түүхийг хэрэгжүүлэхдээ таны вэб програм ашиглалт/туршилтын явцад гацахгүй байхыг анхаарах хэрэгтэй (жишээ нь, webServer.js програм ажиллах явцад гацах ёсгүй). Дэлгэрэнгүй мэдээллийг дараагийн хэсгүүдээс илгээх мэдээллийг харна уу.

Скрам самбар		
#	Хэрэглэгчийн түүх	Стори оноо
1	Хэрэглэгчийн профайлын дэлгэрэнгүй мэдээллийг ашиглалттай хамт өргөтгөх	3
2	коммент дээр @mentions хийх	6
3	Зургийн харагдах байдлыг хянах	4

4	Activity feed	4
5	Сэтгэгдэл, зураг, хэрэглэгчдийг устгах	4
6	Зурган дээр "like" санал өгөх	4
7	Хэрэглэгчдэд зориулсан дуртай зургийн жагсаалт	6
8	Хажуугийн жагсаалт нь хэрэглэгчдийг шинэ үйл ажиллагаатай гэж тэмдэглэдэг	3
9	Зурган дээр тагтлах	10

Зөвлөмж

Загварын хувьд хэрэглэгчийн түүхүүд нь онцлог шинж чанаруудын хувьд нэг л утга санаатай үзүүлэлт биш юм. Энэ нь инженер та түүхэнд ашиглах арга барилог заагаагүй газруудад боломжийн зүйлийг сонгох замаар дэлгэрэнгүй мэдээллийг бөглөх шаардлагатай гэсэн үг юм. Нэг үндсэн зарчим бол “Аппликешныг хэрэглэгч юу хийхийг хүсч байна вэ?” гэсэн асуултад хариулан хийх явдал юм.

Дүгнэх

Даалгавраас бүтэн кредит(оноо) авахын тулд та хамгийн багадаа түүх хэсгээс 12 Стори оноо (story point) буюу (энэхүү даалгаврын 75 оноотой тэнцэх) хэрэглэгчийн түүхийг(story) хийх шаардлагатай. Таны 12-оос дээш оноо авсан Стори оноог даалгаврын оноотой адилаар тооцож хэрэглэнэ (жишээ нь, 12-оос илүү байх Стори оноог хөрвүүлэх байдал нь 1 Стори оноо = 1 даалгаврын

оноо). Та нийтдээ 18 хүртэлх оноотой нэмэлт түүхийг хийх боломжтой. Өөрөөр хэлбэл та нийт 30 Стори оноо хийснээр 18 хүртэлх онооны нэмэлт кредит авах боломжтой. Бид үнэлгээний тооцоололдоо зөвхөн 5 түүхийг л оруулах болно..

Бид хэсэгчлэн хэрэгжүүлсэн түүхүүд нь хэсэгчлэн оноог нарийн байдлаар хуваарилахыг төлөвлөж байна. Цөөн түүхийг бүрэн гүйцэд хийх нь илүү олон түүхийг хэсэгчлэн хийхээс илүү сайн байх болно.

Хэрэв та npm-г ашиглан аливаа шинэ багцуудыг (package) татаж авах юм бол түүнийгээ package.json файлд нэмж оруулах хэрэгтэйг анхаарна уу (--save сонголтыг ашиглан үүнийг хийх хэрэгтэй) Хэрэв та npm-г ашиглахгүйгээр холбох бол өөрийн project8 хавтасдаа хадгалаарай, ингэснээр таны даалгаврыг үнэлэх харгалзагч багш үүнийг ашиглах боломжтой болно.

Энэ даалгаврыг үнэлэхэд та хэрэгжүүлсэн түүхүүдийнхээ талаар тайлбараа хавсаргах шаардлагатай. Project8 zip файл нь submission.txt файлыг татаж авсан. submission.txt нь бидэнд хэрэгтэй мэдээллийг цуглуулах маягтыг агуулсан. Үүнд таны Стори онооны тооцоолол, онооны хэмжээ, таны олж авсан нийт Стори оноо зэрэг багтана.

submission.txt файлаас гадна нэмэлтээр та өөрийн зургийн програм болон хэрэгжүүлсэн түүхийнхээ богино хэмжээний видеог оруулах ёстой. Видеоны урт нь $12 \times k$ секундээс хэтрэхгүй байх ёстой бөгөөд k нь таны гүйцэтгэсэн Стори оноо юм. Ердийн видеоны урт нь 2 минут буюу түүнээс бага байна. Энэ нь аудио бүхий энгийн дэлгэцийн бичлэг байх ёстой. MacOS дээр та QuickTime, Windows дээр VLC ашиглан дэлгэцийн бичлэг хийх боломжтой. Видеог бүтээхэд хэт их цаг зарцуулах хэрэггүй. Та бичлэгийн чанараар үнэлэгдэхгүй гэдгийг анхаараарай.

Урт нь зөвшөөрөгдсөн дээд хэмжээтэй ойролцоо байгаа видеонууд нь бидний илгээх үйл явцад хэт том байх магадлалтай тул

submission.txt файлын холбоосыг бидэнд өгөх шаардлагатай байна.Видеоогоо YouTube эсвэл Vimeo гэх мэт видео хуваалцах боломжтой вэбсайт руу байршуулах нь иймэрхүү холбоосыг авахад хялбар арга болдог. Үнэлгээ тавих үйл явцын нэг хэсэг болгон бид ангийн хэлэлцүүлгийн форум дээр сайн хийгдсэн видеонуудаасаа хуваалцахаар төлөвлөж байна. Хэрвээ бичлэгээ бидэнтэй хуваалцахыг хүсвэл submission.txt файлд зааж өгөөрэй.

Загварын оноо (5 оноо)

Хэрэв таны асуудлын шийдэл нь MVC-ийн зөв задралтай, MEAN стекийн конвенцуудыг дагаж мөрддөг, ESLint-ийн анхааруулгагүй JavaScript бол эдгээр оноог олгоно.Та илгээхээсээ өмнө `npm run lint`-г ажиллуулахыг хүсэх болно. Нэмж дурдахад таны код, загварууд цэвэр, уншихад хялбар байх ёстой бөгөөд таны вэб хуудсууд дор хаяж "боломжийн сайхан" харагдах байдал гоё байх ёстой.

Анхаар: npm-ээс татагдхааргүй хамаарал(dependencies) (жишээ нь, bower-аас) суулгахдаа `npm run lint` скрипт команд нь таны хамаарлыг агуулсан фолдерыг оруулахгүй байхаар package.json файлыг өөрчилнө үү. Ингэснээр ESLint команд нь зөвхөн таны өөрийн JS файлуудыг шалгах бөгөөд таны ашигладаг номын сан/хамааралтай харгалзах JS файлуудыг алгасах болно (энэ нь линтерийг нэвтрүүлэхгүй байж магадгүй).

Илгээхдээ анхаарах зүйлс

Таны фото апп-г үнэлгээнд өгөхдөө шаардлагатай бүх файлыг (зөвхөн package.json-д заасан файлуудаас бусад нэмэлт файл шаардагдахгүйгээр) багцалж оруулах шаардлагатай. Курсын багш нар таны бүх хавтсыг хуулж авна, гэхдээ node_modules/ болон compiled/ дэд хавтаснуудыг оруулахгүй — эдгээр хавтаснуудыг

(ялангуяа test/ хавтасны доторх node_modules/ хавтсыг ч мөн адил) устгасан эсэхээ шалгаарай.

Дараах зүйлсийг шалгах хэрэгтэй:

- Төслийнхөө явцад ашигласан бүх програм хангамжууд package.json-д заагдсан эсвэл төслийн хавтасны доторх дэд хавтас хэлбэрээр (гэхдээ node_modules/ биш) байршсан байх ёстой.
 - npm install --save <module_name> командыг ашигласнаар тухайн модулийг татаж авч package.json-д автоматаар нэмнэ.
 - **Төслөө өгөхөөс өмнө шалгахын тулд node_modules/ хавтсыг устгаад төслийн хавтас дотор npm install гэж бичиж ажиллуул. Дараа нь вэб апп чинь хэвийн ажиллаж байгаа эсэхийг шалгаарай.**
- loadDatabase.js скрипт файл оршин байгаа бөгөөд MongoDB-г цэвэр байдлаар тохирох схемүүд болон объектуудтай ачааллах чадвартай байх ёстой.
 - Шинэ схем нэмсэн бол уг скриптийг тохируулж өөрчлөх шаардлагатай байж магадгүй (жишээлбэл, шинэ коллекци дэмждэг болсон бол өгөгдлийг суулгахаас өмнө тухайн коллекцуудаас бүх объектуудыг устгах шаардлагатай).
 - **mongod-г дахин эхлүүлээд node loadDatabase.js командыг ажиллуулан өгөгдлийн санг шинээр үүсгэж, суулгаж шалгах хэрэгтэй. Үүний дагуу таны апп ажиллахгүй бол бид шалгаж чадахгүй байх магадлалтай.**
- node webServer.js команд вэб серверийг эхлүүлж, стандарт порт дээр localhost дахь MongoDB-тай холбогдож ажиллах ёстой. <http://localhost:3000/photo-share.html> хаягаар таны апп нээгдэх ёстой.

- Вэб сервер болон вэб апп-аа сайтар шалгаж турших хэрэгтэй (жишээлбэл, хэрэглээний явцад сервер уналдахгүй байх ёстой).
- Вэб апп-ийн ажиллахад шаардлагатай бүх бусад файлууд таны оруулах сүүлийн хувилбарын төслийн хавтсан дотор орсон байх ёстой.
 - Эдгээрт өгөгдлийн санг суулгаснаар холбоос үүсэх зургийн файлууд ч багтана.
- submission.txt файл дүүрэн бөглөгдсөн байдалтайгаар төслийн хавтсанд орсон байх ёстой.
- Фото апп-ийн танилцуулга богино видео бичлэг YouTube эсвэл Vimeo-д байршуулсан байх ба түүний холбоос submission.txt файлд орсон байх ёстой.
- ESLint командыг ажиллуулахад таны JavaScript код шалгуур хангаж, алдаагүй байх шаардлагатай. Дэлгэрэнгүйг "Style" хэсгээс үзнэ үү.

Дээрх заавруудын ихэнх нь Project 7 даалгаварт ч мөн хамаарна. Хэрэв шинэ өөрчлөлтүүд хийсэн (жишээлбэл, шинэ програм импортлох, схемд талбар нэмэх гэх мэт) бол эдгээрийг дахин шалгах шаардлагатай. Тодорхой онцлон заасан хэсгүүдийг дагаж мөрдөх нь маш чухал, эс тэгвээс бид таны вэб апп-г ажиллуулж дүн тавьж чадахгүй байж магадгүй.

1. ``mongod --dbpath /usr/local/var/mongodb`,`
(өгөгдлийн сангийн зам өөр байж болно),
2. ``npm install`,`
3. ``node loadDatabase.js`,` and
4. ``node webServer.js``

(мөн зарим тохиолдолд таны submission.txt файлд тайлбарласан нэмэлт зааврыг дагана).

Загвар (5 оноо)

Энэ оноог лекц дээр хэлэлцсэн MVC загварын зөвлөмжийг дагаж зөв зохион байгуулсан тохиолдолд өгнө. **Таны код DOM-д шууд хандаж удирдсан байж болохгүй.** Нэмж хэлэхэд, таны код болон загвар цэвэрхэн, уншууртай байх ёстой. Илгээхээсээ өмнө ESLint багажийг ажиллуулсан байх хэрэгтэй. ESLint ямар нэг алдаа өгөөгүй байна.

Хүлээлгэн өгөх үр дүн

Стандарт классын даалгавар өгөх системийг ашиглан бүтэн аппликэйшнээ (project8 хавтас дахь бүх зүйл) оруулна уу.

Даалгавраа өгөхөөс өмнө зааврын дагуу төслийн хавтсаа цэгцэлж цэвэрлэхээ мартуузай. Ялангуяа test/ хавтсан доторх node_modules/ хавтсыг устгах хэрэгтэйг анхаараарай.

Хэрэглэгчийн профайлын дэлгэрэнгүй мэдээллийг ашиглалттай хамт өргөтгөх

Стори оноо: 3

Хэрэглэгчийн түүх:

Хэрэглэгчийн дэлгэрэнгүй мэдээллийг үзэх үед хэрэглэгчийн дэлгэрэнгүй хуудсанд дараах зүйлийг тусгасан байх ёстой:

- Хамгийн сүүлд байршуулсан зураг. Зургийн жижиг thumbnail зураг, байршуулсан огноо байх ёстой.
- Хамгийн олон сэтгэгдэлтэй зураг. Зургийн жижиг thumbnail зураг байх ёстой бөгөөд сэтгэгдлийг тоо байх ёстой.
- Эдгээр зургуудын аль нэг дээр дарснаар таны өмнө хэрэгжүүлсэн хэрэглэгчийн зургийн харагдац руу шилжих хэрэгтэй. Зураг нь бүх сэтгэгдэлтэй нь харагдах эсэхийг шалгаарай. Нарийн харагдах байдал нь та Төслийн 5 Нэмэлт Оноо хэсгийн фото stepper хэрэгжүүлсэн эсэхээс хамаарна.

Зөвлөмж:

- Та өөрийн серверийн API дуудлагыг бичих хэрэгтэй болно. Мэдээллийн боловсруулалт нь урд талын хянагч руу орох ёсгүй.

Сэтгэгдэл дээр @меншион хийх

Түүхийн оноо: 6

Хэрэглэгчийн түүх:

Хэрэглэгч зурагны сэтгэгдэлд хэрэглэгчдийг @меншион хийх, тухайн хэрэглэгчийг @меншион хийсэн бүх зургийг харах боломжтой байх ёстой

- Одоо байгаа хэрэглэгчийн @меншион хийхийг зөвшөөрөхийн тулд сэтгэгдэл нэмэх дэмжлэгийг сайжруулна

уу. Тайлбар бичих талбарт хэрэглэгчдийг сонгох ухаалаг арга байх ёстой (Зөвлөмжийг үзнэ үү).

- Хэрэглэгчид хүчингүй (байгаагүй) хэрэглэгчийг @меншион хийж чадахгүй байх ёстой бөгөөд хүчингүй хэрэглэгчийн хувьд @меншион хийх нь програмыг гэмтээхгүй байх ёстой..
- Хэрэглэгчийг @меншион хийсэн сэтгэгдэл бүхий бүх зургийн жагсаалтыг харуулахын тулд хэрэглэгчийн дэлгэрэнгүй харагдацыг өргөтгөх, эсвэл @меншион байхгүй тохиолдолд боломжийн зүйлийг харуулах.
 - Зургийн жижиг thumbnail зураг. Зурган дээр дарснаар хэрэглэгчийн зургийн хуудсан дээрх зургийн байршилтай холбогдоно
 - Зураг эзэмшигчийн нэр. Эзэмшигчийн нэр дээр дарснаар эзэмшигчийн хэрэглэгчийн дэлгэрэнгүй хуудас руу холбогдох ёстой.

Зөвлөмж:

- React Mentions: <https://github.com/signavio/react-mentions> (React)

@меншион хийхийг сонгосон сэтгэгдэл болон сэтгэгдэл илгээх хооронд ялгаа байдгийг анхаарна уу. Сонголт дээр сэтгэгдэл бичих хэсэгт @меншион тэмдэглэх аргыг бодож үзээрэй.

Та тодорхой зурагтай холбоотой @меншион хадгалахын тулд мэдээллийн санг өөрчлөх шаардлагатай болно. Зурганд @меншион нэмж, @меншион хийсэн хэрэглэгчийн бүх зургийг буцаах API дуудлагатай байх нь ашигтай байж болох юм.

Зургийн харагдах байдлыг хянах

Стори оноо: 4

Хэрэглэгчийн түүх:

Хэрэглэгч зураг байршуулахдаа аль хэрэглэгчид зураг болон түүний сэтгэгдлийг харахыг хязгаарлах боломжтой байх ёстой. Хэсэг нь дараах байдлаар ажиллах ёстой.:

- Хэрэглэгч зураг байршуулах үед тухайн зургийг харах боломжтой хэрэглэгчдийн жагсаалтыг (хуваалцах жагсаалт) зааж өгөх сонголтыг өгөх ёстой. Жагсаалт заагаагүй бол (жишээ нь, хэрэглэгч жагсаалт тодорхойлох сонголтоос татгалзвал) зураг нь хүн бүрт харагдах ёстой.
- Хэрэв зураг нь түүнтэй холбоотой хуваалцах жагсаалттай бол зөвхөн тухайн зургийн эзэмшигч болон жагсаалтад байгаа хүмүүс зургийг харах боломжтой. Хэрэв тухайн хэрэглэгч хуваалцах жагсаалтад ороогүй бол нэвтэрсэн хэрэглэгчдэд үзүүлэх зургийн тоонд үүнийг харуулах ёсгүй.
- Хуваалцах жагсаалт хоосон байвал зөвхөн тухайн зургийг эзэмшигч л харж болно. Энэ нь "жагсаалт заагдаагүй(no list is specified)" -ээс ялгаатай гэдгийг анхаарна уу.

Зөвлөмж:

- Зургийн холбогдох зөвшөөрлийг хадгалахын тулд та схемээ өөрчлөх шаардлагатай болно.
- Та зөвхөн хэрэглэгч үзэх зөвшөөрөлтэйг нь буцааж өгөхийн тулд зураг буцаах асуулгыг өөрчлөх шаардлагатай (жишээ нь, front end хянагч (controller) дахь зургийг шүүж болохгүй).

Үйл ажиллагааны feed

Стори оноо: 4

Хэрэглэгчийн түүх:

Хэрэглэгч зураг хуваалцах сайт дээр болсон үйл ажиллагааны жагсаалтыг харах боломжтой. Хэсэг нь дараах шаардлагыг хангасан байх ёстой.

- Toolbar дээрх "Үйл ажиллагаа" гэсэн товчлуур нь хэрэглэгчийг тухайн вэб сайт дээр болсон хамгийн сүүлийн 5 үйлдлийг он цагийн урвуу дарааллаар (хамгийн сүүлийнх нь дээд талд) харуулсан харагдуулна. Үзүүлсэн үйлдлүүд нь өвөрмөц байх албагүй гэдгийг анхаарна уу (жишээ нь, 5 зураг байршуулах үйл ажиллагаа байж болно).
- Үйл ажиллагаа бүр нь үйл ажиллагааны огноо, цаг, үйл ажиллагааг гүйцэтгэсэн хэрэглэгчийн нэр, үйл ажиллагааны төрөл (жишээ нь: Зураг байршуулах, шинэ сэтгэгдэл гэх мэт) болон үйл ажиллагааны тодорхой мэдээллийг дараах байдлаар харуулах ёстой.
 - Зураг байршуулах - зургийн жижиг thumbnail зургийг харуулах
 - Сэтгэгдэл нэмэх - сэтгэгдэл үлдээсэн зургийн жижиг thumbnail зургийг харуулах
 - Хэрэглэгч бүртгүүлэх - нэмэлт мэдээлэл байхгүй
 - Хэрэглэгч нэвтэрч байна - нэмэлт мэдээлэл байхгүй
 - Хэрэглэгч гарах - нэмэлт мэдээлэл байхгүй
- Үйл ажиллагааны дэлгэцийн харагдац нь шинэ үйл ажиллагаа гарах үед өөрийгөө шинэчлэх эсвэл шинэчлэх товчлуурыг өгөх ёстой бөгөөд ингэснээр хэрэглэгч шинэчлэлтийг идэвхжүүлнэ.

Зөвлөмж:

- Гүйцэтгэсэн үйл ажиллагааны бүртгэлийг хадгалахын тулд та мэдээллийн сангийн схемийг өргөтгөх шаардлагатай болно. Үүнийг олон янзаар хийж болох боловч MongoDB дээр бичлэгийн газрын (record maps) зураг бүрийг сайн хадгалах шинэ объект үүсгэх арга юм. Үйл ажиллагааны түүхийг тооцоолохдоо үйл ажиллагааны бичлэгийн объектууд дээрх асуулгын тусламжтайгаар хийж болно.
- Та өмнө нь байгаа сэтгэгдэл, зургуудын тэжээлд оруулгатай байх шаардлагагүй (in the seeded database)

Сэтгэгдэл, зураг, хэрэглэгчдийг устгах

Стори оноо: 4 оноо

Хэрэглэгчийн түүх:

Хэрэглэгч сайт дээрх өөрийн эзэмшдэг аливаа зүйлийг устгах боломжтой байх ёстой:

- Түүний аль ч зураг
- Түүний хийсэн ямар нэгэн сэтгэгдэл (үүнд бусад хэрэглэгчдийн зураг дээр бичсэн сэтгэгдэл багтана)
- Бүх хэрэглэгчийн бүртгэл. Хэрэв хэрэглэгч өөрийн бүртгэлээ устгахаар шийдсэн бол эцсийн анхааруулга гарч, дараа нь хэрэглэгч нэвтэрч, мэдээллийн сангаас түүний бүх мэдээллийг устгана. Энэ нь та хэрэглэгчийн объектыг устгахаас гадна энэ хэрэглэгчийн үүсгэсэн бүх зураг, сэтгэгдлийг устгах хэрэгтэй гэсэн үг юм. Үндсэндээ бид хэрэглэгчийг анхнаасаа хэзээ ч байгаагүй гэж үзэх болно. Хэрэгжүүлсэн бусад түүхүүдэд нэмсэн бусад зүйлсийг (жишээлбэл, үйл ажиллагаа, харагдах байдал, меншион гэх мэт) мөн тусгах ёстой.

Зөвлөмж:

- Хэрэглэгч янз бүрийн объектыг устгах үед та мөн устгах шаардлагатай байж болох холбоотой мэдээллийг анхаарч үзэх хэрэгтэй. Энэ нь сервер дээр хийгдэх бөгөөд та холбоо бүхий объектуудыг хэрхэн зөв устгах талаар судалгаа хийх шаардлагатай болж магадгүй юм.
- Хэрэглэгч өөрийн эзэмшдэггүй зүйлийг устгах боломжгүй байх ёстой тул үүнийг бас нягтлаарай.

Зурган дээр “like” дарах

Стори оноо: 4 оноо

Хэрэглэгчийн түүх:

- Facebook-ийн дуртай товчлууртай адилаар хэрэглэгчдэд зураг таалагдах боломжийг олгохын тулд зураг бүрт “like” товчийг хэрэгжүүлээрэй(товчлуур, [icon](#), гэх мэт.).
- “like” товчлуур дээр дарснаар хэрэглэгч зураг дээр like дарсан байх болно (мөн like дарсан товчлуурыг буцаан like-ийг буцаан авахаар дүрслэн харуулна). Нэг хэрэглэгч аль хэдийн таалагдсан зурган дээрх "Like" товчийг дарснаар тухайн зураг таалагдахгүй буюу like-ийг буцааж байх ёстой.
- Зурганд таалагдсан хүний тоо нийт хэрэглэгчдийн тооноос хэзээ ч их байж болохгүй.
- Хэрэв хэрэглэгч аль хэдийн зураг таалагдсан бол тухайн хуудсан дээр тухайн зураг хэрэглэгчдэд аль хэдийн таалагдсан гэсэн харагдац бүхий тэмдэг байх ёстой. Хэрэв хэрэглэгч зураг таалагдаагүй бол энэ дүрсийг арилгах хэрэгтэй.
- Таалагдсан товчлуурын хажууд тухайн зурагт таалагдсан тоог харуулах ёстой. Зурган дээр таалагдсан эсвэл таалагдаагүй бол энэ тоог нэн даруй (харагдахуйцаар) шинэчлэх шаардлагатай.
- Хэрэглэгчийн зургийн хуудсыг таалагдсан тоогоор нь буурах дарааллаар эрэмбэлсэн байх ёстой (хамгийн их таалагдсан зургууд дээд талд байна). Хэрэв зургууд ижил тооны лайктай бол цаг хугацааны тэмдгээр урвуу дарааллаар эрэмбэлнэ үү (хамгийн сүүлд эхнийх). Хэрэглэгч зураг таалагдсан эсвэл таалагдаагүй тохиолдолд үүнийг нэн даруй өөрчлөгддөг байх шаардлагагүй бөгөөд- хуудсыг дахин шинэчлэх хүртэл хүлээгддэг байна.

Зөвлөмж:

- Та зурагтай холбоотой лайк болон лайк үүсгэсэн хэрэглэгчдийг оруулахын тулд схемийг нь шинэчлэх шаардлагатай болно.
- Та лайк бүр хүчинтэй эсэхийг баталгаажуулахын тулд сервер дээр баталгаажуулалт хийх шаардлагатай болно. Мөн уралдааны нөхцөл байдлыг ([race conditions](#)) судалж, энэ нь том хэмжээний системд ямар нэгэн асуудал үүсгэж болзошгүй эсэхийг тодорхойлно уу.

Хэрэглэгчид таалагдсан зургийн жагсаалт

Стори оноо: 6

Хэрэглэгчийн түүх:

Нэвтэрсэн хэрэглэгч дуртай зургуудаа үзэх боломжтой байх ёстой бөгөөд шинэ хуудсан дээр хэрэглэгчийн тусгайлсан дуртай жагсаалт гарч ирэх ёстой (зам: /favorites). Энэ хуудас нь тухайн хэрэглэгчийн дуртай зургуудын жагсаалтыг харуулахад зориулагдсан байх ёстой. Хэрэглэгчид энэ жагсаалтаас дуртай зургуудаа устгах боломжтой байх ёстой.

Хэрэглэгчид дурын зургийн жагсаалтад харагдах дурын зургийн хажууд гарч ирэх товчлуур дээр дарснаар дурын зургийг дуртай гэж тэмдэглэх боломжтой байх ёстой. Зургийг дуртай зургаар сонгохын тулд дараах шаардлагыг хангах ёстой:

- Хэрэглэгчийн зургийн жагсаалтад тухайн зургийг дуртай болгох товчлуурыг идэвхгүй болгох хэрэгтэй.
- Энэ зургийг одоогоор хэрэглэгчид таалагдсан буюу дуртай гэж үзсэн тодорхой үзүүлэлт байх ёстой.

Шинэ харагдацын компонент хэсэг дээр хэрэглэгч дараах үзүүлэлтүүдийг хангасан таалагдсан зургийн жагсаалтыг харах боломжтой байх ёстой:

- Зураг бүрийг жижиг thumbnail зургаар дүрсэлсэн байх ёстой.

- Thumbnail зураг дээр дарснаар зургийн илүү том хувилбарыг харуулах модаль цонх [modal window](#) нээгдэнэ (доорх зөвлөмжийг үзнэ үү). Зөвхөн товшилтоор хэрэглэгчийг зөвхөн тухайн зургийг агуулсан шинэ хуудас руу чиглүүлэх нь хангалтгүй юм.
- Модал нь зурагтай холбоотой огнооноос бүрдсэн тайлбарыг агуулсан байх ёстой.
- Жагсаалтаас мэдрэмжээрээ зургийг устгах (жишээ нь, thumbnail зургийн баруун дээд "x" дээр даргах) хялбар арга байх ёстой бөгөөд ингэснээр хэрэглэгч хуудсыг дахин сэргээсэн болон хаягаас гарсан ч гэсэн энэ нь хэрэглэгчийн таалагдсан жагсаалтад цаашид орохгүйгээр байлгах болно.

Өөр дурдах зүйлс:

- Дуртай зургуудын жагсаалтад url зам /favorites (#!/favorites эсвэл #/favorites) хаягаар хандах боломжтой байх ёстой.
- Хэрэглэгч нэвтэрсэн эсэхээс шалтгаалж, дэлгэцийг хажуу талд (sidenav) эсвэл дээд хэсгийн самбараас (toolbar) таалагдсан хуудсанд нөхцөлтэйгээр харуулна.
- Дуртай зургуудын энэ жагсаалт нь олон сессийн турш хадгалагдах ёстой: өөрөөр хэлбэл, та энэ жагсаалтыг хянахын тулд сервер талын мэдээллийн санг ашиглах ёстой бөгөөд хэрэглэгчид өөрсдийн дуртай хуудсанд зочлох үед жагсаалт үүсгэхийн тулд энэ мэдээлэлд хандах ёстой.

Зөвлөмж:

React Hints

- Та <https://github.com/reactjs/react-modal>, <https://mui.com/material-ui/react-modal/>, эсвэл <https://github.com/react-bootstrap/react-overlays> хэрэгцээтэй гэж бодох байх. Бүрэлдэхүүн хэсгүүдтэй номын санг(component with library) ашиглахдаа баримт бичгийг анхааралтай уншина уу.

- Энэ түүхийн томоохон сорилт бол таны зураг хуваалцах аппликейшнд хамаарлыг (dependencies) суулгаж, зөв тохируулах явдал юм.
- `npm install`-ын үед ямар ч **UNMET PEER DEPENDENCY** **<package-name>** нь одоогийн багцаа суулгахаасаа өмнө **<package-name>**-г суулгах шаардлагатай гэсэн үг юм.

Хэрэглэгчдийн шинэ үйл ажиллагааг тэмдэглэдэг хажуугийн цэс

Стори оноо: 3

Хэрэглэгчийн түүх:

Хажуу талын навигацийн цэсэн дэх хэрэглэгчдийн жагсаалтыг өргөтгөн, зөвхөн нэвтэрсэн хэрэглэгчийн биш, бүх хэрэглэгчийн хамгийн сүүлд хийсэн үйлдлийг харуулдаг болгоно уу.

Үйл ажиллагаа гэдэгт дараах зүйлсийг хамааруулна:

Хэрэглэгч (зөвхөн нэвтэрсэн хэрэглэгч биш) бүрийн хийсэн хамгийн сүүлийн үйлдлийг харуулахын тулд хөтөчийн хажуугийн цэс дээрх хэрэглэгчийн жагсаалтыг сунгах ба үйл ажиллагаанд дараах хэсгүүд орно:

- зураг нийтэлсэн ****зургийн дэлгэцний зураг (thumbnail) шаардлагатай****
- сэтгэгдэл нэмсэн
- хэрэглэгчээр бүртгүүлсэн
- хэрэглэгч нэвтэрсэн
- хэрэглэгч гарсан

Дэлгэц автоматаар шинэчлэгдэх ёстой.

Зөвлөмж:

- Одоогийн хэрэглэгч хэн болох, түүний сүүлийн үеийн үйл ажиллагаа, найз нөхдийн жагсаалтыг сүүлийн үеийн бүх үйл ажиллагаатай нь тодорхой ялгахын тулд хажуугийн самбар дээрх хэсгийг дахин шинэчлэн засварлахыг зөвлөж байна.

Зураг таглах

Стори оноо: 10 оноо

Хэрэглэгчийн түүх:

Хэрэглэгч зураг дээр хүмүүсийг таглах боломжтой. Энэ шинэ функц хэрхэн ажиллах ёстойг харуулсан бүдүүвч зураг эдгээрээс тогтоно:

- Хэрэглэгчид тухайн зургийн тэгш өнцөгт хэлбэртэй хэсгийг сонгож, тухайн хэсэгт байгаа хүнийг таних замаар зургийг таглаж таних болно.
- Зурган дээр хулганыг чирж тэгш өнцөгтийг сонгоно. Хэрэглэгч тухайн хэсэг дээр дарах үед тэгш өнцөгт нь сунаж, багасах ёстой. Хэрэглэгч ямар ч чиглэлд чирэх боломжтой байх ёстой бөгөөд чирж дууссаны дараа тэгш өнцөгт харагдах харагдац хэвээр байх ёстой.
- Хэрэв хэрэглэгч тэгш өнцөгтийг зургийн хэмжээнээс гадагш сунгахыг оролдвол та тэгш өнцөгтийг зургийн хэмжээ хүртэл л сунгах боломжтой.
- Сонгосон тэгш өнцөгт нь хэрэглэгчдэд таалагдахгүй бол түүнийг солихын тулд өөр тэгш өнцөгтийг гарган сонгох боломжтой.
- Хэрэглэгч зургийн хэсгийг сонгосны дараа системд бүртгүүлсэн хэрэглэгчдээс нэр сонгож, хэрэглэгчийг таглах боломжтой байх ёстой. Таг нь илгээгдэж, мэдээллийн санд хадгалагдана.
- Нэг зурагт олон таг байж болох бөгөөд тус бүр өөрийн гэсэн тэгш өнцөгт болон хэрэглэгчийн нэртэй байдаг.
- Аппликешн нь зургийг бүх тагыг нь харуулах механизмыг хангах ёстой: таг бүрийн тэгш өнцөгтийг зураг дээр харуулах бөгөөд хулганыг тэгш өнцөгт дээр аваачихад таг нь хэрэглэгчийн овог, овог нэрийг харуулах ёстой.
- Хэрэглэгчийн нэр дээр дарснаар таглагдсан хэрэглэгчийн хэрэглэгчийн дэлгэрэнгүй харагдац руу шилжих хэрэгтэй.

Зөвлөмж:

- **Гүйцэтгэхэд хэцүү тул энэхүү даалгаврын оноо нь 10 болно!** Front end нь аль нэг талбар сонгох модулийг хайж олох, түүнийг ажиллуулах эсвэл өөрийн талбай сонгох механизмыг бий болгох шаардлагатай..
- Зураг бүр нь зурагтайгаа холбоотой байрлал дээр тагтай байх ёстой. Энэ харьцангуй (relative) байрлалыг ашиглан тагуудыг зөв харуулах боломжийг танд олгоно.
- Байршлын тооцоолол нь цонхны гүйлгэх байрлалд мэдрэмтгий биш гэдгийг баталгаажуулахын тулд хуудсыг өөр газар гүйлгэж байх үед зургуудыг таглахыг туршиж үзээрэй.
- Зарим гуравдагч талын номын сангууд (library) танаас зургийн өргөн ба/эсвэл өндрийг өгөхийг шаардаж магадгүй.