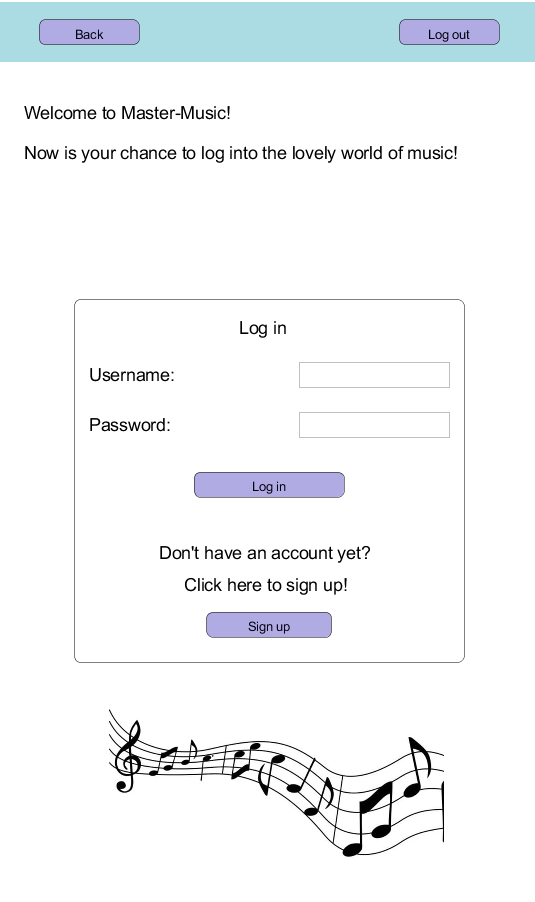
# Digitalt design og udvikling



**Navne**: Cecilia Nygaard, Elisabeth Christensen & Jessie Lorentzen

**Fag**: Digitalt design og udvikling

**Vejledere**: John Gerhard Jensen & Anders Juul Refslund Petersen

**Uddannelse**: H.C. Ørsted Gymnasiet, Lyngby

**Emne**: Udvikling af en musikapp

**Periode**: 22. februar 2021 - 17. maj 2021

Contents

[Digitalt design og udvikling 1](#_Toc72173163)

[Resumé 4](#_Toc72173164)

[Abstract 4](#_Toc72173165)

[Problemidentifikation 5](#_Toc72173166)

[Problemstilling 5](#_Toc72173167)

[Problemanalyse 6](#_Toc72173168)

[Mål for brugerens tidsforbrug på app 6](#_Toc72173169)

[Forventninger til programmet 7](#_Toc72173170)

[Delkonklusion 7](#_Toc72173171)

[Produktprincip 8](#_Toc72173172)

[Krav 8](#_Toc72173173)

[Løsningsforslag 9](#_Toc72173174)

[Løsningsforslag 1 - Musikapp 9](#_Toc72173175)

[Løsningsforslag 3 - Online vendespil app 9](#_Toc72173176)

[Kravmatrix 9](#_Toc72173177)

[Begrundelse for vægtning af point 9](#_Toc72173178)

[Begrundelse for pointgiving 10](#_Toc72173179)

[Delkonklusion (den endelige løsning) 11](#_Toc72173180)

[Produktudformning 12](#_Toc72173181)

[Den oprindelige tanke 12](#_Toc72173182)

[Perfect Pitch Trainer 12](#_Toc72173183)

[Node Name Trainer 13](#_Toc72173184)

[Music Theory Flashcards 13](#_Toc72173185)

[Prototype 13](#_Toc72173186)

[Brugertest af prototype samt tilføjelser til musikappen 18](#_Toc72173187)

[Designkrav 20](#_Toc72173188)

[Produktforberedelse 21](#_Toc72173189)

[Projektstyring 21](#_Toc72173190)

[Endelige krav til produktet 22](#_Toc72173191)

[Software 22](#_Toc72173192)

[Delkonklusion 23](#_Toc72173193)

[Realisering 24](#_Toc72173194)

[Fra tanke til produkt 24](#_Toc72173195)

[Database opbygning 26](#_Toc72173196)

[Sikkerhed 28](#_Toc72173197)

[Brugertest af musikapp 29](#_Toc72173198)

[Konklusion 31](#_Toc72173199)

[Perspektivering 31](#_Toc72173200)

[Bibliography 33](#_Toc72173201)

[Appendiks 34](#_Toc72173202)

[Bilag 1- Spørgeskema 34](#_Toc72173203)

[Bilag 2 - Burndown Chart 39](#_Toc72173204)

[Bilag 3 - Link til indledende brugertest og feedback fra brugere 40](#_Toc72173205)

[Bilag 4: Brugertest af musikapp 41](#_Toc72173206)

[Bilag 5 - Skærmbilleder af den endelige app 42](#_Toc72173207)

[Bilag 6 – Kontrakt 43](#_Toc72173208)

[Bilag 7 – Mødelog 43](#_Toc72173209)

[Bilag 8 – GitHub repository 43](#_Toc72173210)

# Resumé

Denne opgave søger at lave et program der kan besvare problemstillingen *“Hvordan der kan fremstilles et program, som kan hjælpe musikinteresserede i at opnå en bedre forståelse for musikteori og hørelære?”* For at besvare denne problemstilling gør denne opgave først rede for hvad musikteori og hørelære er. Herefter opstilles krav til et program der kan svare på problemstillingen ud fra professionelles udtalelser om hvordan man bedst kan lærer og hvordan man bedst udformer undervisning i hørelære og musikteori.

Den bedste løsning til problemstillingen ud fra disse opstillede krav findes at være en musik app til Android der indeholder tre forskellige hoved funktioner; *Perfect Pitch Trainer* (et minispil der afspiller en tone, og brugeren skal så gætte hvilken (musisk) tone det er, (der er valgmugliheder opgivet i spillet) for at optjene point)*, Node Name Trainer* (et minispil der viser en tegning af en node (vestlig notation), og brugeren skal gætte hvilken node det er (svarmuligheder er opgivet i spillet) for at optjene point)og *Music Theory Flashcards* (Flashcards med generelle spørgsmål omkring musikteori, hvor brugeren kan vende kortet når de er klar og se svaret på spørgsmålet. Herefter kan brugeren selv vurdere hvor svært spørgsmålet var ved at vælge mellem *Easy, Medium* og *Hard*).

Ultimativt blev det fundet at denne Android app i høj grad opfylder de opstillede krav og svarer godt på problemstillingen. Selvom det ikke har været muligt at teste alle krav, da problemstillingen i høj grad fokuserer på en udvikling over længere tid, men det er muligt at sige at langt de fleste krav er opfyldt, og at det udviklede program vil kunne hjælpe musikinteresserede i at opnå en bedre forståelse for musikteori og hørelære.

# Abstract

This assignment seeks to create a program that can answer the question, "How can a program be produced that can help music enthusiasts to gain a better understanding of music theory and auditory learning?" To answer this question, this assignment first explains what music theory and auditory theory are. Then requirements are set for a program that can answer the problem based on professionals' statements on how best to learn and how to best design teaching in auditory learning and music theory.

The best solution to the problem based on these requirements is a music app for Android that contains three different main features. Namely, *Perfect Pitch Trainer* (a mini-game that plays a note, and the user must then guess which (musical) note it is, (there are options given in the game) to earn points), *Node Name Trainer* (a mini-game that shows a drawing of a node (western notation), and the user must guess which node it is (answer options are provided in the game) to earn points) and *Music Theory Flashcards* (Flashcards with general questions about music theory, where the user can turn the card when they are ready and see the answer then the user can assess for themselves how difficult the question was to choose between Easy, Medium and Hard).

Ultimately, this assignment concludes that this Android app essentially meets the set requirements and responds well to the issue. However, it has not been possible to test all requirements, as the problem primarily focuses on development over a more extended period. However, it is possible to say that the vast majority of requirements have been met and that the developed program will help music enthusiasts achieve a better understanding of music theory and auditory theory.

# Problemidentifikation

Musikteori omfatter nodelæsning og analyse af musik, herunder fortolkning af musik og dens historie(Hørsolm Kommunale Musikskole, u.d.), mens hørelære handler om at kunne høre musikken, for eksempel hvordan tonerne passer godt sammen, og om tonerne spilles rent (Hørsolm Kommunale Musikskole, u.d.). En musikinteresseret person, der har et godt indblik i musikteori og hørelære, vil have en klar fordel i forhold til en person, der ikke har så meget viden indenfor området. En større forståelse for musikteori og hørelære vil nemlig resultere i en bedre forståelse for motivet og baggrunden bag musikken, der spilles, og dermed også en bedre formidling af musikken.

Som så mange andre færdigheder kræver læren om musikteori og hørelære træning, og træning er tidskrævende. Det er hverken alle musikinteresserede, som har tiden til at træne dette[[1]](#footnote-1), og det er heller ikke alle, der ved, hvor de skal starte for at lære det[[2]](#footnote-2).

Både musikteori og hørelære er vigtigt for at spille musik, der lyder godt, som er i intonation og for at udtrykke musikken på en måde, der får andre folk til at ville lytte til musikken. Dermed giver det mening at lave en app, der både tager fokus i at lære brugeren musikteori og hørelære.

## Problemstilling

Hvordan der kan fremstilles et program, som kan hjælpe musikinteresserede i at opnå en bedre forståelse for musikteori og hørelære?

# Problemanalyse

I følgende afsnit vil det blive undersøgt, hvordan musikinteresserede mest optimalt kan opnå en større forståelse for musikteori og hørelære ved brug af en app, webside eller lignende.

## Mål for brugerens tidsforbrug på app

Musikteori og hørelære kræver en masse træning at mestre, ved siden af den almene undervisning i et instrument eller i sang. Som beskrevet i problemidentifikationen, er formålet med dette projekt netop at fremstille et program, for eksempel en hjemmeside eller en app, som kan træne musikinteresserede - målgruppen - i disse aspekter af musikundervisningen. Et stort problem i forhold til træning i musikteori og hørelære er som beskrevet i problemidentifikationen, at det er tidskrævende at skulle lære, og mange ved ikke, hvor de overhovedet skal starte. Derfor er det oplagt at fremstille et program, der tager udgangspunkt i, at brugerne træner musikteori og hørelære med appen i 5-15 minutter om dagen, da dette er en overskuelig lille mængde tid pr. dag. Undersøgelser viser desuden, at mange musikinteresserede er mest interesserede i at træne musikteori og hørelære i lignende korte tidsintervaller pr. dag.[[3]](#footnote-3) I forhold til mængden af tid, den gennemsnitlige person bruger på sociale medier om dagen, ses 5-15 minutter også som værende passende, da det estimeres, at den gennemsnitlige dansker mellem 15 og 29 år bruger ca. 60 minutter om dagen på sociale medier (Faktalink, u.d.), mens det for den gennemsnitlige dansker over 30 år er ca. 30 minutter om dagen på sociale medier (Faktalink, u.d.). Sammenlignes denne tid med 5-15 minutters musik træning om dagen på en app, lyder det altså realistisk, at en bruger vil gide at tage 5-15 minutter ud af hverdagen for at træne musikteori og hørelære.

Endnu en fordel ved at bruge 5-15 minutter på en app hver dag, fremfor for eksempel 45-60 minutter to gange om ugen, er, at der er en læringsmæssig fordel i at træne en evne i kortere træningssessions oftere frem for en længere træningssession mindre ofte. En af grundene til dette er, at når man fokuserer på det samme emne i lang tid ad gangen, er sandsynligheden for, at man ikke senere kan huske det, man lærte, mindre, end hvis man kun fokuserer på det samme emne i kortere tid ad gangen (Kwik, 2020). Forskning foreslår desuden, at man har en tendens til primært at huske det første og det sidste, man lærer i en træningssession (Kwik, 2020). Hvis man har meget lange træningssessions, vil man derfor glemme en større del af det, man har lært, da der vil gå længere tid mellem starten og slutningen af træningssessionen. Dermed vil det altså være en fordel at lave et produkt, der fokuserer på kortere træningssessions frem for lange træningssessions, og som opfordrer brugere til at træne med appen hver dag.

## Forventninger til programmet

Når der laves et program, er det vigtigt for brugeren, at programmet er brugervenligt. Hvis programmet ikke er brugervenligt, vil brugeren højst sandsynligt ikke gide bruge tid på programmet, og dermed vil brugeren ikke få en større forståelse for musikteori og hørelære. Derfor er det vigtigt, at programmet er nemt for brugeren at anvende. Her har det desuden en betydning hvilken platform, der benyttes til programmet. Det er vigtigt, at der vælges en platform, som er nem at gå til, især da der lægges op til, at brugeren anvender programmet 5-15 minutter ad gangen. Dette kunne for eksempel være mens, brugeren sidder i et tog, venter i en kø eller lignende. Her ville en oplagt platform være smartphone, da det antages, at de fleste har deres smartphone med, når de er ude on-the-go.

Det er desuden vigtigt, at programmet lærer brugeren musikteori og hørelære samtidig med, at programmet er underholdende og motiverende for brugeren. Hvis programmet ikke er underholdende og motiverende for brugeren, vil brugeren i en mindre grad gide at bruge programmet, og dermed vil brugeren heller ikke lære af programmet. For at gøre programmet underholdende, kunne brugeren for eksempel lave et spil såsom vendespil med musikteori og hørelære, eller/og programmet kunne optælle highscore, hvilket også ville have en motiverende effekt på brugeren.

## Delkonklusion

Ud fra problemanalysen kan det konkluderes, at programmets målgruppe er musikinteressede. Formålet med programmet skal være at træne musikinteresserede i musikteori og hørelære i 5-15 minutter om dagen. En oplagt platform til programmet er smartphone, og det er vigtigt, at programmet er brugervenligt. Derudover er det vigtigt, at programmet virker underholdende og motiverende for brugeren, så brugeren i højere grad vil bruge programmet og dermed lære af programmet. Dette kunne for eksempel opnås ved at lave programmet som et spil med highscore.

# 

# Produktprincip

I dette afsnit vil der blive opstillet nogle krav til programmet samt nogle løsningsforslag. Løsningsforslagene vil derefter blive opmålt i en kravmatrix ud fra kravene, og der vil blive valgt en endelig løsning ud fra kravmatrix.

## Krav

Der vil her blive opstillet nogle krav til programmet, som de følgende løsningsforslag vil blive bedømt efter i en kravmatrix.

Brugervenlighed:

Det skal nemt for brugeren at navigere rundt i programmet. Dermed skal der i programmet tages hensyn til design- og interaktions konventioner, ie. gestaltlove mm. Derudover er det oplagt, at programmet laves til smartphone som platform, da der lægges op til, at brugeren anvender programmet i 5-15 minutter ad gangen, og dermed kan det forventes, at brugeren for eksempel vil bruge appen, mens der ventes på en bus eller et tog, i en kø eller lignende.

Indlæringseffektivitet:

Formålet med programmet er, at brugere skal opnå en større forståelse for musikteori og hørelære over tid. Derfor ville det for eksempel være optimalt, hvis brugeren kunne se en graf over udvikling af highscore eller lignende, så brugeren kan holde styr på sine fremskridt.

Underholdningsværdi:

Brugeren skal være underholdt af programmet, da dette vil medføre, at brugeren i højere grad vil anvende programmet. Dette kan for eksempel opnås ved at formatere programmet som et spil.

Motivering:

Brugeren skal føle sig motiveret ved at bruge programmet, da dette igen vil medføre, at brugeren i højere vil anvende programmet. Dette kan for eksempel opnås ved at inkludere highscore, streaks og lignende som en del af programmet.

Udførlighed:

Programmet skal kunne laves med de givne ressourcer. Dette medfører, at programmet ikke må tage for lang - eller for kort - tid at lave, det må ikke være for dyrt at producere mm. I forhold til penge er det relevant at tænke over, at det vil koste penge at bruge visse databaser.

## Løsningsforslag

Efter en brainstorm er der kommet frem til følgende 3 løsningsforslag, som senere vil blive bedømt ud fra ovenstående krav i en kravmatrix.

## Løsningsforslag 1 - Musikapp

Det første løsningsforslag er en app, hvor brugeren kan spille forskellige spil, der har til formål at øge brugerens evner indenfor musikteori og hørelære. I spillene vil der være et pointsystem, hvor brugeren scorer point, for eksempel ved at svare rigtigt på flest muligt spørgsmål i træk. Den højeste score vil så blive gemt som highscoren. På denne måde vil brugerne kunne konkurrere med med hinanden, hvilket både vil virke motiverende og underholdende for brugeren.

Løsningsforslag 2 - Podcast app

Det andet løsningsforslag er en podcast app, hvor der bliver udsendt et podcast på ca. 10 minutter hver dag. I podcastet vil musikteori og hørelære blive diskuteret, så brugeren dermed vil lære af podcastet. På appen vil det desuden være muligt at gå tilbage til tidligere podcasts, som vil blive gemt under forskellige kategorier, så brugeren kan vælge mellem forskellige emner indenfor musikteori og hørelære, som brugeren vil lære mere om på det nuværende tidspunkt.

## Løsningsforslag 3 - Online vendespil app

Det tredje løsningsforslag er et online vendespil app, hvor kortene vil indeholde forskellige spørgsmål og svar indenfor musikteori og hørelære. Der kunne for eksempel være et kort, der viser en skala, og et tilhørende kort, hvor der står svaret på hvilken skala, det er. Vendespillet foregår online, og dermed kan flere brugere joine det samme spil, eller brugeren kan vælge at spille mod computeren. Der indsamles point sammen ved hvert nyt spil, men pointene går ikke videre til næste spil.

## Kravmatrix

De ovenstående løsningsforslag vil i en kravmatrix blive bedømt ud fra de ovenstående krav. Før selve kravmatrixen vil vægtning af point indenfor kravmatrixen blive begrundet.

### Begrundelse for vægtning af point

Kravene er vægtet i rækkefølge, hvor 5 er det vigtigste krav, og 1 er det mindst vigtige krav. Vi har valgt at vægte indlæringseffektivitet højest, da vores problemstilling tager udgangspunkt i indlæring af musikteori og hørelære. Næsthøjest har vi vægtet underholdningsværdi, da en underholdende app resulterer i, at flere brugere vil bruge appen, og brugerne vil bruge appen i længere tid. Vi har valgt at vægte motivering som den tredje højeste, da motivering igen vil resultere i, at flere brugere vil bruge appen og i længere tid. Vi har vægtet udførlighed med 1, da det er vigtigere, at brugere vil bruge vores app, end hvor svært det er for os at programmere appen.

Table 1: Kravmatrix for de tre forslåede løsningsforslag

|  |  |  |  |
| --- | --- | --- | --- |
|  | Løsning 1:  Musikapp | Løsning 2:  Podcast | Løsning 3:  Online vendespil |
| Brugervenlighed (2) | 5\*2 = 10 | 9\*2 = 18 | 5\*2 =10 |
| Indlærings Effektivitet (5) | 9\*5 = 45 | 3\*5 = 15 | 5\*5 = 25 |
| Underholdningsværdi (4) | 9\*4 =36 | 5\*4=20 | 9\*4 = 36 |
| Motivering (3) | 9\*3 = 27 | 3\*3 = 9 | 5\*3 = 15 |
| Udførlighed (1) | 5\*1 = 5 | 3\*1 = 3 | 9\*1 = 9 |
| **I alt** | 123 | 65 | 95 |

## Begrundelse for pointgiving

**Brugervenlighed**: Podcast app gives flest point i forhold til brugervenlighed, da det rent teknisk set er nemt at navigere rundt i et podcast i forhold til at lære at navigere rundt i en app. Musikapp og online vendespil gives lige mange point, da begge apps vil have samme sværhedsgrad at navigere rundt i.

**Indlæringseffektivitet**: I forhold til indlæringseffektivitet har vi valgt at fokusere på passiv vs. aktiv læring, da vi gav point. Generelt gælder det, at elever lærer bedst ved aktiv læring, hvor eleven aktivt deltager i undervisning, frem for passiv læring, hvor eleven blot modtager undervisningen uden aktivt at deltage (Meylan, u.d.). Her har vi valgt at give podcast app færrest point, da at lytte til podcast vil gå under passiv læring. Musikapp gives flere point end løsningsforslag 3, da det vurderes, at en bruger oftere vil bruge en musikapp end et online vendespil, da det kræver mere af brugeren at starte et helt vendespil, som kan tage flere minutter, i forhold til blot at svare på enkelte spørgsmål eller løse enkelte opgaver, hvor løsningen af en opgave ofte vil tage under et minut.

**Underholdningsværdi**: For underholdningsværdi har vi valgt at give podcast app færrest point, da vi vurderer det som værende mere underholdende at spille et spil/lege med en app aktivt frem for at lytte til et podcast, men dette er jo subjektivt.

**Motivering**: For motivering har vi valgt at give podcast app færrest point, da en bruger ikke får nogen belønning ved at lytte til et podcast (der er ingen high score). Musikapp har vi givet flere point end online vendespil, da brugeren ved online vendespil ikke vil samle highscore sammen over lang tid men blot spille et enkelt spil, som brugeren enten vinder eller taber, og derefter går pointene ikke videre til næste spil.

**Udførlighed**: For udførlighed har vi givet podcast app færre point, da vi selv vil kunne indspille alle podcastne regelmæssigt, og hvis der skal komme nye podcast, skal podcastne også blive ved med at blive indspillet, hvor en app som musikapp eller online vendespil kun skal laves én gang, hvorefter der kun er en lille mængde “vedligeholdelse” af appen derefter. Herudover har vi givet online vendespil flest point, da der er mere at lave i en musikapp med mange funktionaliteter end i et online vendespil.

## Delkonklusion (den endelige løsning)

Ud fra kravmatrix kan det ses, at løsningsforslaget musikapp har fået flest point. På trods af at denne løsning scorer lavere end andre løsningsforslag når det kommer til brugervenlighed og udførlighed, opvejer de point musikappen indhenter fra de andre krav dette. Det kan ses at denne løsning desuden også scorer bedre i forhold til de krav der er vægtet højest. Derfor opfylder dette løsningsforslag de vigtigste krav bedst, og dette vil derfor også være den bedste løsning til at løse vores problemstilling.

Det vil derfor være dette løsningsforslag, der vil blive programmeret, og dermed vil det også være dette løsningsforslag, som resten af rapporten vil tage udgangspunkt i.

# Produktudformning

Løsningsforslaget musikapp er nu blevet valgt. I dette afsnit vil der blive lagt en nærmere plan for udviklingen af appen, så brugeren har mest optimale forhold til læring af musikteori og hørelære.

## Den oprindelige tanke

Den oprindelige idé var at lave et program, hvor det er muligt at oprette sig som en bruger, hvorefter brugeren kan vælge mellem 3 forskellige spil: Perfect Pitch Trainer, Node Name Trainer og Music Theory Flashcards. Disse 3 spil vil blive gennemgået i følgende 3 delafsnit.

### Perfect Pitch Trainer

Direktør for spillemyndigheden, Anders Dorh, har flere gange slået fast, at evnen til at kunne genkende og genspille ens og rene toner kræver øvelse. Denne evne til at kunne genkende og gengive toner samt tonearter kaldes gehør. Omkring 1 ud af 10.000 mennesker har evnen til at kunne bestemme en tonehøjde og gengive den samme tone, uden behov for nogen reference tone (Musikipedia, u.d.). Denne evne kaldes på dansk *absolut gehør* (Musikipedia, u.d.)og på engelsk *perfect pitch*. Selvom meget få har denne evne, så er det påvist, at det ved hjælp af træning er muligt at opnå absolut gehør (Anders Dohn, 2014). Uanset om målet er helt at opnå absolut gehør, eller om målet blot er at kunne spille med en bedre intonation, så vil det være en klar fordel for en musikinteresseret person at øve sig i at kunne genkende toner, og derfor er det oplagt at inkludere træning i absolut gehør i appen.

Derfor har vi i programmet valgt at inkludere en såkaldt *Perfect Pitch Trainer* som et af appens spil. Spillet virker ved, at der trykkes på en “play tone button”, hvorefter der spilles en tone. Brugeren skal herefter komme med et kvalificeret bud på hvilken tone, der spilles. I bunden af skærmen skal der være knapper med de forskellige toner, hvor brugeren skal trykke på den tone, der spilles. Herefter kommer brugeren til en side afhængigt at, om der er svaret rigtigt eller forkert. Derudover får brugeren at vide hvilken tone, der var den rigtige, hvis brugeren svarede forkert. Er tonen gættet korrekt, får brugeren 1 point. Har brugeren svaret forkert, “dør” brugeren, og pointene nulstilles. Med tiden vil brugeren blive bedre til at høre den rigtige tone, og dermed vil brugeren opnå en bedre intonation og være tættere på at opnå absolut gehør.

### Node Name Trainer

Udover en Perfect Pitch Trainer, som har til formål at træne brugeren i hørelære, har vi også valgt at inkludere et spil, der træner brugeren i mere generel musikteori. Dette spil kalder vi Node Name Trainer. Spillet fungerer ved, at der vises en node. Ligesom ved Perfect Pitch Trainer er der et tastatur i bunden af skærmen, og brugeren skal trykke på den knap, der svarer til noden, der bliver vist. Herefter kommer brugeren til en rigtigt- eller forkert-side, alt efter om brugeren har svaret rigtigt eller forkert, og brugeren får at vide hvilken tone, der var den rigtige. Svares der rigtigt, får brugeren 1 point. Svares der forkert, nulstilles pointene. Med tiden vil brugeren dermed blive bedre til at genkende hvilken node, der vises.

### Music Theory Flashcards

En anden måde hvorpå brugeren kan træne musikteori er ved brug af flashcards. Her vil det blandt andet være muligt at træne brugerens viden om tidsperioder, individuelle instrumenter mm. På forsiden af flashcardet ser brugeren et musikrelateret spørgsmål. Ved at trykke på flashcardet, vil det vendes, og svaret kan ses på bagsiden. Herefter skal brugeren selv vurdere hvor svært, brugeren fandt spørgsmålet. Da der ikke som sådan er noget rigtigt svar valgt af brugeren, gives der ikke point. Dette giver blandt andet muligheden for at øve mere komplekse spørgsmål, der ikke har et kort svar. Dette spil kaldes for ‘Music Theory Flashcards’.

## Prototype

For at kunne lave en brugertest af programmet før appen blev programmeret, blev der lavet en prototype af programmet. Denne prototype ses på figur 1-5. Skitserne blev sat ind i MarvelApp (se link og feedback i bilag 3), hvorefter brugertesten blev udført via MarvelApp.

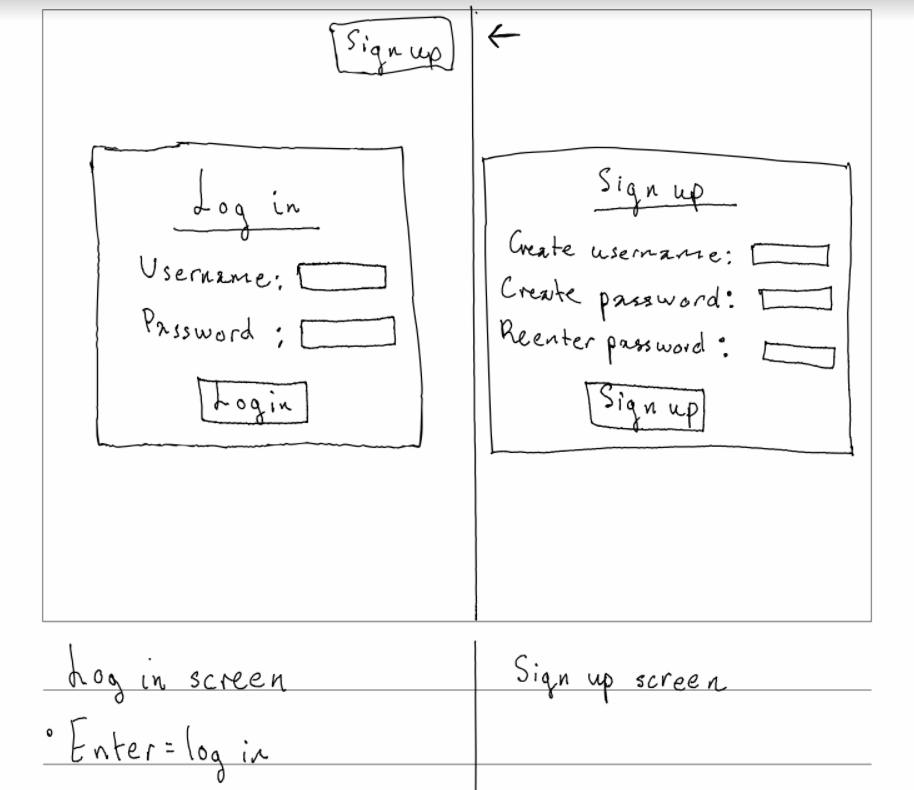


Figure : Skitse af Log-in side og sign-up side

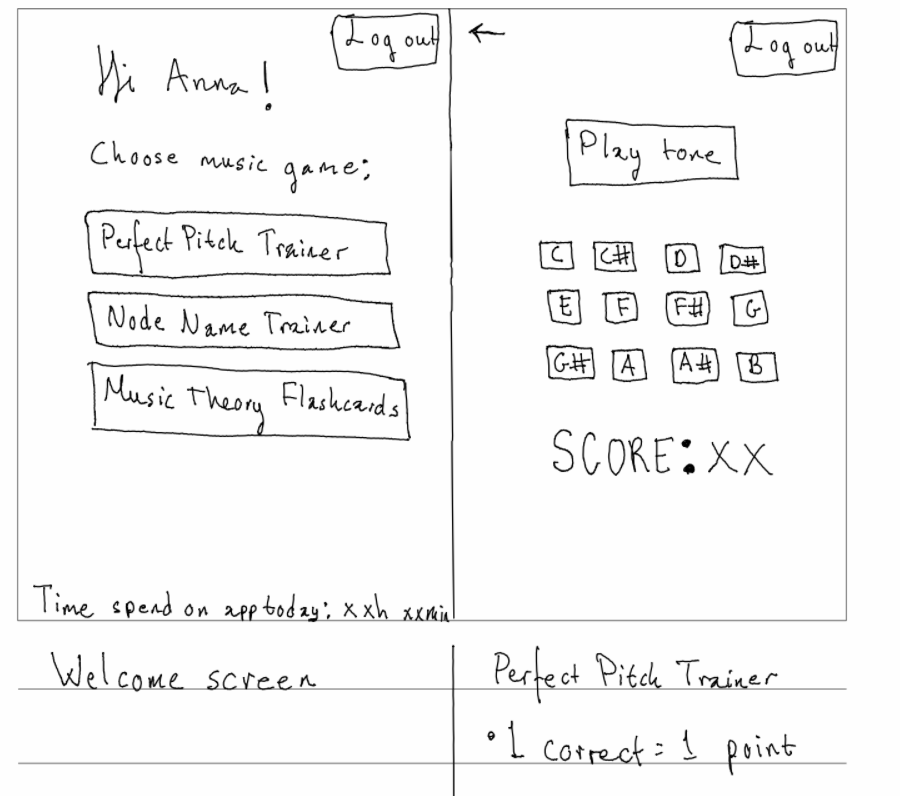


Figure : Skitse af Welcome side og Perfect Pitch Trainer side

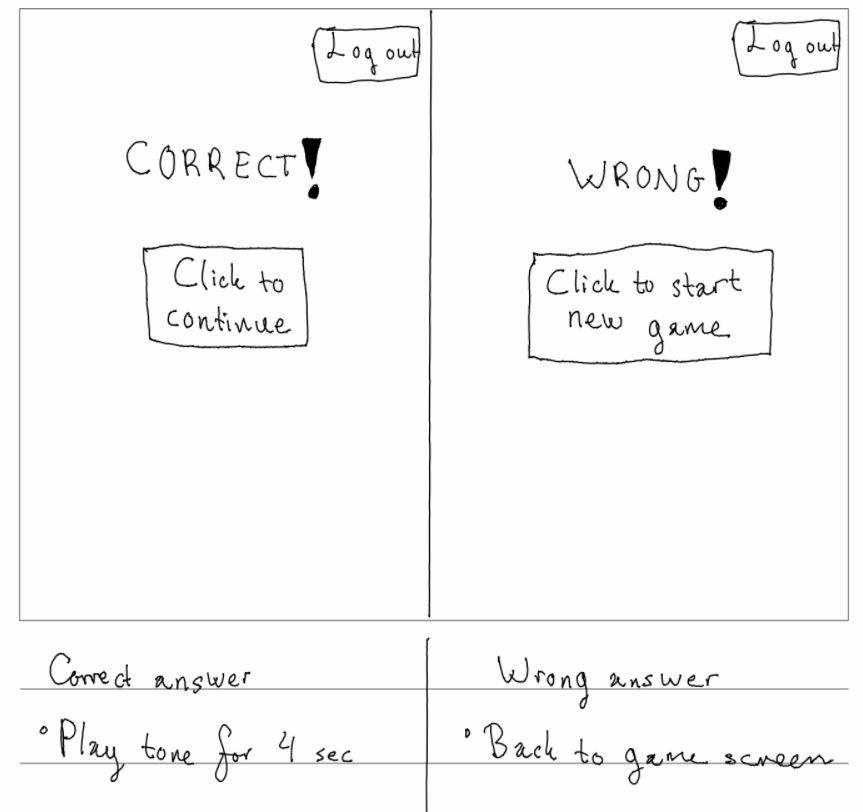


Figure : Skitse af Correct Answer side og Wrong Answer side

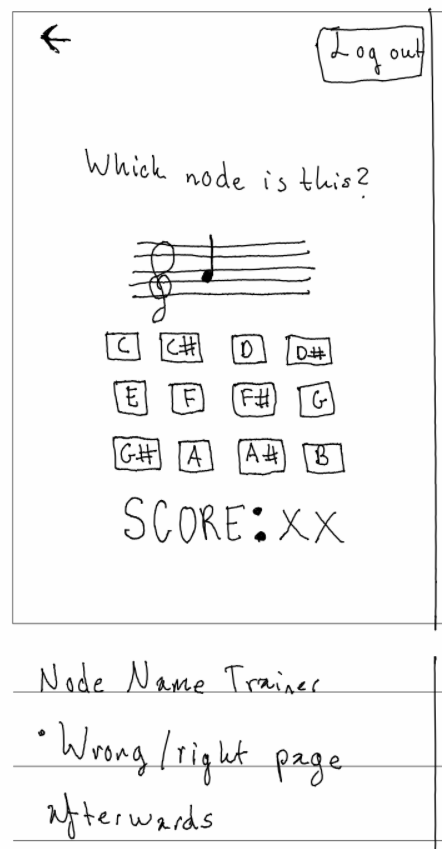


Figure : Skitse af Node Name Trainer Screen

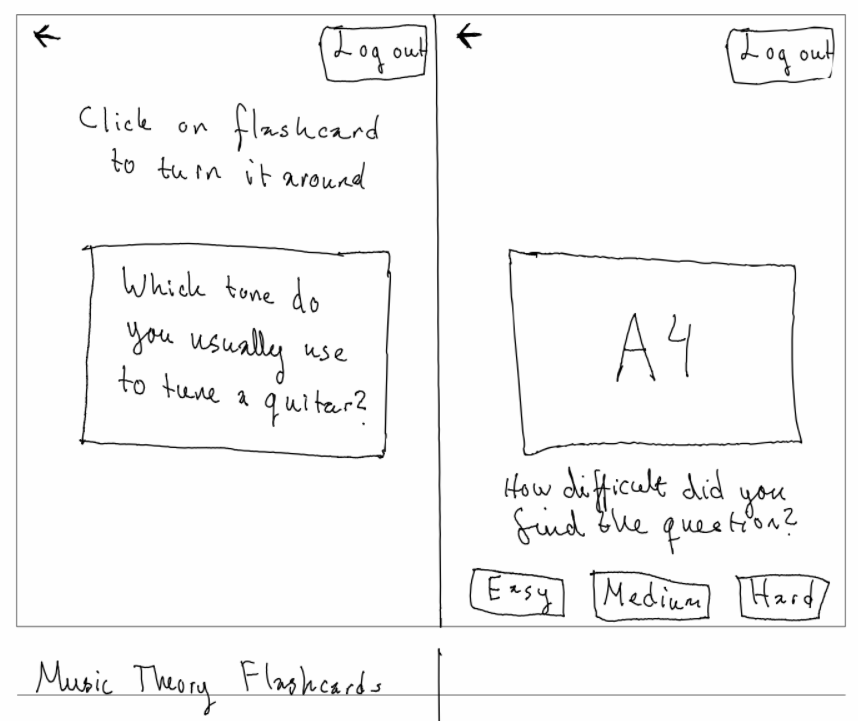


Figure : Skitser af Music Theory Flaschcards skærm 1+2

## Brugertest af prototype samt tilføjelser til musikappen

Ud fra den indledende brugertest lavet via MarvelApp fandt vi frem til, at brugerne ville foretrække, hvis de spørgsmål, de fik stillet i Music Theory Flashcards, var rettet specifikt mod det instrument, de spiller, eller at spørgsmålene ikke var specifikt rettet mod noget instrument.. For at imødekomme dette forslag fra brugerne, tilføjes der når brugeren melder sig til en valgmulighed mellem tre forskellige instrumenter (klaver, violin og guitar). Dette medfører muligheden for, at brugeren vil få stillet forskellige spørgsmål i Music Theory Flashcards alt efter hvilket instrument, de vælger.

Derudover var et forslag fra brugerne, at design eventuelt kunne være rettet mod det instrument, brugeren har valgt, og at der laves et inspirerende design. Dette forslag vil blive behandlet under senere afsnit, “Designkrav”.

I brugertesten var et forslag desuden, at der i stedet for flashcards kunne være en slags quiz, hvor brugeren skulle skrive åbne svar eller vælge mellem svarmuligheder. Denne idé har vores gruppe snakket en del om. Vi har endeligt valgt at beholde spillet “Music Theory Flashcards”, da undersøgelser viser, at flashcards er god for nogle typer af spørgsmål, mens andre typer spørgsmål egner sig bedst til at blive spurgt via quiz-format (Frank, 2020). Dermed ville en udvidelse af appen inkludere, at der vil blive tilføjet et spil ved navn “Music Theory Quizzes”, hvor der vil kunen vælges mellem flere forskellige spil, hvor brugeren skulle svare med åbne svar eller svarvalgmuligheder. Her kunne det så diskuteres, om Perfect Pitch Trainer og Node Name Trainer også skulle tilføjes til denne kategori, eller om de skal blive ved med at være en overordnet kategori i sig selv.

Andre forslag fra brugerne inkluderer, at der tilføjes en “i alt” i footeren af siderne, hvilket vil blive inkluderet i vores app. Desuden var der en kommentar om, at det ikke er helt indelysende, hvordan flashcards virker. Dette kunne blive løst ved at have en “About” side tilføjet til vores program, hvilket ville være en god fremtidig tilføjelse til vores program. Desuden forslog brugerne, at der kunne være niveaudeling indenfor spillene, og at der kunne inkluderes en test, som brugere skulle gennemgå, når de oprettede sig, så brugerne kunne starte på et sværere niveau end begynder, hvis de for eksempel allerede havde spillet musik i mange år. Det kunne desuden væer en god detalje, hvis tonen, der blev spillet i Perfect Pitch Trainer kom fra det instrument, brugeren havde valgt.

## Designkrav

Målet med designet af musikappen er, at designet skal være med til at inspirere brugeren til at blive ved med at spille og øve musik. Derfor er målet, at designet for appen skal være meget frit og flydende, så man som bruger bliver inspireret til at lavet noget selv. Dette gennemføres ved at lave en app, som er meget “fri”, når det kommer til farverne. Der vil være en farvepalette, som vi vil gå ud fra, men der må meget gerne være mange forskellige farver, som spiller sammen. Dermed dannes et meget luftigt og kreativt miljø for vores bruger.

De generelle krav til vores design er dermed:

* Det skal udstråle en kreativ følelse for brugeren.
* Brugeren skal få lyst til at være mere kreativ med brug af vores app.
* Mange farver, som skal spille sammen.

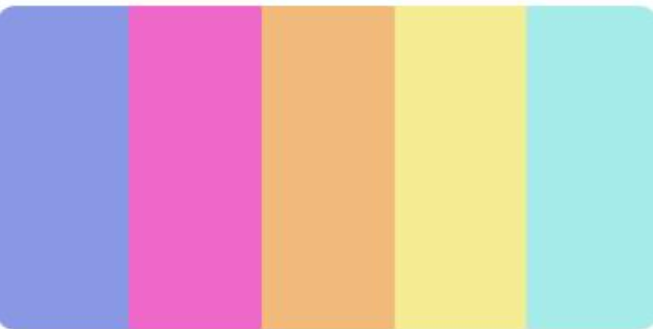


Figure : Farvepalette for vores design

Vi har fundet frem til vores farvepalette ved at lave en brugerundersøgelse i forhold til hvilken farvepalette, der for dem symbolisere frihed og kreativitet mest muligt. Vi er efter denne undersøgelse kommet frem til, at den farvepalette der ses på figur 6 vil være den mest oplagte til vores hjemmeside.

Som man kan se på vores farvepalette, så har vi valgt en variation af farver, som er lidt mere på den “vilde” side, for netop, at skabe et frit og sjovt miljø. Selvom farverne ikke nødvendigvis er ens eller af samme nuancer, så spiller de stadig rigtig godt sammen, for at danne et bedre flow. Der er også blevet snakket om, at det i fremtiden kunne være en mulighed, at når man vælger hvilket form for instrument, som man gerne vil fokusere på, så ville design grundlaget, også afspejles i hvilket instrument, som man har valgt. Som skrevet ovenstående valgte vi at benytte os af disse 5 farver til vores program. Da selve design delen så egentlig begyndte, kom vi frem til, at der var for mange, og at det ville få alt til at se rodet ud. Vi endte derfor med kun at benytte os af den blå og lilla i farvepaletten ovenstående, til designet af vores app.

# Produktforberedelse

I dette afsnit vil der blive redegjort for projektstyring gennem projektet, vores endelige krav til musikappen, samt vores valg af software.

## Projektstyring

Gennem projektet har vi benyttet os af flere forskellige metoder indenfor tidsplanlægning, så vi nemmere gennem projektet har kunnet holde et overblik over hvor langt, vi var kommet, og hvad vi manglede. Herunder har de overordnede metoder været tidsplan, Burndown Chart, mødelog og iterationer på GitHub. Under bilag 2 ses vores Burndown Chart, som er blevet benyttet til at holde overblik over hvor langt, vi var, med udgangspunkt i timerne. Gennem Burndown Chartet har vi holdt et overblik over hvor mange lektioner, vi har brugt på de forskellige overordnede tasks, og hvor mange timer, der var tilbage i projektet. Der er derfor timemæssigt blevet holdt tæt øje med hvor mange timer, der bliver brugt, og hvad der er tilbage. Hvis vi har været bagud i forhold til hvor vi gerne ville være, har Burndown Chartet reflekteret denne forskel. Burndown Chartet har taget udgangspunkt i de timer, vi har haft tilgængelige i skolen. Det tager derfor ikke højde for de timer, vi har brugt uden for skolen, men det har været et godt hjælpemiddeltil at danne os et generelt overblik.

Vi har også haft lavet en almindelig tidsplan (Fisher projektion), som kunne holde styr på de mere grove faser af projektet og produktet. Vi har dog ikke i denne tidsplan anført mindre opgaver, specifikke opgaver som for eksempel at lave en bestemt klasse eller lave bestemte sider. Disse specifikke opgaver er oprettet som tasks i GitHub, så vi let kunne se hvem, der lavede hvad, og hvor langt vi var med hver task (et link til GitHub kan ses i bilag 8).

Vi har desuden holdt en mødelog, som vi både opdaterede når vi startede arbejde i skolen, og når vi var færdige. Denne mødelog kan ses i bilag 7 i vores afleverede fil. Der er også gennem vores mødelog blevet holdt styr på de individuelle personer i gruppen, og hvad alle har lavet for hver gang, så der på den måde også er blevet sikret, at ingen arbejder på det samme på samme tid, hvis der ikke er behov for det. Det har også været en måde at undersøge, om vi overholdt vores tidsplan.

## Endelige krav til produktet

I starten af projektet blev der opstillet nogle generelle krav til vores produkt, som gennem projektet er blevet justeret, så det passede bedre til tidsplanen og behovet, og der er derfor blevet udarbejdet disse krav, som de endelige krav til produktet:

* **Brugervenlig**. Det er vigtigt, at vores program er brugervenligt, så brugeren kan finde ud af at bruge programmet og dermed vil forsætte med at bruge programmet. Dette krav vil vi forsøge at opfylde ved hjælp af et brugervenligt og simpelt design.
* **Inspirerende design**. Et inspirerende design af musikappen vil give brugeren lyst til at blive ved med at øve sig i musikteori og hørelære, hvilket er vigtigt for, at brugeren kan lære optimalt, så brugeren ikke bare stopper med at øve sig på appen efter f.eks. en uge eller to.
* **Motiverende og underholdende**. Det er vigtigt, at vores app virker motiverende og underholdende for brugeren, så brugeren vil fortsætte med at bruge appen. Dette forsøger vi at skabe ved at have flere forskellige spil og ved at inkludere en highscore.
* **Gode muligheder for læring indenfor musikteori og hørelære**. Som beskrevet i vores problemstilling er dette vores primære formål ved vores musikapp.

## Software

Til programmering af musikappen er der brugt Processing som editer, da det er denne editer, vi er blevet undervist i gennem Digital Design og Udvikling lektionerne. Som programmeringssprog er der først blevet brugt Java, hvorefter koden senere er eksporteret til Android i Processing. Dette er gjort på den måde, da det tager længere tid at køre programmet via Android end via Java, når programmet har skullet testes. Grunden til, at vi har valgt at programmere med Java, skyldes at det er det sprog, vi er blevet undervist i. Vi har brugt Android, så koderne kan blive overført til en telefon, som en app. En anden grund til at vi valgte at bruge Android, var også at Android er et åbent source miljø, hvorimod IOS er lukket og kræver, at man køber licens hos dem for at få lov til at komme ind. En bonus ved at bruge Java og Android sammen er desuden, at disse to programmeringssprog minder meget om hinanden, så det er forholdsvis nemt at skifte mellem de to programmeringssprog, når der for eksempel skal eksporteres til Android for at køre appen på en mobil.

Som database er der som det første blevet brugt Access til at lave skitseringerne af selve databasen. Undervejs er Access filen IKKE blevet opdateret, alle ændringer er foregået direkte i databasen. Den rigtige database er så efterfølgende blevet dannet i SQLite. Der er blevet valgt SQLite som database, da der generelt er blevet undervist mere i SQLite databaser fremfor nosql databaser. Udover dette arbejder SQLite godt sammen med Processing, og det en af de databaser, som også er gratis at benytte sig af, som samtidig tilbyder alle de funktioner, som der var behov for. Det eneste problem, der har været med SQLite, er, at den gratis version er klientbaseret og ikke serverbaseret. Dette resulterer i, at brugere i den nuværende version af musikappen for eksempel ikke vil kunne konkurrere om highscore med hinanden, skrive med hinanden; brugerne vil simpelthen ikke kunne interegere med hinanden. Dette vil dog kunne blive løst ved at købe en server til SQLite, hvis vi havde penge til rådighed. En anden ting ved SQLite er, at den API’en, der virker med Java - Bezier SQLite - ikke virker med Android. Dog virker en anden API - Kentai SQLite - med SQLite og Android. Derfor har vi struktureret vores kode på en måde, hvor vi først brugte Bezier som API, da vi programmerede med Java. Derefter kommenterede vi den Bezier-relaterede kode ud og erstattede det med tilsvarende for Kentai, da vi eksporterede til Android. På den måde vil vi forholdsvis let kunne skifte tilbage til Java, hvis vi skulle få brug for dette i fremtiden.

I vores musikapp har vi valgt at lave en såkaldt Perfect Pitch Trainer. Her skulle vi inkludere lyde af forskellige toner, som brugerne skulle kunne identificere. I stedet for at optage alle de toner, vi gerne ville inkludere med rigtige instrumenter, valgte vi at bruge et af Processings egne lydbiblioteker ved navn “Sound”. Dette lydbibliotek kan generere sinuskurver med tonefrekvenser, og dermed kan der genereres lyd, som ved specifikke frekvenser vil svare til toner. Dette lydbibliotek blev valgt frem for andre, da vi i dette lydbibliotek med lethed kan ændre og justere på de komponenter, der udgør en sinuskurve, for eksempel frekvenserne og amplituderne, så den får en lyd, der stemmer overens med de toner, vi gerne vil inkludere i Perfect Pitch Trainer. Principielt er den maksimale frekvens, som Sound-lydbiblioteket kan spille, bestemt af den enhed, programmet kører på, og dennes specifikke højtaler-kapacitet.

## Delkonklusion

Vi har i dette afsnit gennemgået vores overordnede former for projektstyring indenfor tidsplanlægning, hvilket inkluderer tidsplaner, Burndown Chart, mødelog og iterationer i GitHub. Derudover har vi opstillet nogle generelle krav for vores produkt, hvilket inkluderer brugervenlighed, en motiverende og underholdende app, inspirerende design samt god mulighed for træning indenfor musikteori og hørelære. I forhold til software er der valgt at skrive i editeren Processing med programmeringssproget Java, når programmet kører på Windows, og programmeringssproget Android, når programmet kører på Android mobil. SQLite er brugt som database, og API’en til databasen har været Bezier SQLite, når der er kørt Java Processing, og Kentai SQLite, når der er kørt Android Processing. Til at generere toner i Perfect Pitch Trainer er der brugt lydbiblioteket “Sound”, da der med dette lydbibliotek kan genereres sinuskurver af lyd, så der let kan genereres rene toner ud fra sinuskurvernes frekvens.

# Realisering

I dette afsnit vil processen af realisering for vores musikapp blive gennemgået.

## Fra tanke til produkt

Det første, vi gjorde, da vi startede med at skrive kode, var at danne selve UI’et. På den måde kunne vi nemt skabe os et overblik over, hvordan de enkelte funktioner skulle virke for at få programmet til at opfylde de opstillede krav og ligne vores prototype. UI’et består af flere forskellige panels: Klassen “Screen” extender “Panel”, og tilsammen indeholder disse egenskaberne, som de forskellige sider i programmet har. Derfor extender de forskellige sider klassen “Screen”. Derudover er der lavet en klasse, “Application”, som indeholder forskellige funktioner, som siderne skal have. Denne sider extender dermed også “Panel”, da den også skal have de samme egenskaber som siderne. Herudover er der lavet to yderligere klasser, “Header” og “Footer”, som altid bliver vist. Der kan dog ikke trykkes “Back” eller “Log out” på log in siden, og der kan ikke trykkes “Log out” på sign up siden, da brugeren her ikke er logget ind. Til sidst er der lavet en seperat klasse til henholdsvis knapper (to forskellige typer, “Button” og “InstrumentButton”), tekst (“Text”), inputfelter (“Input”), teksten i inputfelter (“StringVariable”) samt en liste af de forskellige udseender af noder (“NodeList”). At have en seperat klasse til for eksempel knapper og tekst er smart, da alle knapper og al tekst på den måde kan formateres ens, så designet af programmet bliver pænere. Havde der ikek været brugt Processing, havde disse klasser været gjort private for at indkapsle det, så andre programmører ikke vil kunne ændre på det.

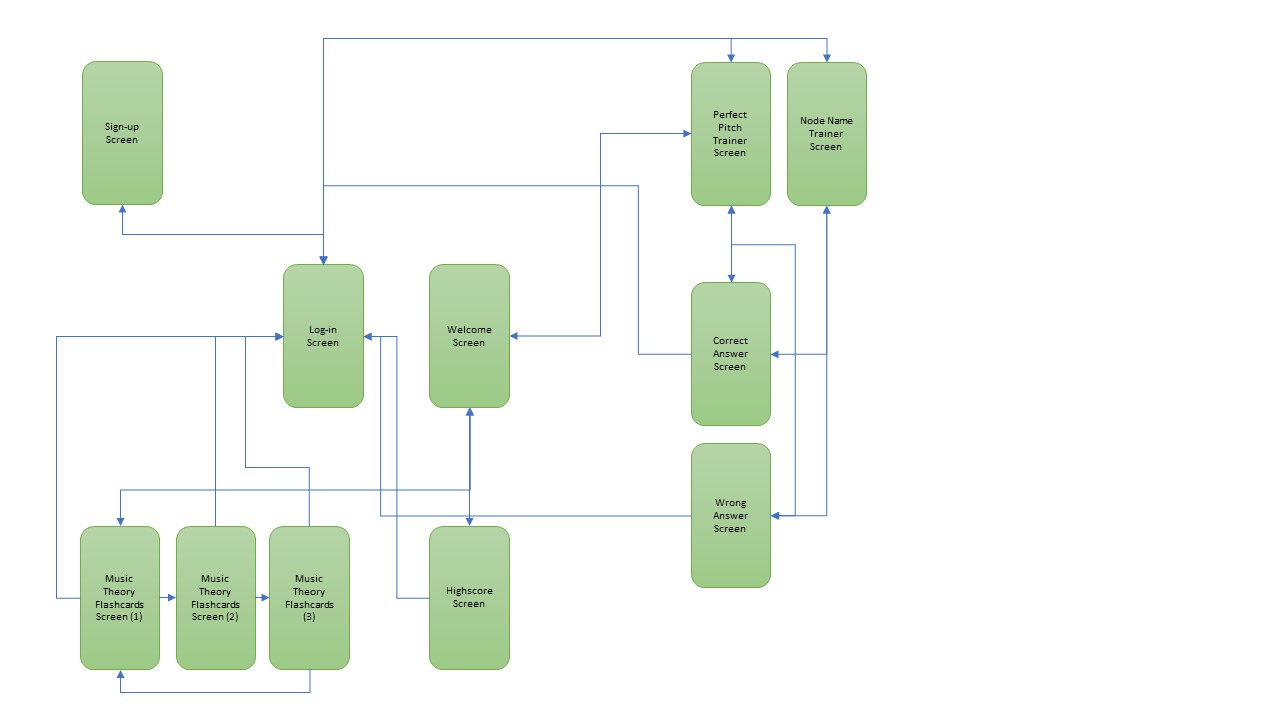
Der er desuden lavet en klasse, “Event”. Denne klasse er kopieret fra en anden bruger, som beskrevet i filen. Funktionen af klassen er med til for eksempel at kunne skifte sider ved at udnytte *bindEvent*. På figur 7 kan der ses et flowdiagram over programmets opbygning.

Figure : Flowdiagram over programmets opbygning

* Når man starter programmet vil man komme ind på *Log-in Screen*. Herfra vil man kunne bevæge sig til enten *Welcome Screen* hvis man allerede har en bruger, eller *Sign-up Screen* hvis man ikke har en profil.
* Fra *Sign-up Screen* kan man kun komme tilbage til *Log-in Screen* (enten ved at trykke “Sign up” eller ved at trykke “Back”).
* Fra *Welcome Screen* kan man komme videre til *Highscore Screen, Perfect Pitch Trainer Screen, Node Name Trainer Screen*, og *Music Theory Flashcard Screen (1)*. Man kan også komme tilbage til *Login Screen*.
* Fra *Perfect Pitch Trainer Screen* kan man komme videre til enten *Correct Answer Screen* eller *Wrong Answer Screen*. Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Node Name Trainer Screen* kan man komme videre til enten *Correct Answer Screen* eller *Wrong Answer Screen*. Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Correct Answer Screen* kan man enten komme videre til *Node Name Trainer Screen* eller *Perfect Pitch Trainer Screen*, afhængigt af hvilken skærm man kommer fra (som i: man vil komme tilbage til den samme skærm man kom fra). Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Wrong Answer Screen* kan man enten komme videre til *Node Name Trainer Screen* eller *Perfect Pitch Trainer Screen*, afhængigt af hvilken skærm man kommer fra (som i: man vil komme tilbage til den samme skærm man kom fra). Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Music Theory Flashcards Screen (1)* kan man komme videre til *Music Theory Flashcards Screen (2)*. Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Music Theory Flashcards Screen (2)* kan man komme videre til *Music Theory Flashcards Screen (3)*. Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Music Theory Flashcards Screen (3)* kan man komme videre til *Music Theory Flashcards Screen (1)*. Man kan også komme tilbage til *Welcome Screen* eller *Login Screen*.
* Fra *Highscore Screen* kan man komme tilbage til *Welcome Screen* eller *Login Screen*

## Database opbygning

Figure 8: Tabel oversigt over databasens opbygning

Databasen, som er en SQL-database, er opbygget af tabeller. Oprindeligt var der lavet en skitse i Access over hvilke relationer, vi ville have i strukturen af denne database. Under udformningen af programmet fandt vi dog ud af at visse ting gav bedre mening på en anden måde, så vi adapterede vores database struktur undervejs, mens vi skrev databasen i SQLite. Den endelige database struktur består af 5 tabeller, som kan ses på figur 8. Vi tog beslutning om at strege e-mailadressen fra users (Bruger) tabellen, da vi ikke rigtigt brugte den til noget. Det kan dog overvejes, om det er brugbart at have en e-mailadresse for brugerne, hvis der skal reklameres med nye funktioner i appen eller for at have et overblik over brugerne (hvilket også kunne være en forsikring mod, at der blev oprettet flere af den samme bruger). Desuden besluttede vi at inkludere to kolonner med highscores for henholdsvis NodeNameTrainer og PerfectPitchTrainer (highscore\_NodeName og highscore\_PerfectPitch), hvor meget tid man daglig har brugt på appen (timeToday) og hvor meget tid man i alt har brugt på appen (timeTotal). For at sørge for at der ikke kunne oprettes brugere med det samme brugernavn, er der i kolonnen ‘username’ tilføjet en UNIQUE constraint, der sørger for at alle brugernavne skal være forskellige.

For at skabe mulighed for at videreudvikle programmet i fremtiden, valgte vi desuden at adskille instrumenterne (instruments) og brugerne (users), således at instrumenterne fik deres egen tabel og id (instrument\_id). På denne måde er der mulighed for at brugere i fremtiden kan have mere end ét instrument knyttet til deres profil. For nu er der dog kun ét instrument knyttet til hver profil.

Spørgsmålene (questions) er struktureret således, at de er afhængige af et instruments id (instrument\_id). Fordi en del af spørgsmålene lægger sig op ad et instrument, for eksempel er det ikke relevant for en violinist at vide noget om klaverskalaer, men spørgsmålene kun ligger i en tabel, er spørgsmålets id dannet ved at gange instrumentets id med 100, og lægge et unikt tal mellem 0 og 99 til. Da spørgsmålene IKKE skal oprettes af brugerne, men skal indkodes manuelt, vi dette ikke skabe problemer i databasen. Naturligvis begrænser det antallet af spørgsmål pr. instrument (der kan kun være 100 spørgsmål for hvert instrument), men da der ikke i vores projekt skal være enormt mange spørgsmål, er dette ikke relevant, men hvis appen skal videreudvikles i fremtiden, kunne det eventuelt være en god ide at danne forskellige tabeller for det forskellige typer af spørgsmål (et for hver instrument).

Grunden til at det er sådan vi har valgt at strukturere spørgsmål id’erne, er at der i programmet, når der skal hentes et spørgsmål ud til flashcard-funktionen (musicTheoryFlashcardsScreen, musicTheoryFlashcardsScreen2 og musicTheoryFlashcardsScreen3) vælges et tilfældigt spørgsmål tilhørende instrumentet. Således findes instrumentets id (instrument\_id) først ved en query, hvorefter denne ganges med 100, og der lægges et tilfældigt tal mellem 0 og 3, da der ligger tre spørgsmål til hvert instrument. På denne måde findes der et tilfældigt spørgsmål til flashcard funktionen. Se databasekald og query herunder, som svarer til det, der stod i Windows-versionen af appen:

//Først defineres de variable. Deres aktuelle værdi er ikke angivet her.

String varUsername;

String varPassword;

int varUser\_id;

int varInstrument\_id;

int varQuestion\_id;

String varQuestion;

String varAnswer;

if (MasterMusic\_db.connect()) {

//Hvis databasen er forbundet, findes brugeren i databasen ved brug af variablerne, som opdateres ved en anden funktion (Log-in funktion).

MasterMusic\_db.query("SELECT \* FROM users WHERE username = \""+varUsername + "\"AND password = \""+varPassword+"\"");

// Find først user\_id ved username

if (MasterMusic\_db.next() && MasterMusic\_db.getString("username").equals(varUsername) && MasterMusic\_db.getString("password").equals(varPassword)) {

varUser\_id = MasterMusic\_db.getInt("user\_id");

}

//Find instrument\_id ved user\_id

MasterMusic\_db.query("SELECT \* FROM instruments\_users WHERE user\_id = \""+varUser\_id+"\"");

if (MasterMusic\_db.next() && MasterMusic\_db.getInt("user\_id") == varUser\_id) {

varInstrument\_id = MasterMusic\_db.getInt("instrument\_id");

//Find question\_id ved brug af instrument\_id (struktur), derefter spørgsmål

varQuestion\_id = varInstrument\_id \* 100 + int(random(0, 3));

}

MasterMusic\_db.query("SELECT \* FROM questions WHERE question\_id = \""+varQuestion\_id+"\"");

if (MasterMusic\_db.next() && MasterMusic\_db.getInt("question\_id") == varQuestion\_id) {

varQuestion = MasterMusic\_db.getString("question");

varAnswer = MasterMusic\_db.getString("answer");

Da programmet blev konverteret til Android, blev denne kode ændret en smule, da Android bruger det bibliotek der hedder ketai.data.\* i stedet for det bibliotek der blev brugt i Windows-versionen, nemlig de.beizer.sql.\*

## Sikkerhed

Da vi har lavet et program, der er brugerbaseret, er det vigtigt at betragte sikkerhed. Der er ikke nogen form for datadeling mellem brugerne, og der er generelt ingen kommunikation mellem brugerne gennem appen. De brugerdata, der skal gemmes, er derfor kun de data, der ligger i databasen. De skal kun gemmes én vej, så der er ikke behov for at kryptere noget data, da der i princippet ikke skal hentes personlig data ud af databasen igen. For at sikre sikkerheden vælger vi derfor blot at hashe kodeordet (*password*) og gemme dette hashede kodeord i databasen. Fordi kodeordet kan hashes, når der logges ind, og så bare sammenlignes med det gemte kodeord (som også er hashet), er der i princippet ikke behov for at gemme kodeordet nogen steder. Vi har valgt en rimelig simpel form for hashing. Herunder ses et eksempel på vores hashing, når man logger ind:

// preHashPassword vil her være det der er hentet fra et input-felt som brugeren kan skrive i.

String preHashPassword;

String password;

try {

MessageDigest md = MessageDigest.getInstance("SHA-256");

md.update(preHashPassword.getBytes());

byte[] byteList = md.digest();

StringBuffer hashedValueBuffer = new StringBuffer();

for (byte c : byteList)hashedValueBuffer.append(hex(c));

password = hashedValueBuffer.toString();

println(password);

}

catch (Exception e) {

System.out.println("Exception: "+e);

}

Dette er som sagt en rimelig simpel form for hashing(SHA-256 hashing), som måske ikke er helt sikkert nok til at programmet kan køres i en professionel kapacitet, da denne hashing ret nemt kan dekrypteres ved at søge på Google, men det vil være fint i den sammenhæng, vi bruger den i, da vi ikke har behov for professionel sikkerhed, med mindre appen en dag ville skulle ud på det virkelige marked.

## Brugertest af musikapp

Onsdag d. 12/5-2021 lavede vi en brugertest af vores musikapp. På dette tidspunkt var appen ikke helt færdig, men de vigtigste funktioner var lavet. Dog var appen ikke overført til Android på dette tidspunkt. Her testede vi appen på 4 elever fra HCØ Lyngby, hvoraf en af dem var musikfaren i forvejen. Det er naturligvis ikke helt muligt at teste, hvordan vores brugere ville forbedre sig over lang tid, og om de lærer noget af vores app, men vi kan teste hvad deres førstehåndsindtryk af appen er, og hvilke ting de føler er mangelfulde eller bør laves om. Feedbacken fra brugertesten kan ses i bilag 4.

Generelt var der blandt alle brugerne en smule kritik til generelle funktioner i appen, der ikke virkede helt, som de kunne have ønsket. Blandt andet udtrykte flere brugere et ønske om at kunne skrive tal i deres password og brugernavn. Desuden mente de, at det ville være oplagt at erstatte ordene med stjerner eller prikker, når man skriver sit password. Dette kunne også bidrage til en form for sikkerhed ved at sørge for, at passwordet ikke nemt kan læses af en uvedkommende person, når passwordet tastes ind af brugeren. De udtrykte også, at der var nogle problemer med nogle af knapperne, hvor man for eksempel skulle trykke to gange på de knapper, der var skabt via “InstrumentButton” klassen. Denne fejl skyldtes, at der var byttet om på nogle linjer, da knapperne blev implementeret. Denne fejl er blevet fikset, så knapperne virker rigtigt nu. Derudover var en kommentar, at valget af instrumentet skulle fungere mere som en rigtig knap, således at brugeren skal kunne se hvilket instrument, der er valgt. Denne del er nu også implementeret i koden. Endmere var det et problem, at når man er på log-in skærmen vises der stadig en header som indeholder en ‘Log out’ knap og en ‘Back’ knap. Disse kan være lidt misvisende, da man ikke kan logge ind eller gå tilbage, altså er det ikke engang muligt at trykke på knapperne. Flere brugere gav også udtryk for, at det ville være en god ide at gøre skriften lidt større på *Music Theory Flashcards Screen (1-3).* Disse små fejl og ønsker til mere strømlinede kunne man nemt behandle i fremtiden, således at helhedsindtrykket af appen bliver mere professionelt.

Udover disse små ting kom brugerne også med forslag til ting, der kunne tilføjes til eller ændres på funktionerne i appen. Dem der ikke havde forrig musikerfaring gav udtryk for at det ville være vellidt at få en begynder/øvet adskillelse af *Perfect Pitch Trainer* og *Node Name Trainer,* såvel som instruktioner der fortalte hvordan disse funktioner fungerer. Derudover var der et generelt ønske om en forklaring til, hvordan flashcards - *Music Theory Flashcards* - virker. Dette kunne være løst ved at lave en “About” side, hvor de forskellige spil bliver forklaret. En differentiering mellem forskellige niveauer kunne opnås ved at lade der være færre toner at vælge imellem på de lavere niveauer, så der f.eks. kun var 3 toner på et let niveau, 6 toner på et medium niveau osv. Der blev desuden gjort udtryk for et ønske om at kunne se sin highscore inde i selve spillene, *Perfect Pitch* *Trainer* og *Node Name Trainer*, og ikke bare i *HighscoreScreen*.

Den bruger, der havde forrig musikerfarring, gav udtryk for, at det ville være rart, hvis man kunne vælge imellem flere instrumenter. Dette skyldes nok mest, at denne person selv spillede trompet, som ikke var en valgmulighed. Som forlængelse af dette skulle der også differentieres mellem noderne for instrumenter, hvor noderne ser anderledes ud end ved de mere almindelige instrumenter såsom klaver, violin og guitar. Et af de instrumenter, hvor noder og toner navngives forskelligt fra mere almindelige instrumenter, er netop trompeter. Disse typer instrumenter kaldes transponerede instrumenter (Grubbe, 2020).

Endnu et forslag fra brugerne var at give brugeren en mulighed for selv at ligge flashcards op. På den måde kunne mere musikerfarne brugere tilrettelægge læringen, så den var mere personlig til de ting, brugeren helt specifikt ønskede at kunne huske udenad. Dette stemmer også overens med, at forskning viser, at selvlavede flashcards øger hukommelse af de lavede flashcards (Frank, 2020). Dette krav vil være lidt sværere at implementere, da det ville kræve en fundamental omstrukturering af programmet, da spørgsmålene lige nu kun kan tilføjes manuelt, hvilket betyder at det er nemt at gøre deres id afhængige af instrumentets id.

# Konklusion

Vi havde følgende problemstilling *“Med dette projekt vil det blive undersøgt, hvordan der kan fremstilles et program, som kan hjælpe musikinteresserede i at opnå en bedre forståelse for musikteori og hørelære.”.* Denne problemstilling har vi forsøgt at imødekomme ved at lave en musikapp, der inkluderer spil, som har til formål at træne brugerens evner indenfor musikteori og hørelære. Det har ikke været muligt decideret at teste, om brugerne på længere sigt vil opnå en større forståelse indenfor musik. Derfor tages der i stedet fokus i de opstillede krav i produkt forberedelse til vurdering af vores app. I forhold til brugervenlighed konkluderes det, at vores program er brugervenligt grundet vores inspirerende design. Dermed er kravet inspirerende design også opfyldt. Kravet motiverende og underholdende vurderes også at være opfyldt, da vi har inkluderet spil og highscore i vores musikapp. I forhold til om vores app er lærerig kan endnu ikke vurderes direkte, da det kun er os som programmører, der har testet programmet langvarigt. Dog vurderes det, at dette krav højst sandsynligt også er opfyldt, da vores app er brugervenlig, motiverende og underholdende, har inspirerende design, og der er stort fokus på, at spillene i appen både træner brugeren i hørelære med *Perfect Pitch Trainer* og musikteori med *Node Name Trainer* og *Music Theory Flashcards*.

# Perspektivering

Vi havde i vores design del skrevet, at vi gerne ville have, at designet af musikappen afspejlede hvilket instrument, brugeren valgte ved sign up. Dette var det eneste fra selve design kravene, som vi ikke nåede at få gennemført, da vi længere inde i projektet opdagede, at det var for tidskrævende for os. Vi valgte derfor at lave et “musikneutralt” design i stedet. Det kunne være en mulighed, at vi i fremtiden lavede indstillinger, der ændrer farverne og designet ud fra hvilket instrument, brugerne har valgt. Spørgsmålene til *Music Theory Flashcards* reflekterer dog stadigvæk det instrument, brugeren har valgt, så det kunne være en mulighed at knytte designet til instrumentet på en lignende måde.

I fremtiden ser vi også, at vores app skal kunne overføres til iPhone. Som skrevet i realiseringen var det for os ikke en mulighed, da Apple er et lukket miljø, og vi havde derfor kun mulighed for, at eksportere vores app over på Android devices. Derudover minder det sprog, der bruges på Android, meget mere om Java end det sprog, der bruges på IOS-enheder.

I vores endelige program er der desuden mindre ting i databasen som ikke helt opfylder de krav, vi havde stillet til dem. For eksempel gemmes den tid der bruges på appen (både om dagen og i alt) lige pt ikke i databasen. Der er gjort plads til, at man vil kunne gemme det i databasen, så det ville være en mulighed hvis man skulle videreudvikle programmet at sørge at kunne gemme denne tid. Desuden er databasen ikke helt optimalt struktureret i forhold til spørgsmålene, som nævnt tidligere i rapporten. Hvis der skulle videreudvikles på appen, ville det derfor nok også være en god ide at ændre lidt på denne struktur, således at det ville være lettere at lave en stor mængde spørgsmål til hvert instrument, og eventuelt også udarbejde en måde, hvorpå man kan oprette generelle spørgsmå,l der vedrører alle instrumenter. Desuden kunne det også være muligt at lave flere forskellige valgmuligheder for instrumenter.

Af mindre detaljer, vi ikke nåede at rette i vores kode, kunne der for eksempel have været flere knapper i Node Name Trainer i stedet for at slå knapper såsom D-sharp og E-flat sammen til én knap. Grunden til, at vi valgte at slå knapperne sammen, skyldes, at tonerne i virkeligheden lyder ens, selvom de teknisk set ikke er ens. Skulle de have været seperate knapper, ville der i alt være 17 knapper, hvilket er et primtal og derfor ikke kan deles op i tastaturformat, og dermed ville designet ikke se lige så pænt ud. Dog ville der kunne indføres en ny knap, som brugeren ville kunne trykke på for at differentiere mellem “sharp” og “flat” noder, så denne knap ville fungere ligesom en shift-knap på et normalt computertastatur.

Som beskrevet tidligere i rapporten, burde knapperne “Back” og “Log out” heller ikke være i headeren på login in siden, og “Log out” burde heller ikke være der på sign up siden. Derudover ville det være optimalt, hvis tiden blev gemt i databasen. Desuden er der en fejl ved, at usernames og passwords fra forrige bruger ikke bliver slettet, når brugeren logger ud igen.

I master\_music.pde connectes der til databasen, når teksten for flashcards skrives ovenpå selve flashcardsne i Music Theory Flashcards. Koden til dette er på nuværende tidspunkt skrevet i draw-funktionen, hvilket hurtigt kan blive et stort problem, da det er enormt hårdt for et program at skulle connecte til databasen rigtig mange gange pr. sekund. Dog kan koden ikke bare som sådan flyttes ud af draw-funktionen, da dette vil kræve en hel omstrukturering af koden. Det vil dog klart være en fordel, hvis denne kode blev omstruktureret og flyttet ud af draw-funktionen i fremtiden, så vores program kan køre en del lettere.

# Bibliography

Anders Dohn, E. A.-V. (2014, april). *Musical Activity Tunes Up Absolute Pitch Ability*. From JSTOR: https://www.jstor.org/stable/10.1525/mp.2014.31.4.359?seq=1

Faktalink. (n.d.). *Danskerne og sociale medier*. Retrieved maj 17, 2021 from Faktalink - bibliotek og undervisning: https://faktalink.dk/node/1615

Frank, T. (2020, august 19). *8 Better Ways to Make and Study Flash Cards*. From College InfoGeek: https://collegeinfogeek.com/flash-card-study-tips/

Grubbe, L. (2020, april 17). *Transponerende instrumenter*. From Musikipedia: https://www.musikipedia.dk/transponerende-instrumenter

Hørsolm Kommunale Musikskole. (n.d.). *MUSIKTEORI HØRELÆRE KOMPOSITION IMPROVISATION*. Retrieved maj 17, 2021 from Hørsolm Kommunale Musikskole: https://musikskolen.horsholm.dk/undervisningstilbud/teori-og-hoerelaere

Kwik, J. (2020). *Limitless.*

Meylan, C. (n.d.). *From passive to active learning: Pros, cons and 6 strategies*. Retrieved maj 17, 2021 from EHL Insights: https://hospitalityinsights.ehl.edu/passive-active-learning

Musikipedia. (n.d.). *Absolut Gehør*. Retrieved maj 17, 2021 from Musikipedia: https://www.musikipedia.dk/leksikon/absolut-gehoer

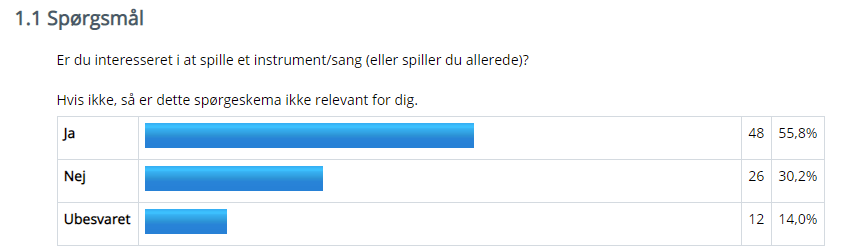
# 

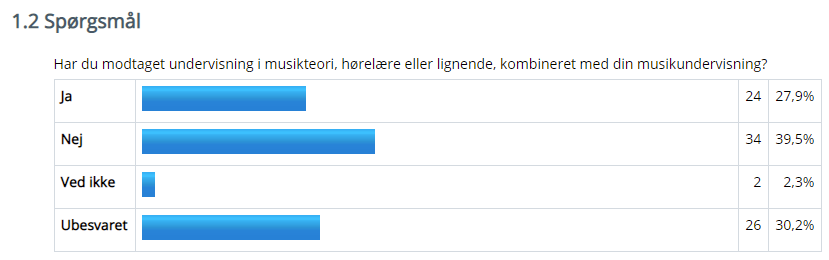
# Appendiks

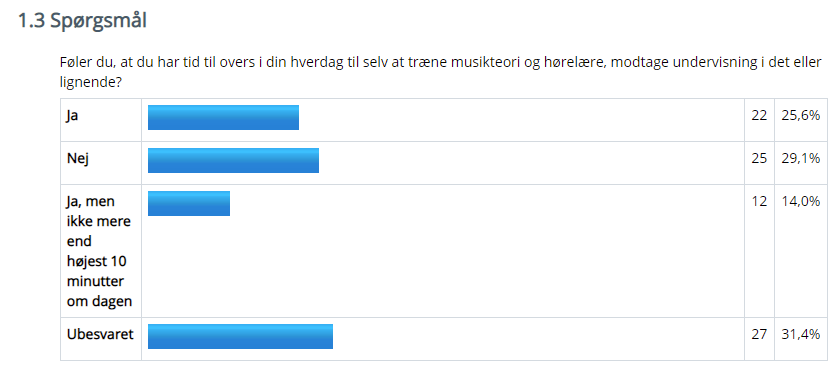
I følgende afsnit gennemgås de bilag, der er refereret til gennem rapporten.

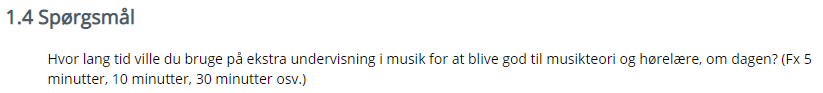
## Bilag 1- Spørgeskema

Med dette spørgeskema er musikinteresserede HTX-elevers og -læreres mening om og erfaring med undervisning i musikteori og hørelære blevet undersøgt. I spørgsmål 1.4. og 1.6. er de respondenter, der har svaret blankt, blevet fjernet.

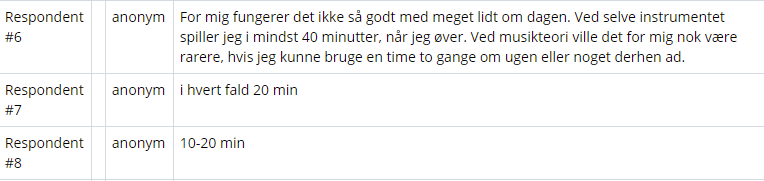




















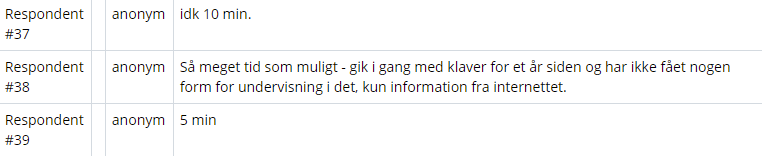








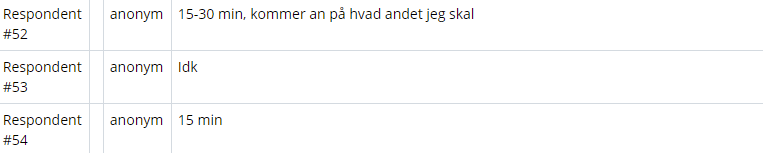










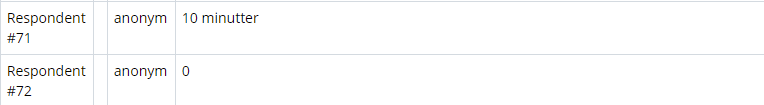


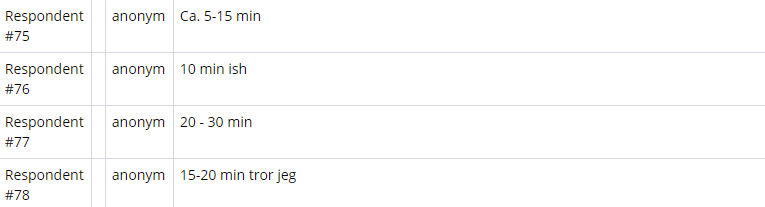




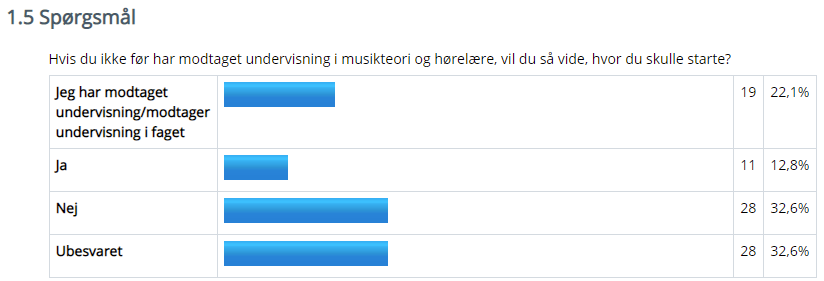


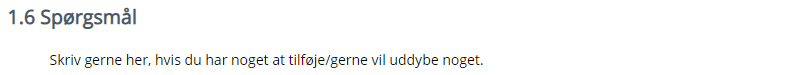


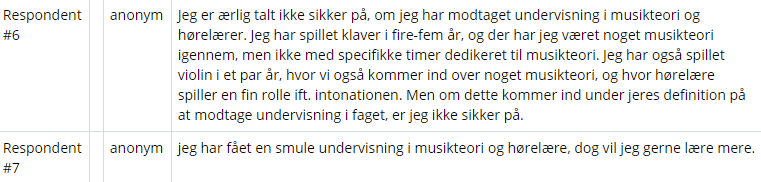










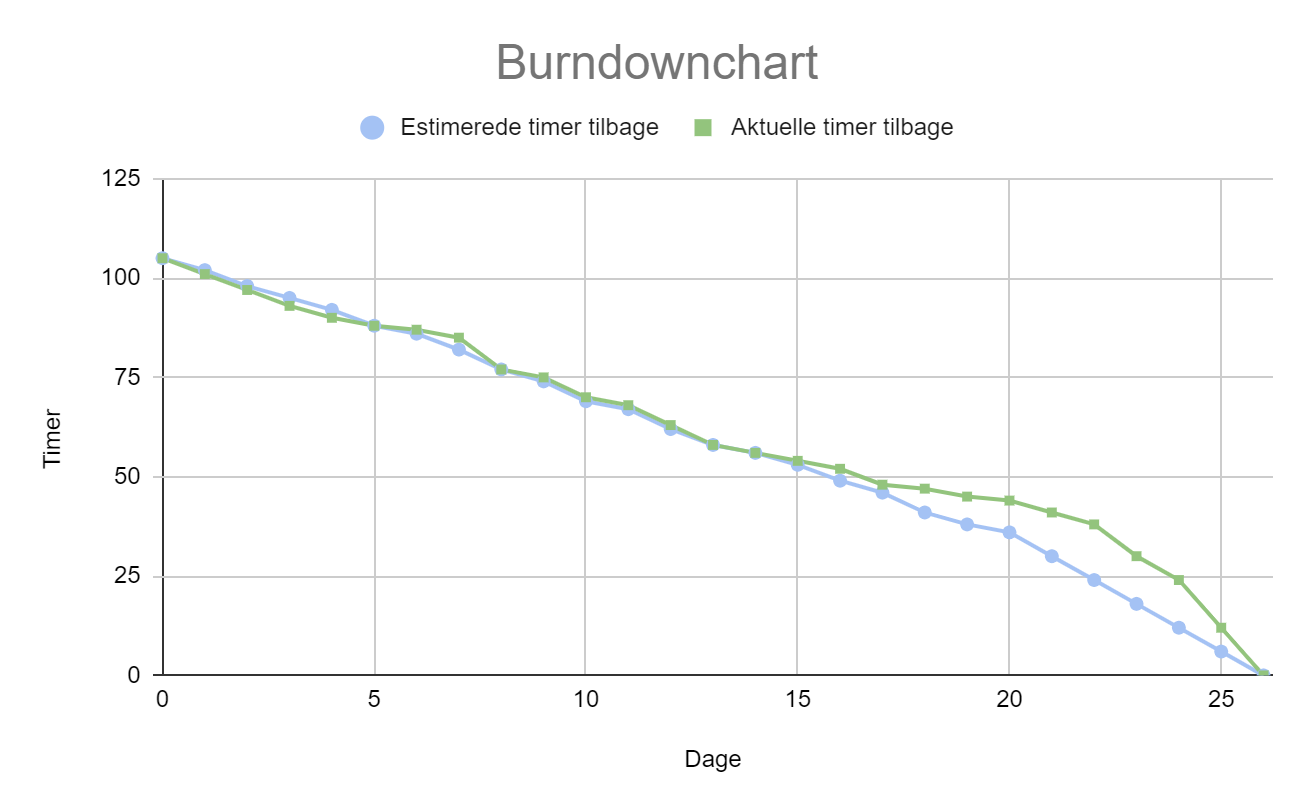


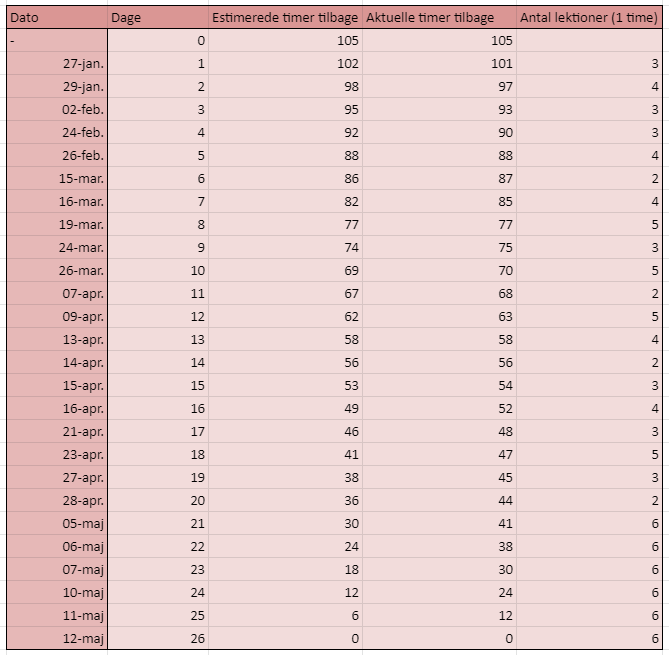




## 

## Bilag 2 - Burndown Chart





## Bilag 3 - Link til indledende brugertest og feedback fra brugere

Link til interaktive brugertest: <https://marvelapp.com/prototype/6i86g2j>

|  |  |  |
| --- | --- | --- |
| **Brugerfeedback** | | |
| Bruger 1 | 23-årig musikerfaren. | * Ikke spørgsmål målrettet mod instrument - ellers skal man vælge det instrument, man spiller, for at spørgsmålene giver mening. * Arbejde med design - farver - fx vælge farver alt efter det instrument, man har sagt, man spiller. * Niveauer er vigtigt (kunstig intelligens) - målrette appen mod en større målgruppe.  Evt. vælg kategori, og derefter sværhedsgrad/sværhedsgrad ændrer sig via. kunstig intelligens analyse. * Tone i Perfect Pitch Trainer kommer fra det instrument, man spiller. * Lave en test, der vurderer startniveau (evt. via. kunstig intelligens) eller start med at vælge det niveau, man er på. * Svarmuligheder/skriv svar i boks til flashcards. I dette tilfælde er sværhedsgrader på det vendte flashcards irrelevant. |
| Bruger 2 | 23-årig ikke-musikerfaren. | * Eventuelt muligheder ved flashcards. * Ikke helt indlysende, hvordan flashcards virker. * ”I alt” tid i bunden af startside. * Giver ellers god mening. |

## Bilag 4: Brugertest af musikapp

Følgende brugertest blev udført på skolen onsdag d. 12/5-2021.

Henrik Knudsen 3b1

Musikerfaren: Nej.

Feedback:

· Vis prikker i stedet for tegn ved password.

· Gør det muligt at skrive tal ved username og password.

· UI ”choose instrument” funktionen bør fungere som en reel knap (app konventioner).

· Oplagt med en tutorial for, hvordan man genkender toner i Perfect Pitch Trainer.

· Tutorial for Node Name Trainer.

· Oplagt med levels hvor der i lettere levels er færre knapper ad gangen i PPT & NNT.

· Tutorial til hvordan flashcards fungerer.

· Highscore inde i selve PPT & NNT.

Oscar Strauss 3e

Musikerfaren: Nej.

Feedback:

· Større tekst på flashcards.

· Begynder mode/advanced mode (to tutorial or not) i PPT & NNT.

· Fejl i timeren (øh, der er ikke gået 3 timer).

Ea Charlotte Kiel Andersen 3c (og Benjamin Fynbo Keiding 3d2 - ikke musikerfaren person)

Musikerfaren: Ja.

Feedback:

· Tal i password.

· Klik to gange på hvide knapper for at de virker.

· Flere instrumenter, og tag hensyn til andre instrumenter (fx trompeter) hvis toner og noder ikke svarer til de almindelige.

· Større tekst i flashcards.

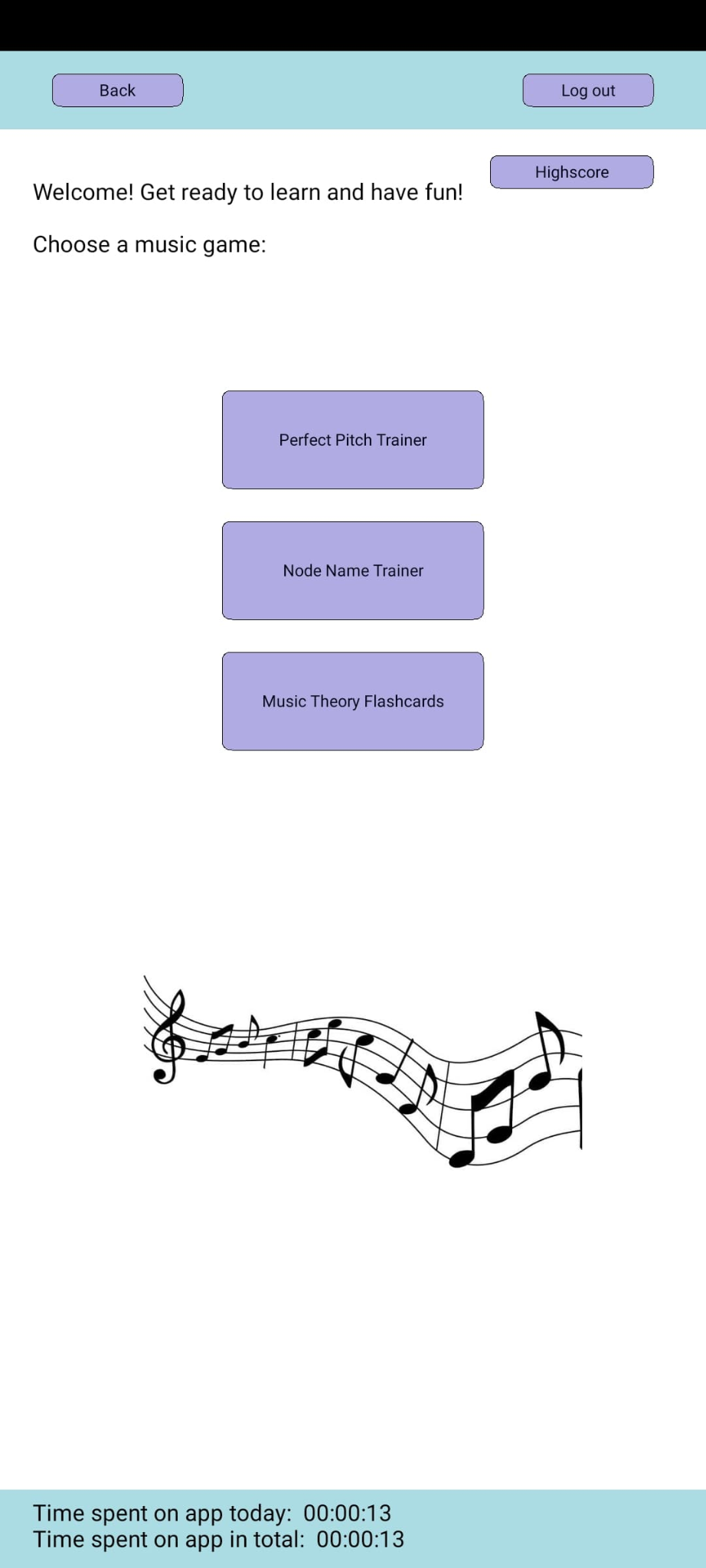
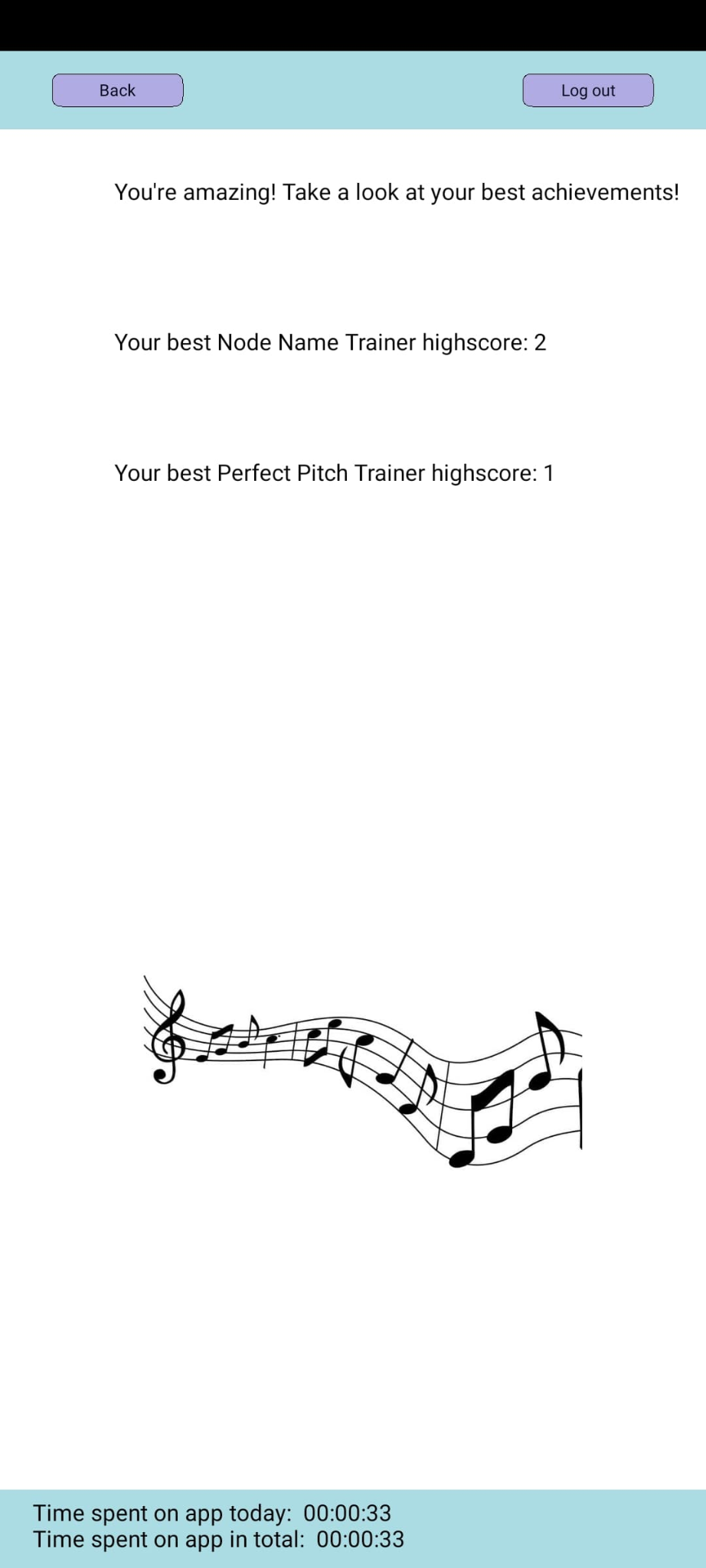
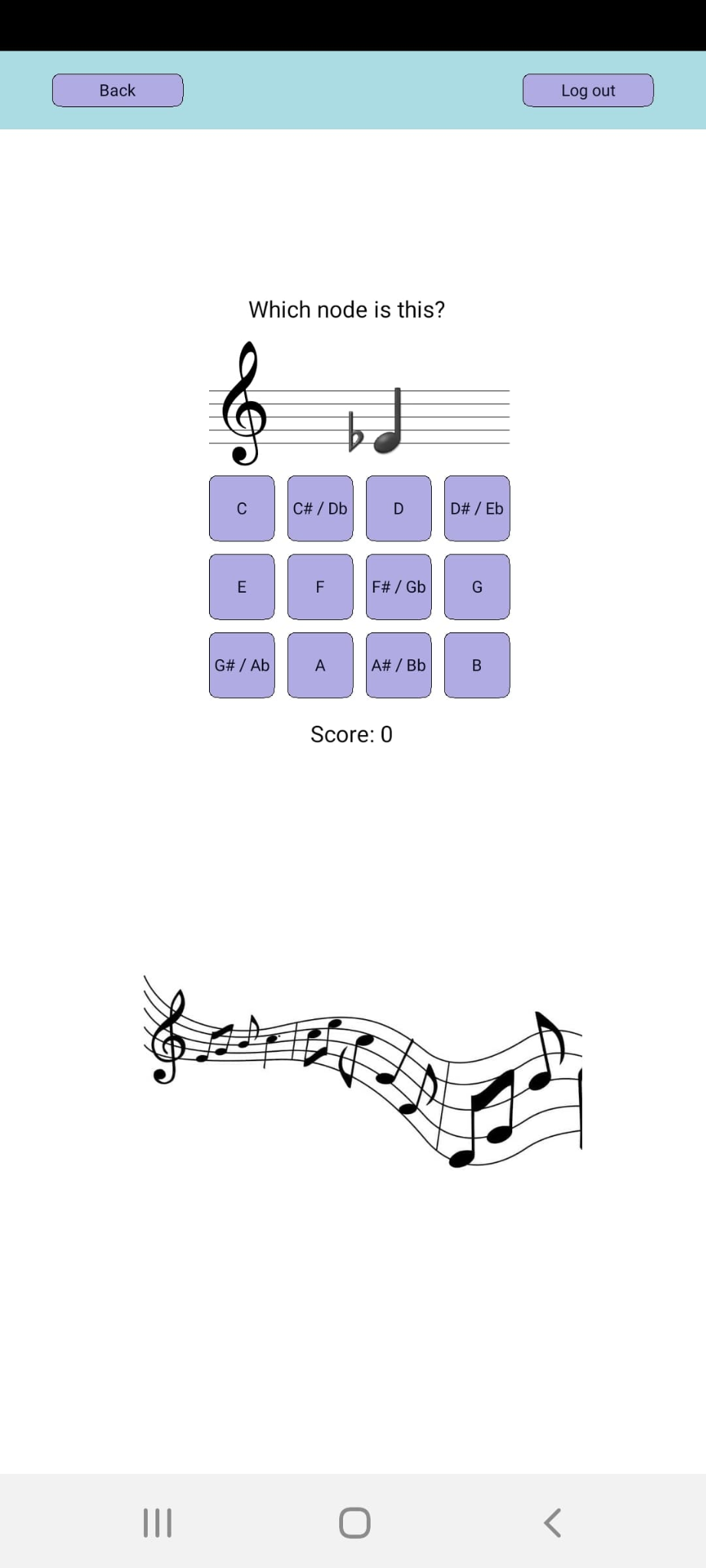
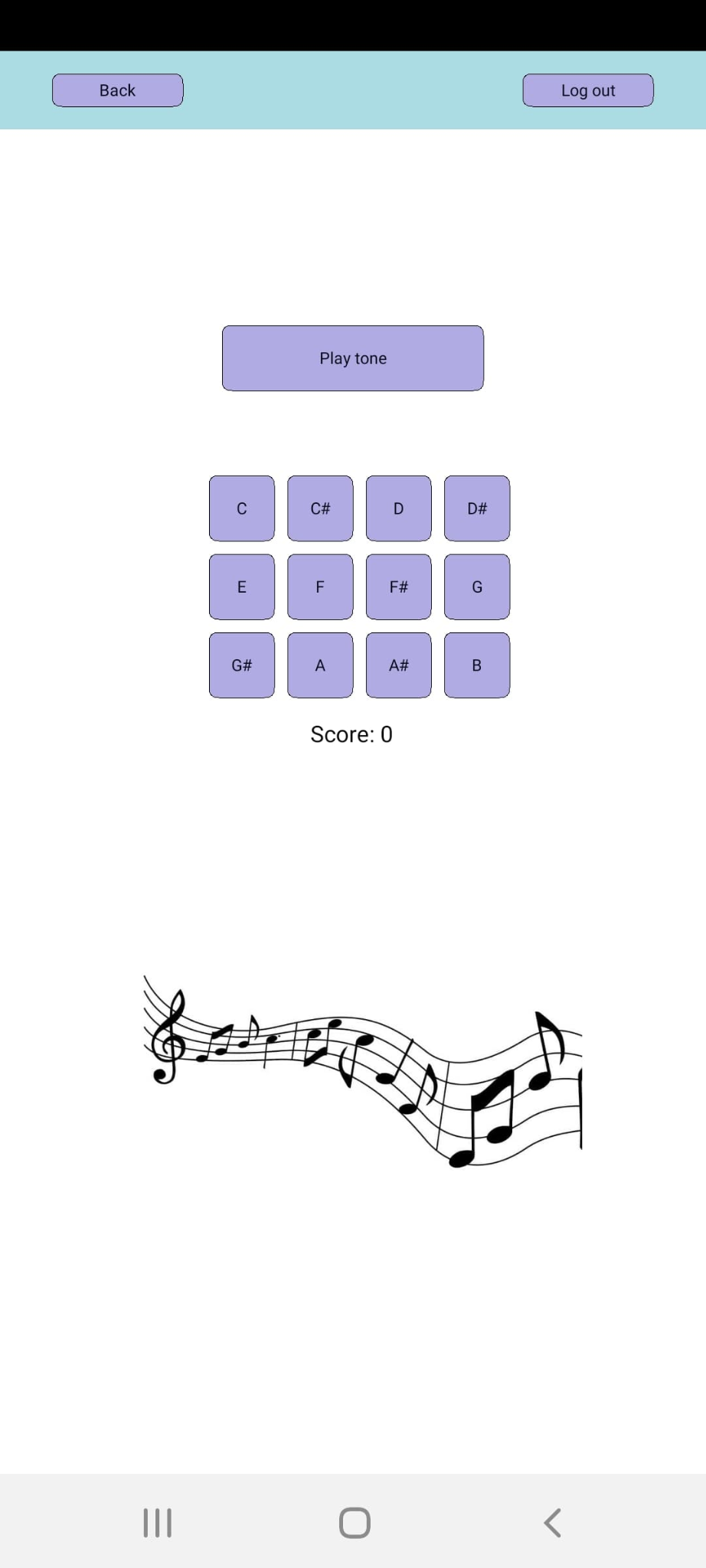
· Stavning (spent ikke spend & sætning på welcomeScreen).

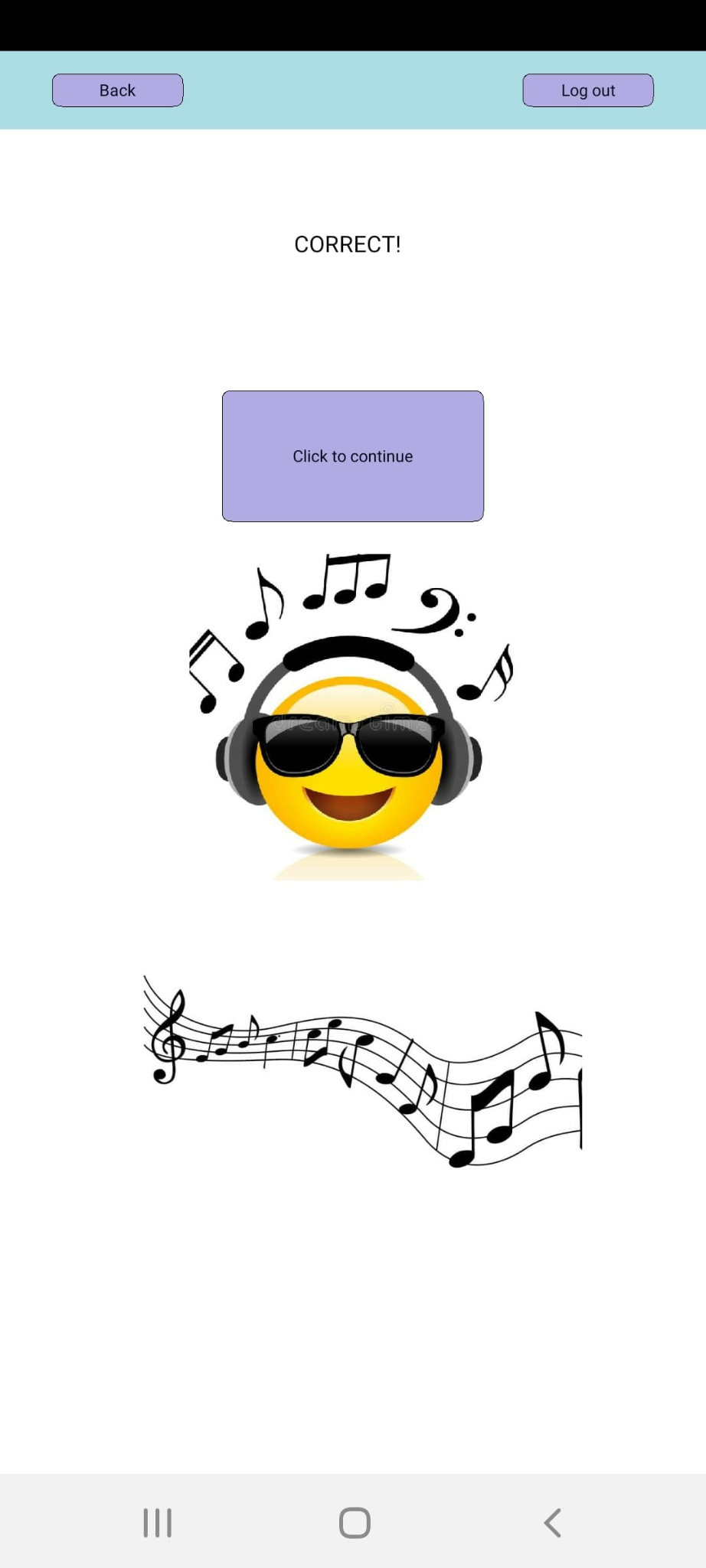
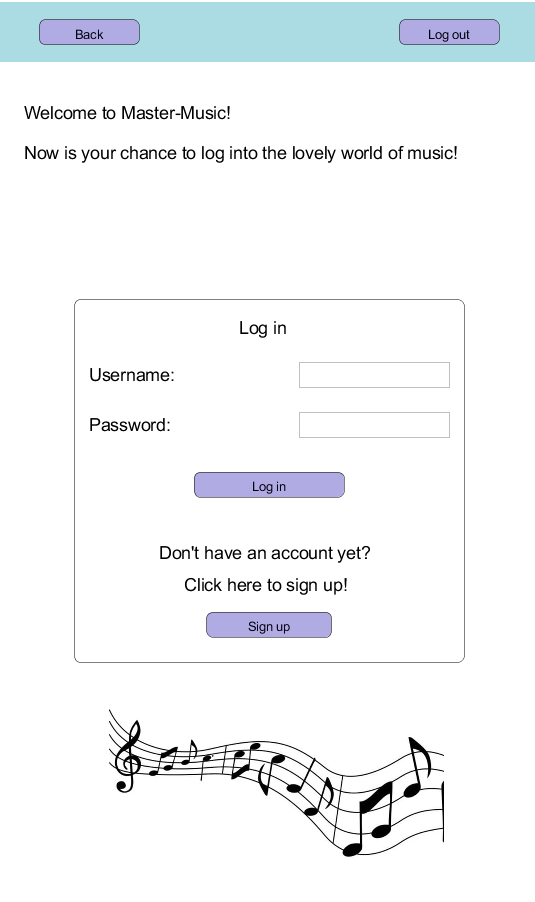
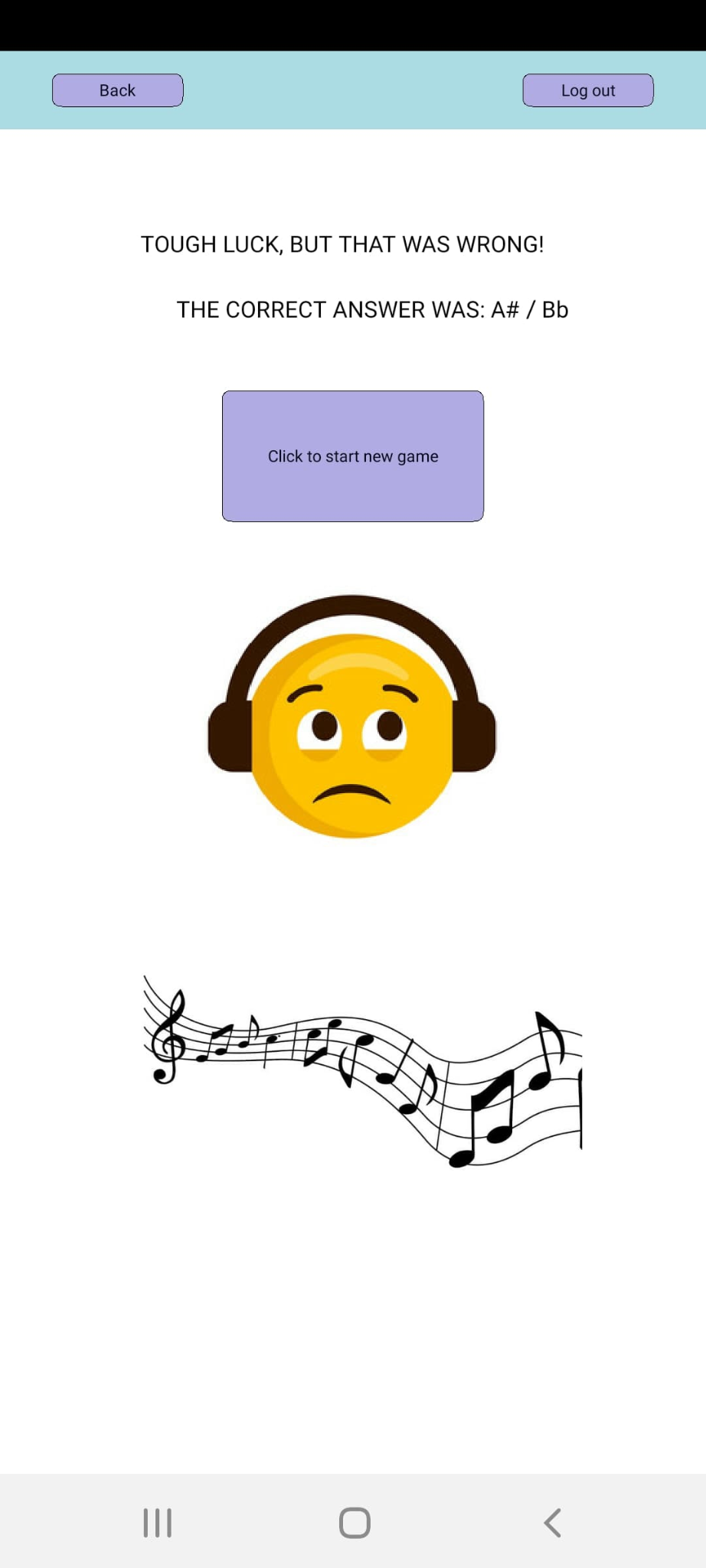
· Ændre ”Play tone” til ”Pause” når playButton trykkes på.

· Muligt at uploade egne flashcards.

· Misvisende knapper i Header på forskellige sider (fx Log Out på loginScreen).

## Bilag 5 - Skærmbilleder af den endelige app

## Bilag 6 – Kontrakt

Link til kontrakt

<https://drive.google.com/file/d/1iKS3T8n__Fln1eDBpwJ46lb3svCHfn0_/view?usp=sharing>

## Bilag 7 – Mødelog

Link til mødelog

<https://docs.google.com/document/d/1Eg-q5HLH7uhzPtIOEUUvc9VW0MLHOnJT8lNenXsLZ6Q/edit?usp=sharing>

## Bilag 8 – GitHub repository

Link til GitHub repository

<https://github.com/cecilianygaard/master-music>

1. Bilag 1. [↑](#footnote-ref-1)
2. Bilag 1. [↑](#footnote-ref-2)
3. Bilag 1. [↑](#footnote-ref-3)