

# Drilldex - Functioneel en Technisch Ontwerp

**Student:** Ezra George **Opleiding:** Bootcamp Full Stack Developer (FSD) v4.3 **Inleverdatum:** 16 februari 2026 **Project:** Drilldex

---

## Inhoudsopgave

1. Inleiding .....	Pagina 1
2. Probleemstelling en oplossing .....	Pagina 1
3. Kernfunctionaliteiten (user stories) .....	Pagina 1
4. Use case tabellen .....	Pagina 2
5. System stories .....	Pagina 3
6. Functionele en niet-functionele eisen .....	Pagina 3
7. Domeinmodel (klassendiagram) .....	Pagina 4
8. Sequentiediagrammen .....	Pagina 5
9. Datamodel en relaties .....	Pagina 6
10. Visuele inspiratie .....	Pagina 6
11. Wireframes (handgetekend) .....	Pagina 9
12. Digitale schermontwerpen .....	Pagina 14
13. Figma/Adobe XD link .....	Pagina 18
14. Conclusie .....	Pagina 18

---

## 1. Inleiding

Drilldex is een lokale marktplaats voor producers en kopers van muziekbestanden. Producers (ARTIST) uploaden beats, packs en kits. Kopers (USER) kunnen deze items bekijken, beluisteren, kopen en downloaden. Administrators (ADMIN) keuren uploads goed of af om de kwaliteit te bewaken. De applicatie bestaat uit een React frontend, een Spring Boot backend en een PostgreSQL database. Alle bestanden worden lokaal opgeslagen zodat de applicatie offline beoordeeld kan worden.

## 2. Probleemstelling en oplossing

**Probleem:** Producers hebben moeite om hun digitale audioassets overzichtelijk te beheren en te verkopen. Kopers zoeken een veilige en gestructureerde manier om content te ontdekken en te licenseren. Veel platforms zijn afhankelijk van externe services, wat niet wenselijk is in een lokale beoordelingsomgeving.

**Oplossing:** Drilldex biedt een lokale, rolgebaseerde workflow. Producers uploaden content, admins keuren dit goed, en kopers kunnen veilig afrekenen via een gesimuleerde betaling. Licenties en downloads worden lokaal gegenereerd en opgeslagen.

## 3. Kernfunctionaliteiten (user stories)

1. Als bezoeker wil ik kunnen registreren en inloggen zodat ik toegang heb tot de marktplaats.
2. Als artist wil ik beats, packs en kits kunnen uploaden zodat ik ze kan verkopen.
3. Als user wil ik items kunnen kopen en downloaden zodat ik ze direct kan gebruiken.
4. Als admin wil ik uploads kunnen goed- of afkeuren zodat de catalogus kwaliteit behoudt.

## 4. Use case tabellen

### UC-01 Authenticatie

**Actor:** Bezoeker / User **Preconditie:** Bezoeker heeft een e-mailadres. **Hoofdscenario:**

1. Bezoeker opent login/registratie.
2. Bezoeker vult gegevens in.
3. Systeem valideert invoer.
4. Systeem maakt account aan of logt in.
5. Gebruiker wordt doorgestuurd naar dashboard.

**Alternatieven:**

- A1: Onjuist wachtwoord -> foutmelding.
- A2: Account bestaat al -> doorverwijzing naar login.

### UC-02 Upload content (artist)

**Actor:** Artist **Preconditie:** Artist is ingelogd. **Hoofdscenario:**

1. Artist opent uploadformulier.
2. Artist kiest audio + cover en vult metadata in.
3. Systeem valideert input.
4. Upload wordt lokaal opgeslagen en op "pending" gezet.
5. Admin keurt upload later goed.

**Alternatieven:**

- A1: Bestand ontbreekt -> validatiefout.
- A2: Onjuist formaat -> upload geweigerd.

### UC-03 Kopen en downloaden

**Actor:** User **Preconditie:** Item is goedgekeurd. **Hoofdscenario:**

1. User voegt item toe aan winkelmand.
2. User bevestigt gesimuleerde betaling.
3. Systeem maakt aankoop en licentie aan.
4. User downloadt het item via bibliotheek.

**Alternatieven:**

- A1: Betaling faalt -> bestelling geannuleerd.
- A2: Download ontbreekt -> foutmelding + retry.

## UC-04 Moderatie door admin

**Actor:** Admin **Preconditie:** Er zijn pending uploads. **Hoofdscenario:**

1. Admin opent wachtrij.
2. Admin bekijkt preview.
3. Admin keurt goed of af.
4. Systeem update status en logt actie.

### Alternatieven:

- A1: Preview ontbreekt -> admin wijst af.
- A2: Admin geeft geen reden -> systeem voegt standaardredenen toe.

## 5. System stories

1. Als systeem wil ik uploads valideren zodat ongeldige bestanden niet worden opgeslagen.
2. Als systeem wil ik aankopen loggen zodat users opnieuw kunnen downloaden.
3. Als systeem wil ik role-based access afdwingen zodat alleen admins kunnen modereren.
4. Als systeem wil ik de database seeden zodat de applicatie direct bruikbare demo-data heeft.

## 6. Functionele en niet-functionele eisen

### Functionele eisen (FE)

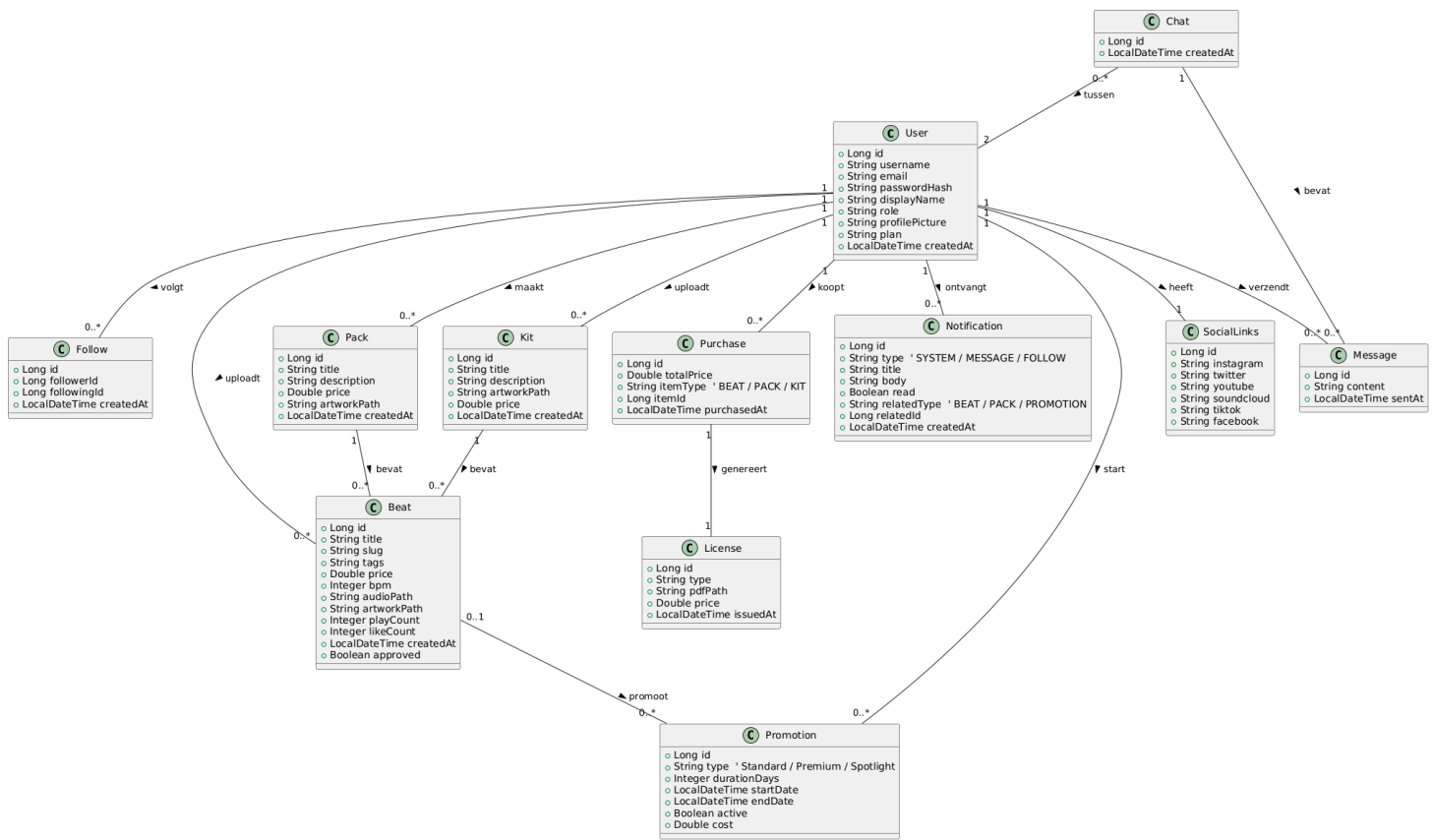
- FE-01 Gebruiker kan registreren met e-mail en wachtwoord.
- FE-02 Gebruiker kan inloggen en uitloggen.
- FE-03 Systeem geeft JWT tokens uit voor sessies.
- FE-04 User kan profielgegevens bekijken en aanpassen.
- FE-05 User kan profielfoto en banner uploaden.
- FE-06 Artist kan beats uploaden.
- FE-07 Artist kan packs uploaden.
- FE-08 Artist kan kits uploaden.
- FE-09 Uploads worden lokaal opgeslagen.
- FE-10 Uploads starten in status "pending".
- FE-11 Admin kan uploads goedkeuren.
- FE-12 Admin kan uploads afkeuren.
- FE-13 Users kunnen beats zoeken.
- FE-14 Users kunnen packs zoeken.
- FE-15 Users kunnen kits zoeken.
- FE-16 Users kunnen filters toepassen (bijv. genre, bpm, prijs).
- FE-17 Users kunnen items previewen.
- FE-18 User kan items liken en unliken.

- FE-19 User kan comments plaatsen op items.
- FE-20 User kan comments verwijderen (eigen comments).
- FE-21 User kan items toevoegen aan winkelmand.
- FE-22 User kan checkout starten.
- FE-23 Checkout gebruikt lokale testbetaling.
- FE-24 Systeem maakt purchase record aan.
- FE-25 Systeem genereert licentie-PDF bij aankoop.
- FE-26 User kan licentie downloaden.
- FE-27 User kan aangekochte items downloaden.
- FE-28 User kan aankopen in bibliotheek bekijken.
- FE-29 Admin kan items markeren als featured.
- FE-30 User kan trending/nieuw/populair lijsten bekijken.
- FE-31 Users kunnen elkaar volgen.
- FE-32 Users kunnen chatberichten sturen.
- FE-33 Systeem maakt notifications aan bij acties.
- FE-34 User kan notifications bekijken en markeren als gelezen.
- FE-35 Artist kan eigen uploads bewerken of verwijderen.

### **Niet-functionele eisen (NFE)**

- NFE-01 React frontend en Spring Boot backend (multi-tier).
- NFE-02 Database is relationeel (PostgreSQL).
- NFE-03 Minimaal 1 one-to-one en 1 one-to-many relatie.
- NFE-04 Authenticatie en autorisatie op rollen.
- NFE-05 Input validatie op DTO niveau.
- NFE-06 Exception handling in de API.
- NFE-07 Lokale opslag voor alle bestanden.
- NFE-08 App werkt volledig offline in lokale omgeving.
- NFE-09 Data-integriteit via constraints.
- NFE-10 CRUD via repositories.
- NFE-11 Tests met JUnit en Spring Boot tests.
- NFE-12 Duidelijke API error responses.
- NFE-13 Max uploadgrootte wordt afgedwongen.
- NFE-14 Loggen van belangrijke acties (moderatie, purchase).
- NFE-15 Pagina navigatie en UI responsief.
- NFE-16 Geen externe cloudservices vereist.
- NFE-17 Veilig omgaan met wachtwoorden (hashing).
- NFE-18 Toegangsrechten consistent in frontend en backend.
- NFE-19 Bestanden blijven beschikbaar na restart.
- NFE-20 Documentatie in PDF is leesbaar en gestructureerd.

## **7. Domeinmodel (klassendiagram)**

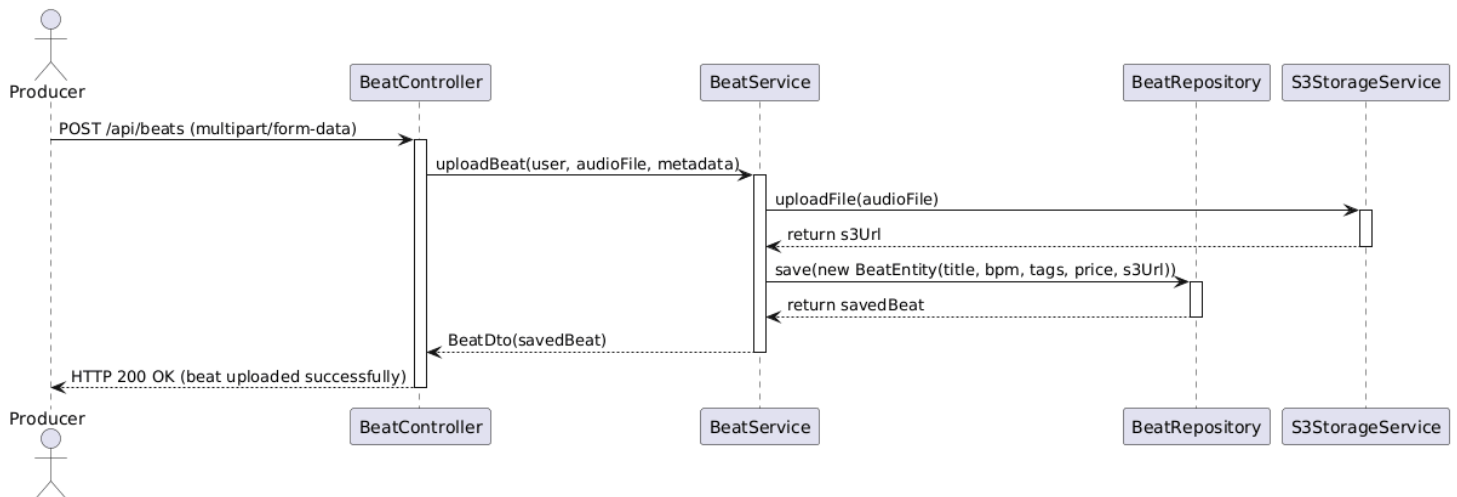


Domeinmodel

## 8. Sequentiediagrammen

### Upload beat (artist)

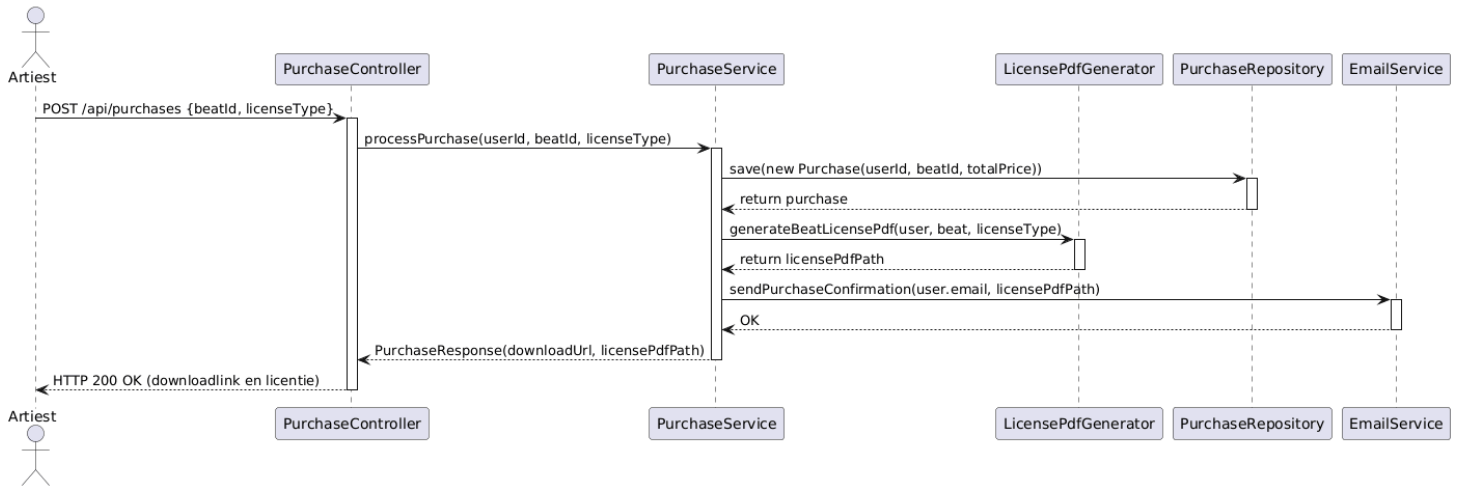
Sequentiediagram - Producer uploadt een beat



Sequentiediagram upload

### Aankoop en licentie (user)

Sequentiediagram - Artiest koopt een beat



Sequentiediagram purchase

> Opmerking: in de actuele implementatie wordt lokale opslag gebruikt voor bestanden.

## 9. Datamodel en relaties (samenvatting)

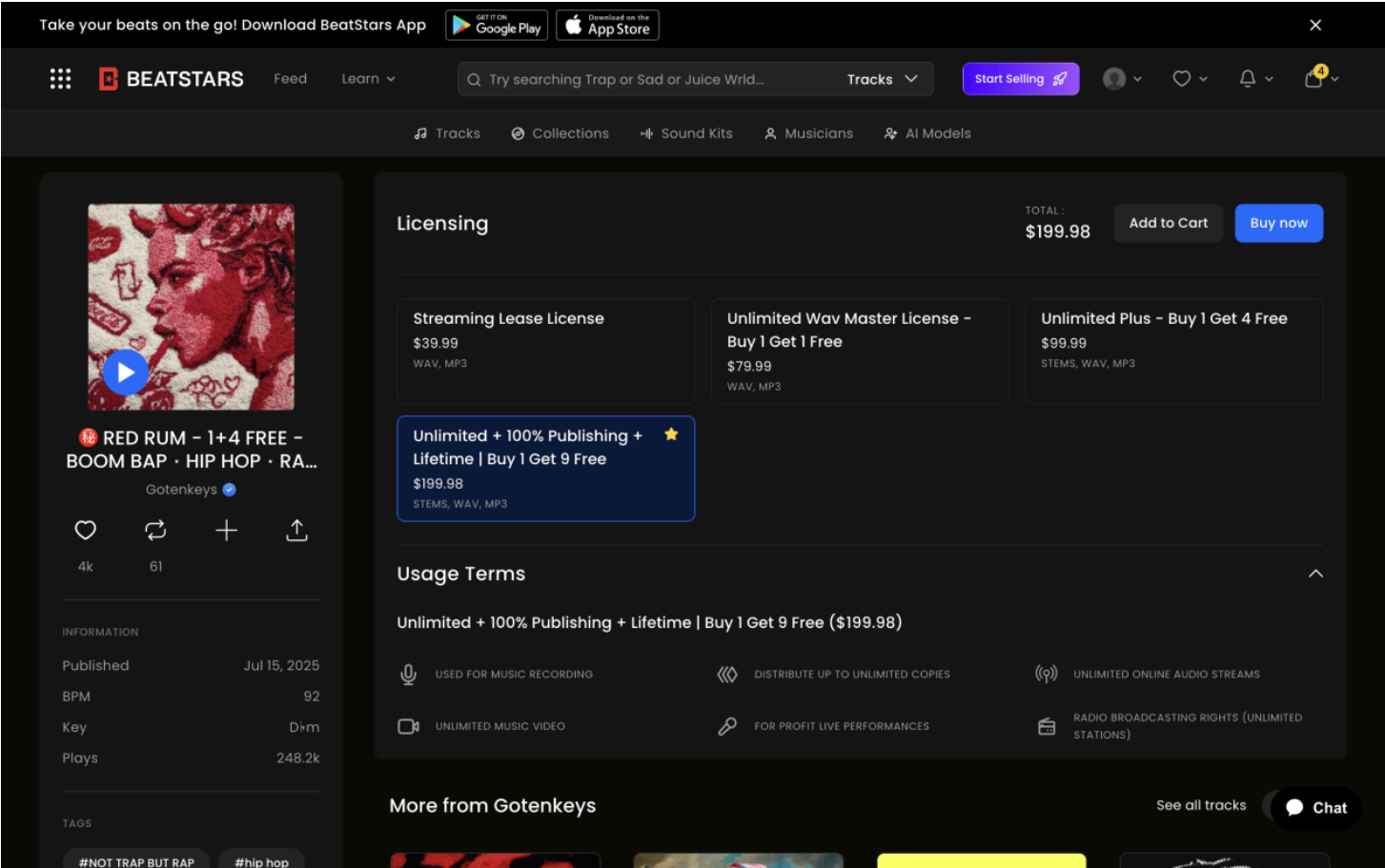
- **User** (id, email, role, profielvelden)
- **Beat** (id, title, price, approved, previewPath)
- **Pack** (id, title, price, approved, previewPath)
- **Kit** (id, title, price, approved, previewPath)
- **Purchase** (id, userId, itemType, itemId, price, licensePath)
- **License** (id, type, price, pdfPath)

Relaties:

- User 1..\* Beat/Pack/Kit
- User 1..\* Purchase
- Purchase 1..1 License


## 10. Visuele inspiratie



Bron 1: BeatStars (licensing layout)




Inspiration 1

Bron 2: Traktrain (catalogus layout)




[LOOPS](#)
[ADVERTISE](#)
[INFO](#)
[TRACKS](#)
[START SELLING](#)
[LOG IN](#)




### FEATURED INSTRUMENTALS




**Curley Fry**  
Need You | Lil Uzi Vert x MexikoDro Type Beat




**Stash**  
LIT UP - 50 CENT X 2000S TYPE BEAT




**Stash**  
RINGIN - 50 CENT X 2000S TYPE BEAT



**Curley Fry**  
Ozone | Gucci Mane x Zaytoven Type Beat




**OCSol**  
Dah Dum



**Stash**  
TOUGH LUCK - 50 CENT X 2000S TYPE BEAT

FREE



**TOUGH LUCK - 50 CENT X 2000S TYPE BEAT**  
\$tash  
R&B • Hip Hop

98 BPM

50 cent type beat 2000s type beat 2000s club beat


AD

\$30.00

SAVE

...

FREE



**Veeze X BabyFace Ray X Lucki Sample Type Beat "Light"**  
Fakez  
Hip Hop

184 BPM


veeze babyface ray lucki sample detroit

\$50.00

SAVE

...

FREE




**are u ok\_448hz**  
xy1va

\$19.00

SAVE

...

FREE



**.il Uzi Vert x MexikoDro Type Beat [157bpi]**  
Curley Fry  
Ambient • Hard • Plugg

157 BPM


SAVE

◀ ▶ 🔊 🔍

Unlimited \$149.99


\$29.99

AD



AIMING FOR PLACEMENTS?

- ☒ Drake
- ☒ Beyonce
- ☒ SZA
- ☒ Ye
- ☒ Jungkook

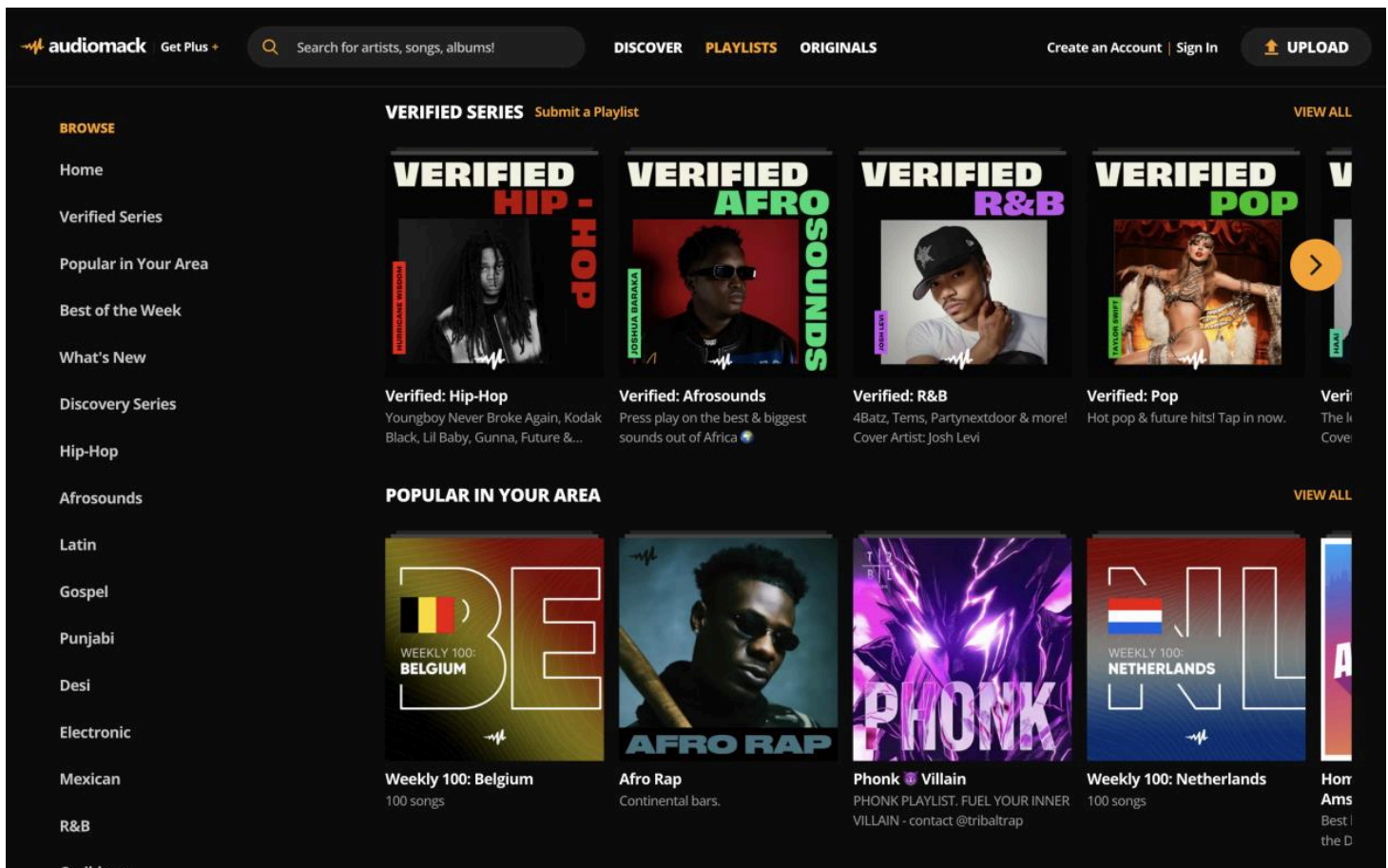


Get Your

## Inspiration 2

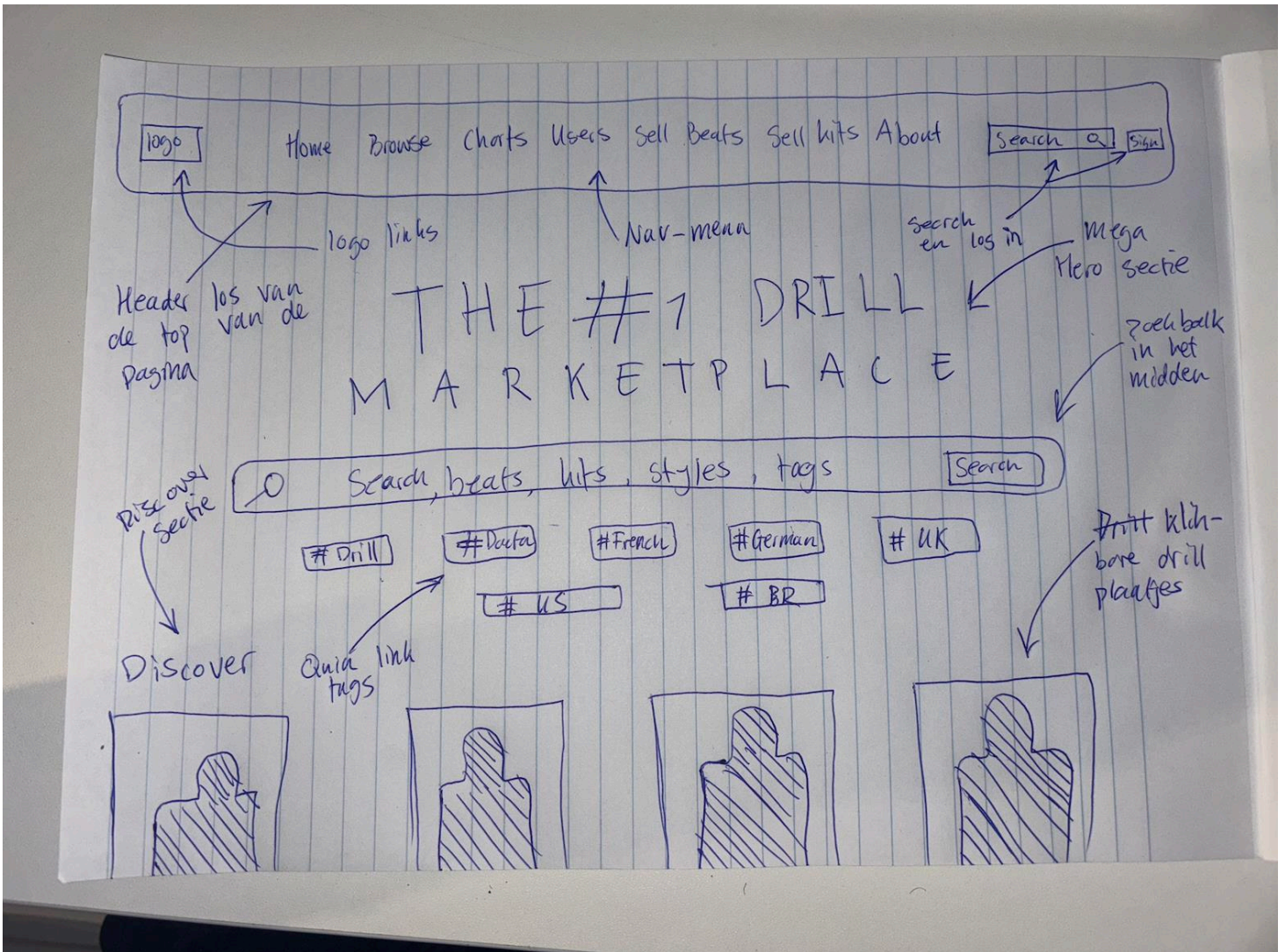
### Bron 3: Audiomack (donkere stijl + discovery)





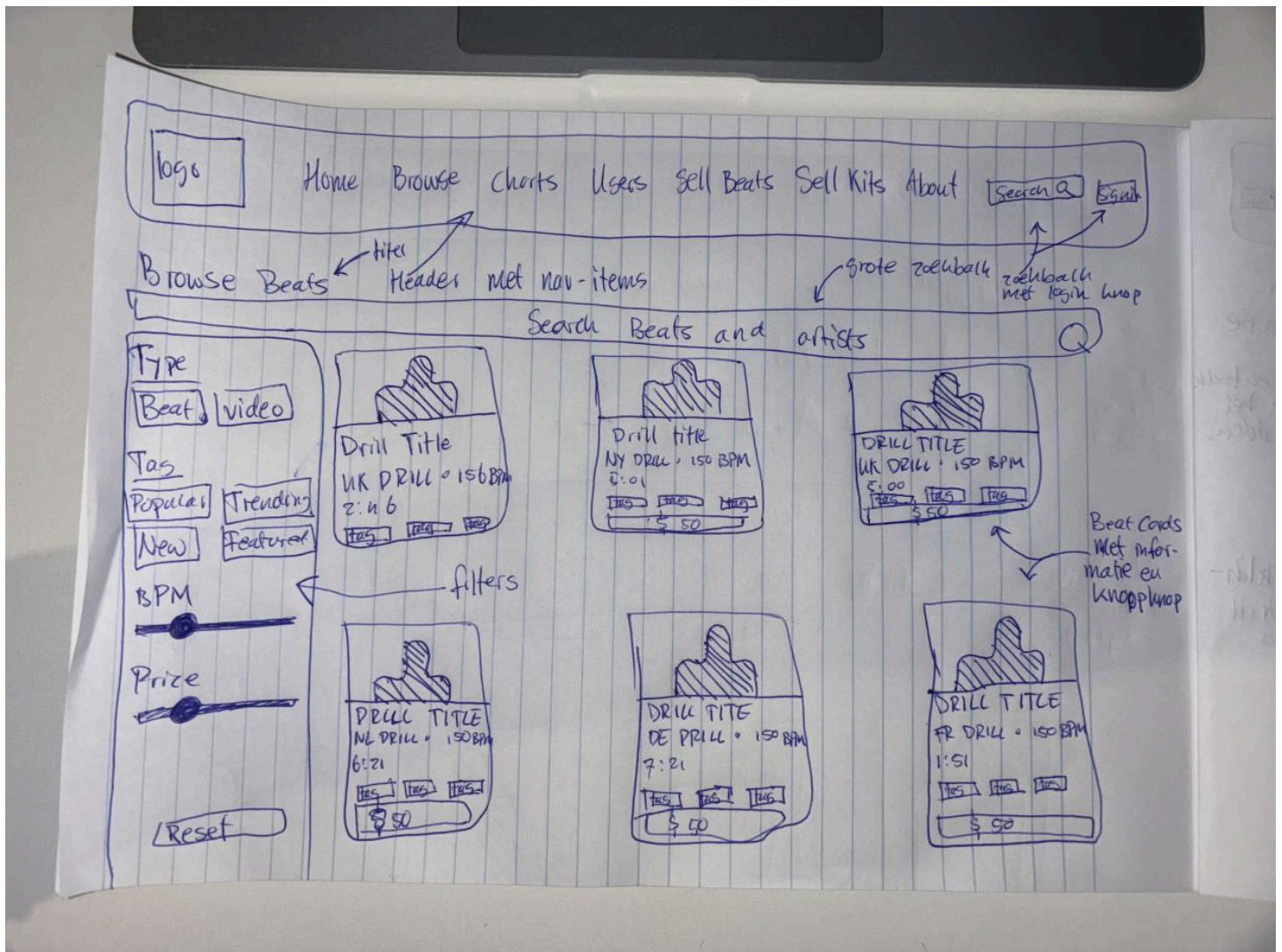
Inspiration 3

## 11. Wireframes (handgetekend)

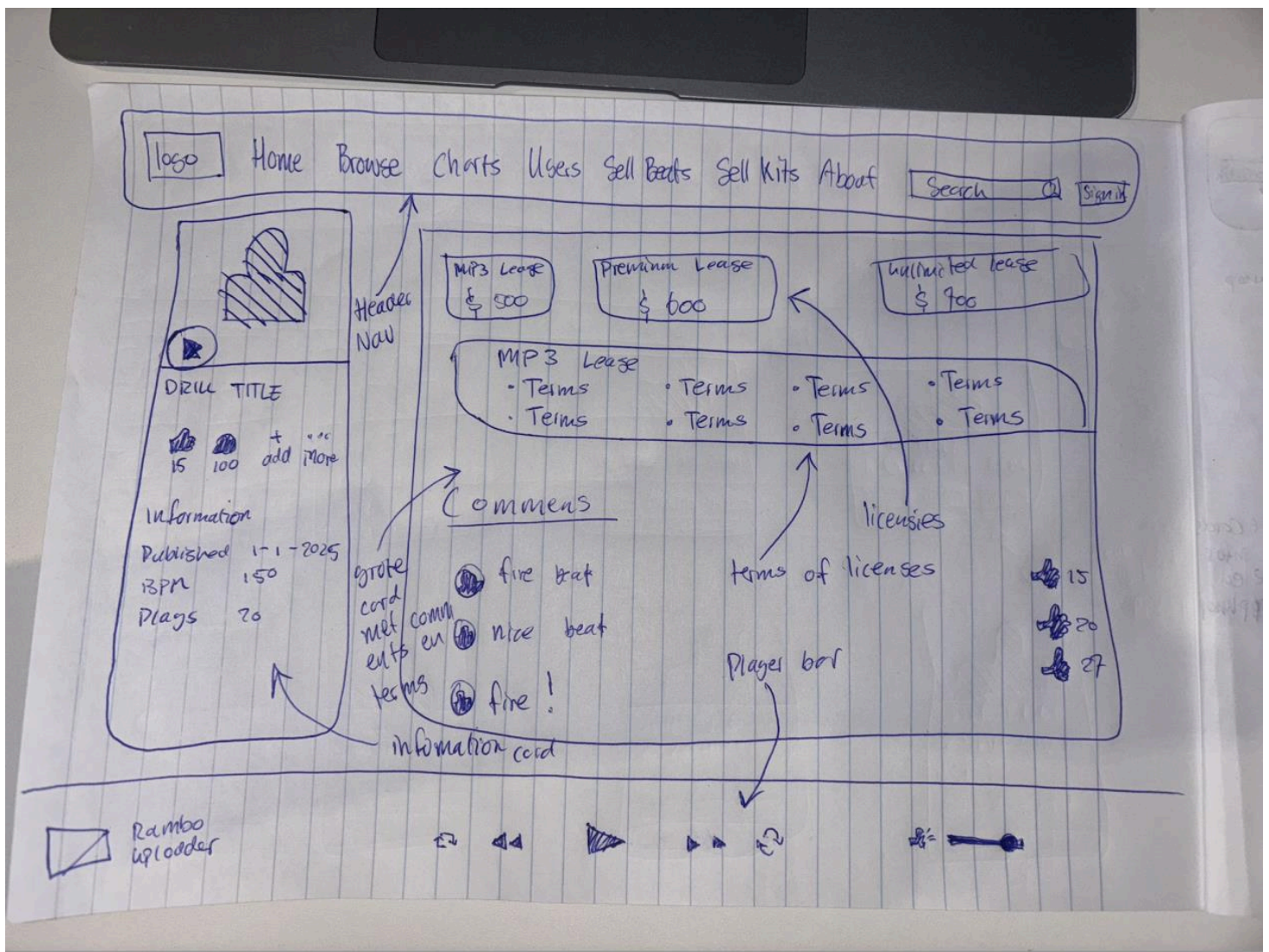


Wireframe landing



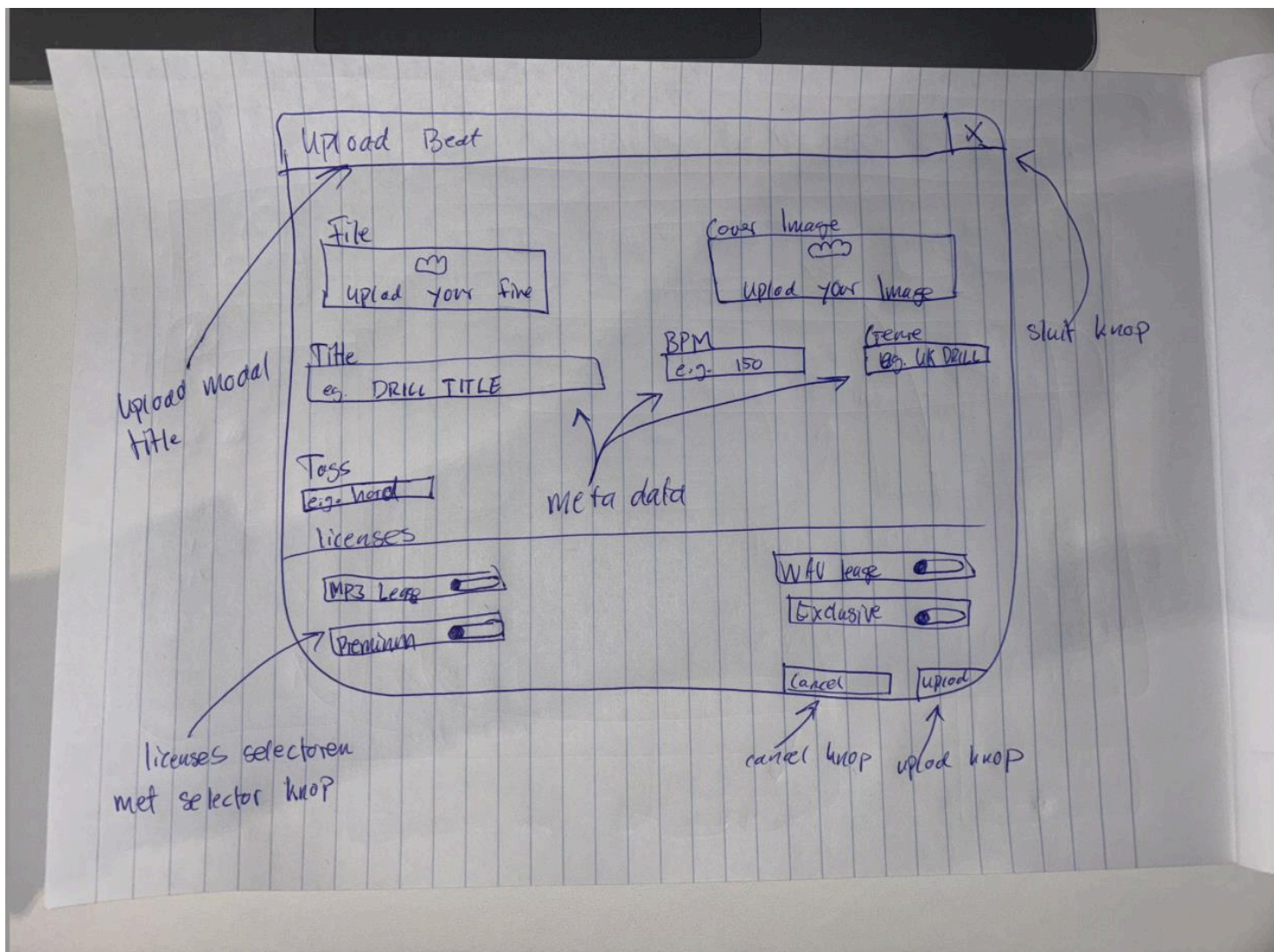


Wireframe library

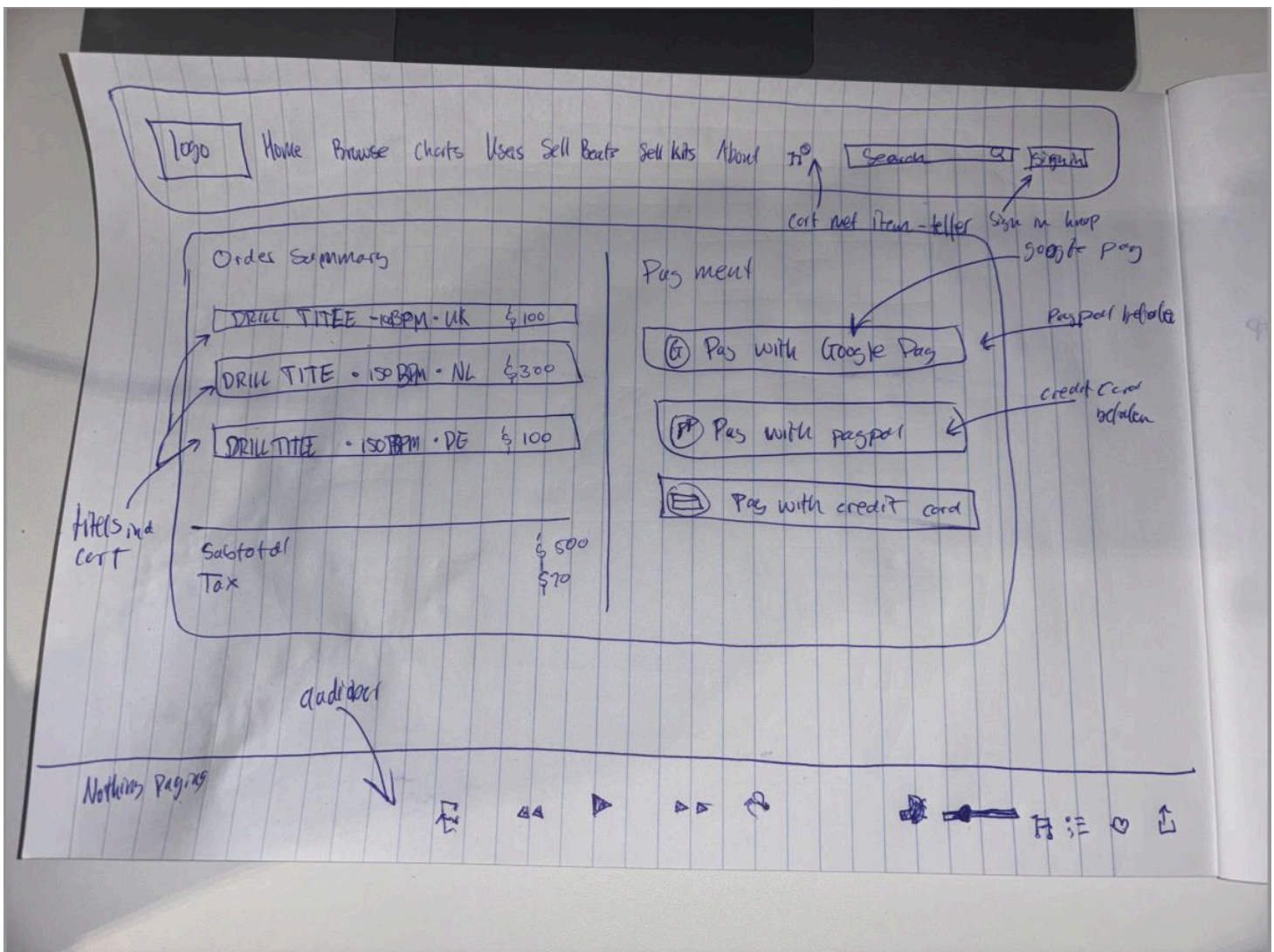


Wireframe detail



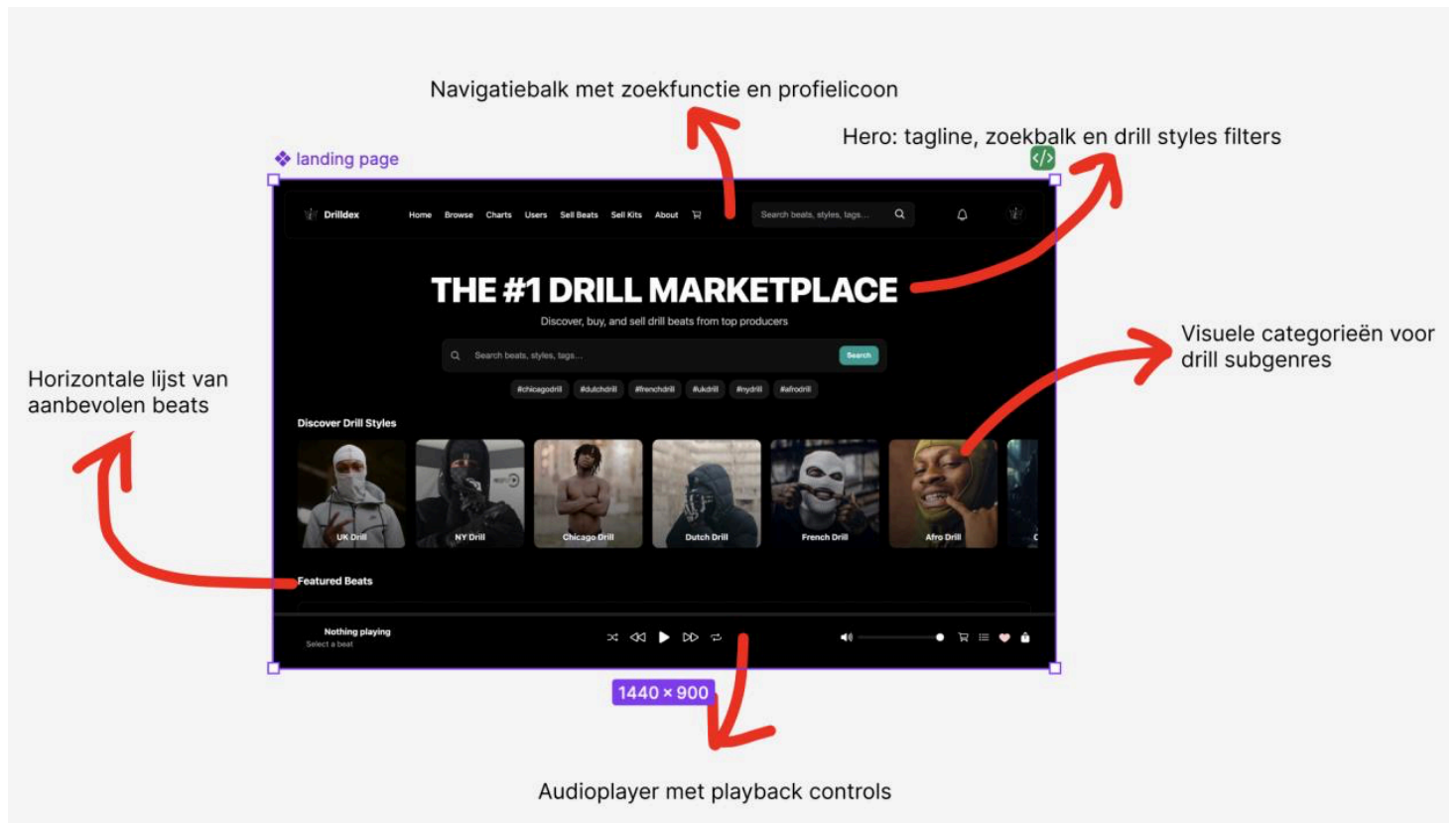


Wireframe upload

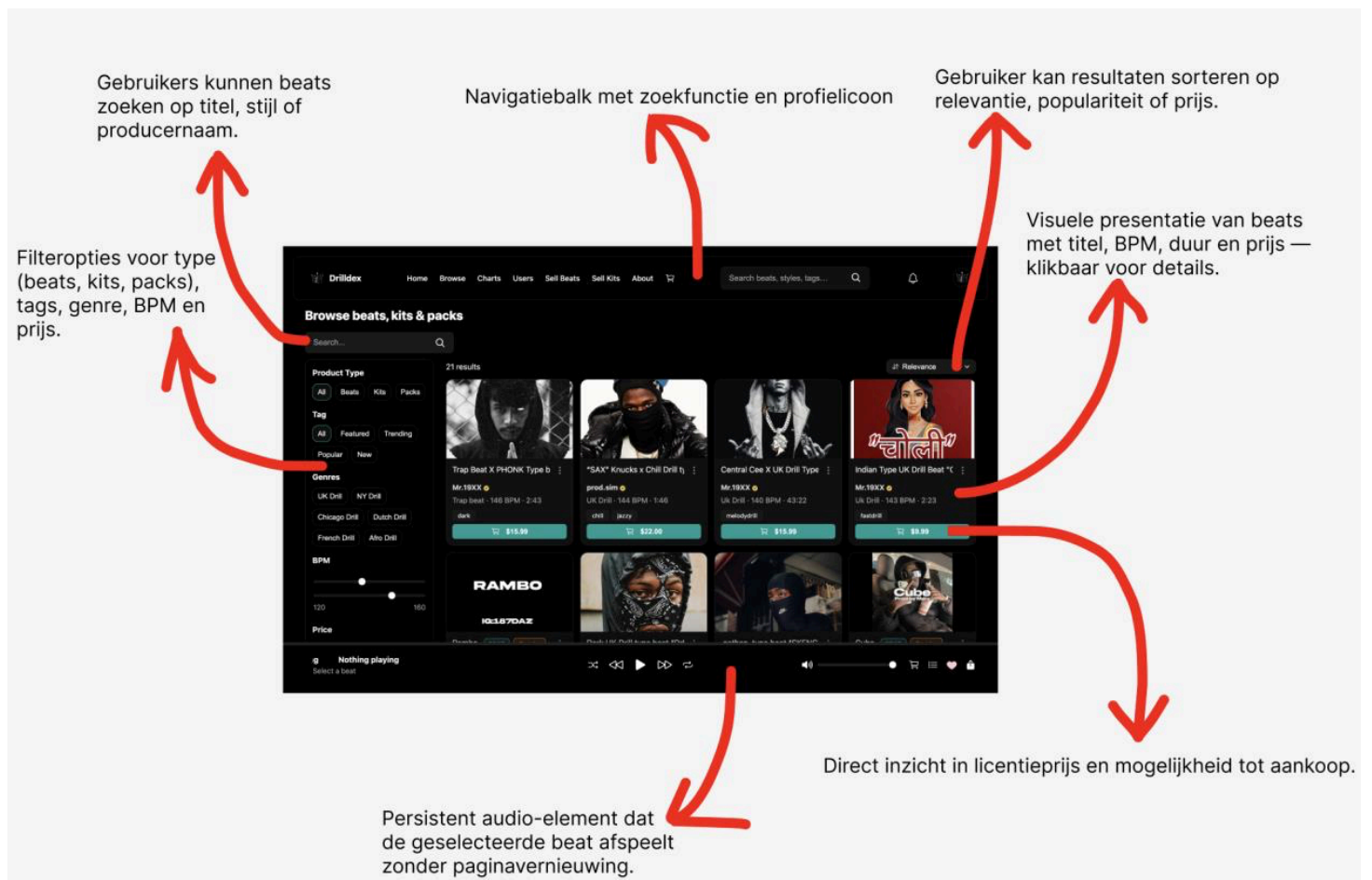


Wireframe checkout

## 12. Digitale schermontwerpen

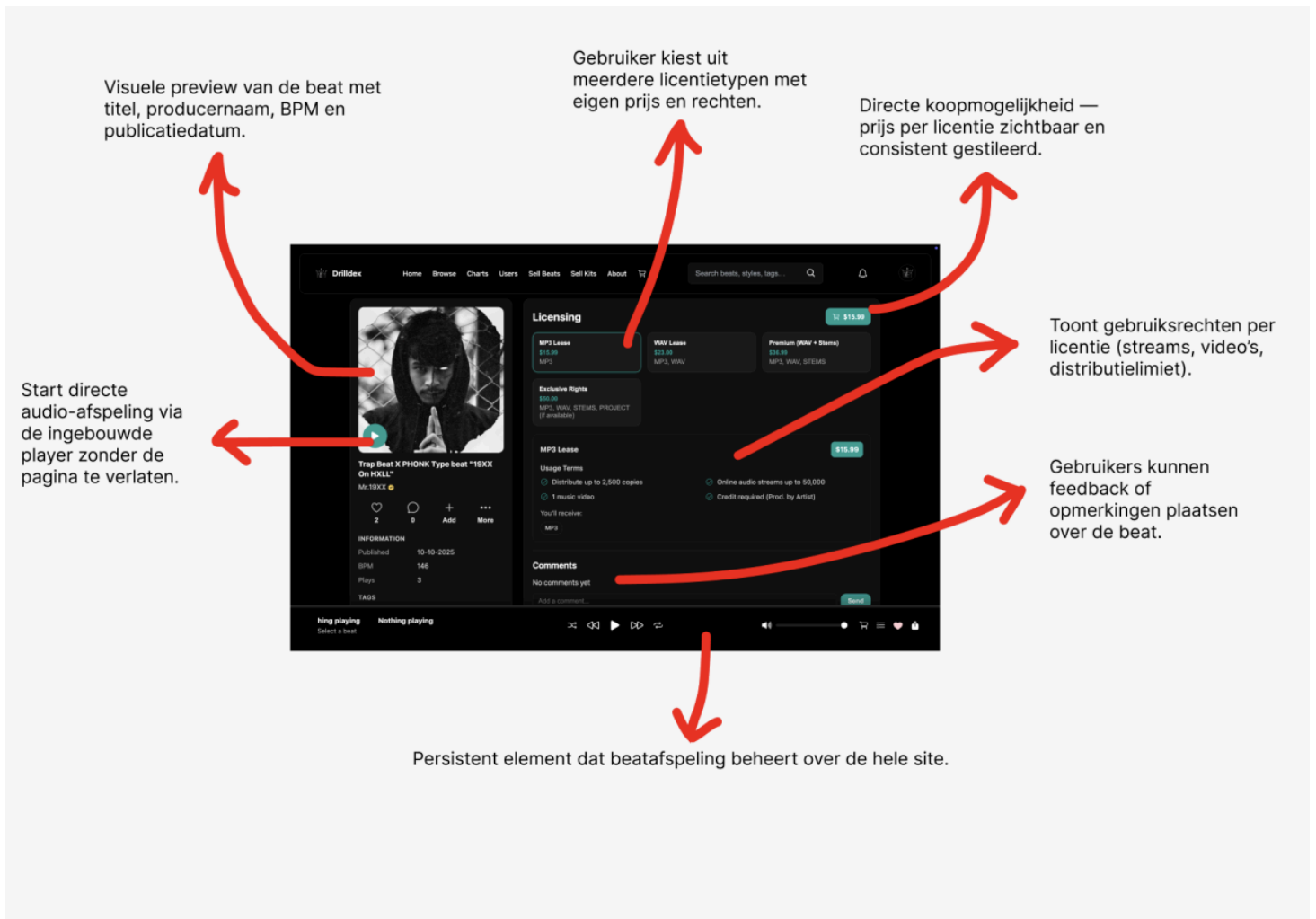


Design landing



Design library





Design detail



# Sell Your Beats

Upload your beats and license them to artists and content creators.

↑ Upload beat

Balance

**\$1,204.67**

Earnings this month








**\$329.15**

Earnings this year

**\$6.451.02**

Total earnings

**\$16.229.37**

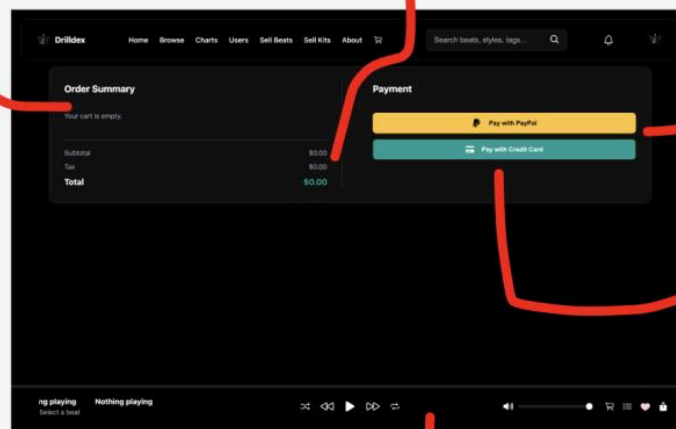
	Beat	Plays	BPM	Key	Tags	Price	Licenses sold	
	After Hours	1.8k	126	Chm	soulful	chill ▾	\$29	29
	Grind Mode	2.3k	140	Dm	piano	soulful	\$35	43
	No Time	976	F3a	G#	underground	trap	\$24	5
	Skylines	1.5k	120	Gm	dark	dark	\$28	18
	Stratosphere	2.1k	128	D#m	hard	hard	\$32	1
	Cold Streets	2.1k	132	Am		hard	\$30	22

Design upload

Toont de huidige inhoud van het winkelmandje inclusief subtotaal, belasting en totaalbedrag.

Het systeem berekent automatisch de totale kosten zodra producten worden toegevoegd.

Gebruiker kan veilig betalen via meerdere betalingsmethoden, met duidelijke visuele knoppen.



Elke knop heeft eigen kleur en icoon voor snelle herkenbaarheid en gebruiksgemak.

Gebruiker behoudt audio-afspeelfunctie tijdens checkoutproces

Design checkout

## 13. Figma/Adobe XD link

Figma link (oorspronkelijk ontwerp):

<https://www.figma.com/design/dLDvX9Qz3ATByaqSU3220t/Functioneel-en-technisch-ontwerp---Drilldex?node-id=0-1>

## 14. Conclusie

Drilldex is ontworpen als een lokale, rolgebaseerde marktplaats met duidelijke upload-, moderatie- en aankoopflows. Het ontwerp sluit aan op de huidige implementatie en voldoet aan de eisen van de bootcamp opdracht.