

Kursbaserad SRS-applikation för studieunderlättande

Examensarbete inom högskoleprogrammet Datateknik

Oskar Vajlok, Kevin Lundkvist

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

Kursbaserad SRS-applikation för studieunderlättande

Oskar Vajlok, Kevin Lundkvist

5 juni 2026

Examensarbete inom högskoleprogrammet Datateknik



UNIVERSITY OF
GOTHENBURG

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2026

© Oskar Vajlok & Kevin Lundkvist, 2026.
Handledare: Sakib Sisteek
Examinator: Johannes Åman Pohjola

Abstract

This project creates an application that utilizes spaced repetition as a means to simplify for students to learn course material. Spaced repetition spreads out study sessions over time to optimize learning and retention. The application was developed to allow course responsables to create courses in the form of concepts with related material and review questions. Students have the ability to join courses that have been published. Material is only made available to students when its prerequisites have been fulfilled. In order to fulfill a prerequisite, a student must review the material and correctly answer questions to increase their level for the material. When a sufficient level has been reached a prerequisite is considered fulfilled. Review sessions are scheduled according to intervals based on the well established algorithm SM-2 and research by Hermann Ebbinghaus. After the application was implemented a user test was conducted over the course of a week, which had its results quantified through a survey of its participants. The overall impression of the application was positive although its effect on learning outcomes is yet to be proven. To show that such an effect exists would require a more extensive study.

Sammanfattning

Projektet skapar en applikation som med hjälp av spaced repetition ska underlätta för studenter att lära sig kursmaterial. Spaced repetition sprider ut studietillfällena för att optimera inläring och kunskapsbevarande över tid. Applikationen utvecklades för att tillåta kursansvariga att skapa kurser i form av koncept med tillhörande material och repetitionsfrågor. Studenter har möjlighet att ansluta till kurser som har publicerats. Material ska enbart tillgängliggöras för studenter när dess förkunskapskrav är uppfyllda. De uppfylls genom att studenter repeterar material och korrekt besvarar frågor för att öka sin nivå för materialet, när en tillräckligt hög nivå uppnåtts anses ett förkunskapskrav vara uppfyllt. Repetitionstillfällena sker utefter intervall baserade på den välundersökta algoritmen SM-2 och forskning av Hermann Ebbinghaus. Efter färdigställd applikation har utvärdering skett via ett användartest som varade i en vecka och kvantifierats sedan via en enkät besvarad av deltagarna. Deltagarnas intryck av applikationen är positiva men påverkan på studier är ännu obeprövad då syftet med spaced repetition ej testats. För att testa effekterna av applikationens användning krävs en mer omfattande och längre studie.

Nyckelord: SRS, spaced repetition, inläring, webbapplikation, PWA, minne

Förord

Idén till detta examensarbete kommer ifrån våra egna upplevelser under studietiden. Vi har vid ett flertal tillfällen upplevt att det hade varit fördelaktigt, för oss själva och för klasskamrater, om kunskapen från tidigare kurser hade varit färskare i minnet. Detta blev utgångspunkten för arbetet.

Vi vill även speciellt tacka några personer för deras insikter och hjälp under projektets gång. Vi vill tacka vår handledare Sakib Sisteck för den värdefulla vägledningen och återkopplingen under projektets gång. Vi vill tacka vår examinator Johannes Åman Pohjola för återkopplingen om förbättring på rapporten. Slutligen vill vi tacka Sebastian Danckwardt och Joel Andersson för insikter om rapportens förbättringsmöjligheter.

Oskar Vajlok & Kevin Lundkvist, Göteborg, Juni 2026

Akronymer

API - application programming interface

BFS - breadth-first search

DAG - directed acyclic graph

EF - easiness factor

JSON - JavaScript object notation

PWA - progressive web application

SRS - spaced repetition system

Nomenklatur

Active recall	Att träna minnet genom aktiv återkallning av information.
Backend	Systemets serversida som hanterar logik och data.
Cramming	Att studera mycket under en kort tidsperiod.
Cookie	En liten mängd text webbsidor sparar i webbläsare för att spara information om användaren.
Cron-jobb	Schemalagd uppgift som körs automatiskt.
Endpoints	Specifika URL:er i en API där en klient kan skicka en förfrågan.
End-to-end typsäkerhet	Att samma typdefinitioner används genom hela applikationen för att möjliggöra kompilator att i förväg varna för fel.
Forgetting curve	En kurva skapad av resultaten från Hermann Ebbinghaus experiment om minne och glömska.
Frontend	Applikationens användargränssnitt.
Fuzzy matching	Jämförelse mellan strängar som tillåter stavfel.
Progressive web application	En webbsida med förmåga att laddas ner som applikation.
Progressbar	En mätare av hur stor andel av något som är uppfyllt.
Routes	Typade API-endpunkter som är anropbara som funktioner.
Session token	En unik sträng som en server skickar till användares webbläsare för att identifiera användaren, exempelvis för att inloggning av webbsida inte ska behövas.
Spaced repetition	Att fördela studietillfällen utspritt under en längre tid.
Spacing effect	Att inlärning förbättras när studier sker på utspridda tillfällen.
μ	Medelvärde.
σ	Standardavvikelse.

Innehåll

Sammanfattning	V
Förord	VI
Akronymer	VII
Nomenklatur	VIII
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Mål	2
1.4 Avgränsningar	2
2 Metod	3
2.1 Utvecklingsprocess	3
2.2 Verktyg och ramverk	3
2.3 Utvärdering	3
3 Teori	4
3.1 Spaced repetition	4
3.2 Active recall	4
3.3 Algoritmer för intervallberäkning	4
4 Systemkonstruktion	6
4.1 Ursprunglig design	6
4.2 Övergripande arkitektur	6
4.3 Databasdesign	6
4.4 Inloggning och autentisering	8
4.5 Spaced repetition-algoritm	8
4.5.1 Inlärningsfas (nivå 0-3)	9
4.5.2 Mellansteg (nivå 4)	9
4.5.3 SM-2 progression (nivå 5-10)	9
4.5.4 Alla nivåer	9
4.5.5 Hantering av felaktiga svar	9
4.6 PWA och notifikationer	10
4.7 Förkunskapskrav och beroendehantering	10
4.8 Funktionalitet och användarupplevelse	12
4.8.1 Kursansvariga	12
4.8.2 Studenter	12
4.9 Designavvikelser	13
4.10 Driftsättning för användartest	13
5 Resultat	14
5.1 Applikationen	14
5.2 Enkäten	16
6 Diskussion	18

6.1	Modulbaserad kontra kortbaserad repetition	18
6.2	SRS-algoritm	18
6.3	Notifikationer	18
6.4	Motivation och progression	19
6.5	Användbarhet	19
6.6	Enkätens begränsningar	19
6.7	Vad hade gjorts annorlunda	20
6.8	Etiska aspekter	20
6.9	Miljöaspekter	20
7	Slutsats	21
7.1	Utvärdering av arbetsprocess	21
7.2	Projektets styrkor och svagheter	21
7.3	Vidareutveckling	21
	Källförteckning	23
A	Ursprunglig gränssnittsdesign	25
B	Ursprunglig databasdesign	29
C	Systemets utseende	30

Figurer

1	Övergripande arkitektur av applikationen, skapad i Excalidraw	6
2	ER-diagram över systemets databasschema, skapat i Excalidraw	7
3	En simplifierad version av inloggningsflödet, skapad i Mermaid	8
4	Sekvensdiagram av cron-jobbet som skickar ut notifikationer, skapad i Mermaid	10
5	Modulrelationer, skapad i Excalidraw	11
6	Problemet med cykel mellan moduler, skapad i Excalidraw	11
7	Vy med interaktiv graf	12
8	Kursansvarigas modulvy, uppdelad i: 1.Teori, 2.Frågor, 3.Förkunskapskrav. Redigerad i Paint.	14
9	Studenters kursvy, uppdelad i: 1.Progressionskort, 2.Repetitionsknapp, 3.Inlärningsknapp, 4.Modulsektion. Redigerat i Paint	15
10	Studentvyn för en fel besvarad fråga	15
	Figurer i bilagor	25
11	Hemskärm och kursöversikt för studenter, skapad i Figma	25
12	Modulöversikt och studeringsskärmar med, samt utan svar, skapad i Figma	26
13	Översikt av kurser skapade av kursansvarig, skapad i Figma	26
14	Kursöversikt för kursansvarig, skapad i Figma	27
15	Modulöversikt för kursansvarig, skapad i Figma	27
16	Frågevy för kursansvarig, skapad i Figma	28
17	Graf av modulers beroende, skapad i Figma	28
18	Ursprunglig databas design, skapad i Excalidraw	29
19	Startsida för applikationen	30

20	Översikt av de kurser en kursansvarig har skapat	30
21	Kursöversikt för kursansvarig	31
22	Modulöversikt för kursansvarig	31
23	Exempel av en färdig flervals-fråga	32
24	Exempel av en tom fråga	32
25	Graf över modul-relationer	33
26	Användarinställningar för kursansvarig, redigerad med Gimp	33
27	”Dark Mode” på kursöversiktssidan	34
28	Inloggningsskärm för mobiler	34
29	Översikt av kurser studenten är med i	34
30	Studenters översikt av en kurs	35
31	En öppnad modul	35
32	Inlärnings sidan för flera moduler	35
33	Sista modulen i en inlärnings-session	35
34	En fråga med svarsalternativ valt	36
35	En fel besvarad fråga	36
36	Studenters inställningsmeny, redigerad med Gimp	36
37	”Dark Mode” för en student på kursöversiktssidan	36

1 Inledning

Ett missförstånd som vissa studenter har är att målet med en kurs är att klara tentamen. Detta missförstånd kan leda till att dessa studenter optimerar sin inläring för en godkänd tentamen, ofta genom att enbart studera kort inpå tentamen. Även om detta tillvägagångssätt kan leda till godkända tentor motverkar denna metod kursens egentliga mål, att studenten lär sig innehållet och behåller kunskapen över tid.

1.1 Bakgrund

Spacing effect är ett fenomen som har sitt tidigaste stöd på 1800-talet. Den tyska psykologen Hermann Ebbinghaus gjorde experiment som publicerades redan 1885 på tyska och översattes 1913 till engelska [1]. Det har i de över 100 följande åren stötts i ett flertal studier som McIntyre och Munson har noterat [2]. Den effekten visar att mängden kunskap som bevaras vid inläring är korrelerat till tidsspridningen av studietillfällena och säger att samma mängd studietid resulterar i mer bevarad kunskap när det finns fler tillfällen.

Den studiemetod som istället minimerar tidsspridningen, och till följd mängden kunskap som bevaras över tid, kallas för cramming. Trots dessa begränsningar så är det enligt McIntyre och Munson [2] den studiemetod som många studenter använder sig av under sin studietid. De fann att studenter som använt sig av cramming negativt besvarat påståendet att de minns kursers innehåll sex månader efter dess slut, μ (medelvärde) = 2.25 och σ (standardavvikelse) = 1.37 på en sju-poängsskala som rapporterats i deras appendix A.

En inlärningsmetod som istället har försökt optimera tidsspridningen är spaced repetition. Kang [3] beskriver spaced repetition som en metod där studietillfällen sprids ut, detta förbättrar den studerandes förmåga att återkalla kunskapen och därav lära sig den. Vidare påpekar Kang på att spaced repetition inte enbart hjälper med att komma ihåg kunskapen utan även förmågan att tillämpa den till nya scenarion. Det kan därför vara till studenters nytta att lära sig kursinnehållet med hjälp av spaced repetition för att underlätta framtida kurser som bygger på den nuvarande.

1.2 Syfte

Universitet förväntar att studenter behåller kunskap från kurser genom hela utbildningen. Men studenters vanligt förekommande studiemetod, cramming, motverkar långsiktigt kunskapsbevarande. Genom att i samband med en kurs tillgängliggöra spaced repetition av kursmaterial enligt forskningsbaserade intervall ska projektet underlätta studenters inläring, samt fortsatta kunskapsbevarande även efter kursens slut. Detta stärker deras förutsättningar i pågående kurser samt framtida kurser som bygger på kunskapen från dessa.

1.3 Mål

Målet är att utveckla en kursbaserad SRS-applikation med följande delmål:

- Funktionalitet för kursansvariga att skapa kurser.
 - Kurser ska bestå av en uppsättning moduler.
 - Varje modul kan ha förkunskapskrav i form av andra moduler och låses upp först när dessa är uppfyllda, så att studenten har de nödvändiga förkunskaperna innan en ny modul påbörjas.
 - Varje modul ska innehålla studiematerial samt frågor om materialet.
- Studenter ska kunna repetera kursmaterial med spaced repetition
 - Studenter ska erbjudas möjlighet att delta i kurser.
 - När en student går med i en kurs ska enbart moduler som saknar förkunskapskrav vara tillgängliga.
 - En ny modul låses upp när studenten uppfyllt dess förkunskapskrav.
 - Modulens nivå ökas genom korrekt besvarande av frågor relaterat till modulens material.
- Repetitionsintervallen ska baseras på en väl undersökt algoritm vald utifrån en undersökning av existerande SRS-algoritmer.

1.4 Avgränsningar

- Applikationen utvecklas som en PWA (Progressive Web Application) och görs endast tillgänglig via webbläsare, inte appbutiker.
- Applikationen testas ej i verklig kursmiljö under projektets gång.
- Enbart begränsat kursmaterial skapas i syfte av användartestning.
- Applikationen är ett studiestöd och används inte för betygsättning eller annan bedömning.

2 Metod

Detta avsnitt beskriver projektets utvecklingsprocess, verktyg och ramverk som användes samt hur projektet utvärderades.

2.1 Utvecklingsprocess

Utvecklingen inleddes med en övergripande databasdesign samt detaljerade gränssnittsdesigner i Figma för applikationens vyer. Implementationen följde därefter en gränssnittsdriven process. För varje vy implementerades först frontend-komponenterna utifrån gränssnittsdesignen. Därefter skapades de backend-endpoints och servicefunktioner som krävdes för att förse vyn med data eller hantera användarinteraktioner.

2.2 Verktyg och ramverk

Applikationen byggdes med T3-stacken, som ger end-to-end typsäkerhet genom hela applikationen. T3-stacken [4] består av Next.js [5], tRPC [6], Prisma [7], NextAuth [8], TypeScript och Tailwind CSS. Databasen bygger på Prisma-modeller mot en PostgreSQL-databas. Prisma agerar som ett abstraktionslager mellan databasen och applikationen, och hanterar övergången från relationsdatabasen till den objektstruktur som applikationen sedan använder. TypeScript-typer genereras från databasschemat och tRPC propagerar dessa typer till frontend via typinferens, vilket genomgående ger typsäkerhet. tRPC använder Zod [9] för att validera inkommande data i API (Application Programming Interface) anrop, vilket validerar typer under programkörning utöver den statiska typsäkerheten. Vidare hanterar NextAuth inloggning och sessioner. Gränssnittet utformades och stiliserades med Tailwind CSS i kombination med komponentbiblioteket Shadcn. Valet av T3-stacken baserades på dessa egenskaper i kombination med tidigare erfarenhet hos författarna.

Utöver T3-stacken används TipTap [10], en textredigerare med avancerad formatering, för att skapa modulinnehåll och frågor. Nodemailer [11] används för att skicka inloggningslänkar via e-post. Bun [12] valdes som pakethanterare för dess snabba installationstider samt lagringseffektivitet.

2.3 Utvärdering

Ett användartest med 12 deltagare utfördes i återkopplingssyfte. Användartestet påbörjades när applikationen var färdigutvecklad och varade i en veckas tid. Deltagarna fick tillgång till webbapplikationens med enbart en förfrågan om att använda applikationen. Endast studentens användning av applikationen testades, funktionaliteten för kursansvariga utvärderades ej. Efter testveckans slut skickades en länk ut till deltagarna med förfrågan att besvara en enkät. Enkäten utformades med en poängskala från ett till sex, vilket inte ger ett neutralt svarsalternativ för att kräva ställningstagande till alla frågor. Resultaten från denna enkät analyserades för att bedöma deltagarnas upplevelse av applikationen samt för att skapa underlag för vidareutveckling.

3 Teori

I det här avsnittet presenteras bakgrundsteori relevant till projektet om spaced repetition, active recall, testing effect och algoritmer för intervallberäkning.

3.1 Spaced repetition

Spaced repetition är en inlärningsmetod där man sprider ut studietillfällena över tid till skillnad mot att studera allt vid ett tillfälle. Grunden för spaced repetition kommer från Hermann Ebbinghaus [1] experiment om minne och glömska. I experimentet så var mer än hälften bortglömt en dag efter inlärningsstillfället och mer än tre fjärdedelar efter en månad. Vidare visade han också att man kan motverka detta genom att repetera informationen.

Fördelen med spaced repetition över cramming tydliggörs först efter några dagar. Från en studie av McIntyre och Munson [2] framkom att när kunskapen testas dagen efter inläring, presterar cramming och spaced repetition någorlunda likvärdigt. Men när testet flyttas till en vecka efter inläring presterar spaced repetition mer än dubbelt så bra i kunskapsbevarande. En tolkning av detta är att korttidsskillnaderna är mindre tydliga, men under längre perioder blir de uppenbara till fördel av spaced repetition. Vidare bekräftas att spaced repetition ger bättre kunskapsbevarande än cramming i en omfattande meta-analys av Cepeda et al [13].

3.2 Active recall

Active recall innebär att man aktivt försöker återkalla kunskap från minnet till skillnad mot att passivt läsa om materialet. Detta visades, i en studie av Roediger och Karpicke [14], leda till bättre långtidsminne än upprepade läsningar. Vidare visade Rawson och Dunlosky [15] att repeterande av active recall förstärker långtidsminnet ytterligare.

3.3 Algoritmer för intervallberäkning

SM-2 är en schemalägningsalgoritm för spaced repetition av inlärningskort. Den publicerades av Woźniak [16] 1990, och har sedan dess utgjort grunden för flera av de mest populära SRS (Spaced Repetition System) applikationerna, exempelvis Anki [17]. Algoritmen använder sig av en EF (Easiness Factor) och fungerar enligt följande:

- Första intervallet: 1 dag.
- Andra intervallet: 6 dagar.
- Resterande intervall: Föregående intervall multiplicerat med EF.

EF har ett golv på 1.3 och ett startvärde på 2.5 och justeras per kort. Justeringen är baserat på användarens bedömning av svårighetsgrad på en sexgradig skala från lätt till svår. En låg upplevd svårighetsgrad höjer kortets EF, vilket ger längre intervall till nästa repetition, medan en hög svårighetsgrad sänker EF och ger kortare intervall. Vid fel svar återställs kortets progression till det första intervallet.

Det har skett nyare forskning kring alternativa SRS-algoritmer. Su et al. [18] presenterade 2023 algoritmen SSP-MMC, som modellerar minnet som en Markovprocess och använder

stokastisk optimering för att minimera antal repetitioner. Algoritmen har i simuleringar jämförts med fem andra SRS algoritmer, däribland en variant av SM-2. Den visade sig i dessa reducera antal repetitionstillfällen mot samtliga algoritmer med samma sannolikhet för att återkalla information. Då algoritmen är maskininlärningsbaserad är den mer komplex än SM-2.

4 Systemkonstruktion

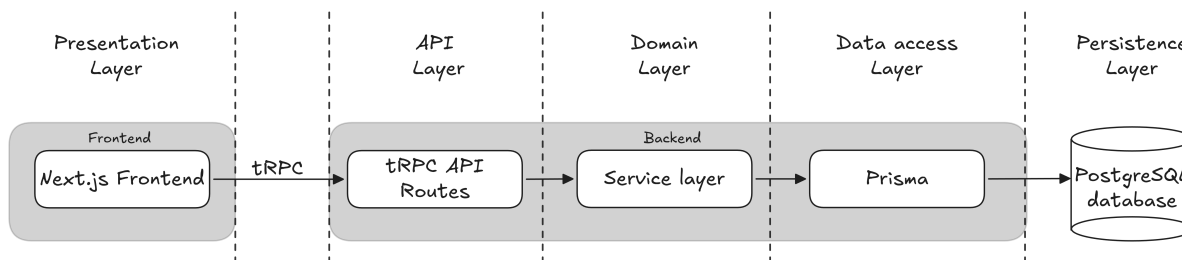
Detta avsnitt avser att beskriva relevanta delar av implementationen.

4.1 Ursprunglig design

Applikationens implementation har sitt ursprung i en design. Denna design bestod av en ursprunglig databas design, som finns i bilaga B, samt en visualisering av gränssnitt, som finns i bilaga A. Det ursprungliga gränssnittet har konsekvent utnyttjats genom implementation i en gränssnittsdriven utvecklingsprocess.

4.2 Övergripande arkitektur

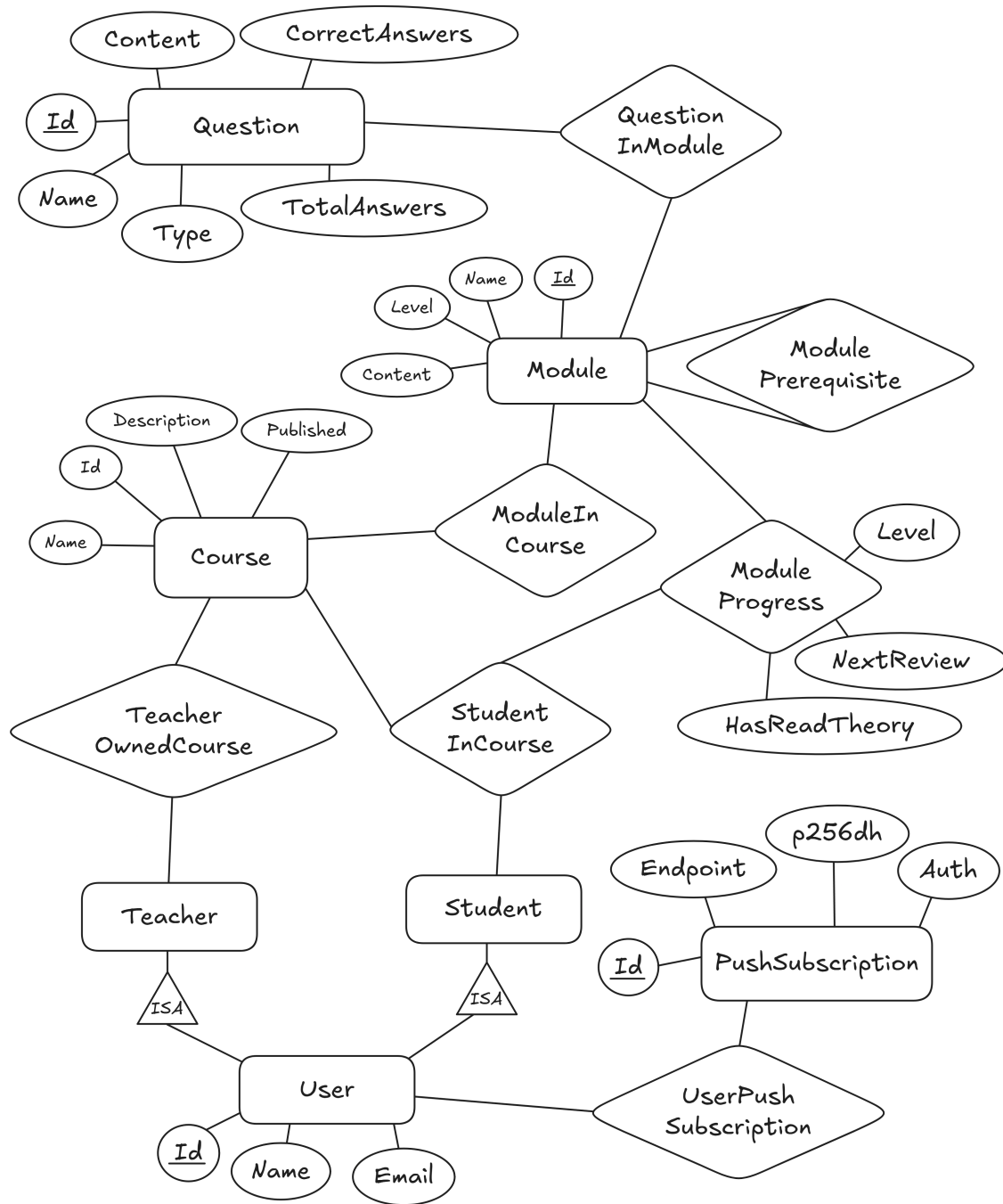
Applikationens arkitektur är uppdelad i frontend, backend och databas. Frontend och backend lever i samma kodbas men är tydligt separerade, strukturen visas i figur 1. Frontend kommunicerar med backend via tRPC vilket ger typsäkra anrop genom hela kedjan. I backend tar tRPC-routes emot API-anrop, validerar indata med zod, och delegerar sedan till ett servicelager. Servicelagret innehåller applikationens domänlogik, bland annat hantering av förkunskapskrav mellan moduler och beräkning av nästa repetitionstillfälle. Servicelagret använder Prisma för att läsa och skriva till PostgreSQL-databasen. Uppdelningen i routes, servicelager och databasåtkomst separerar domänlogiken från API-hantering och databasdetaljer, vilket gör koden lättare att underhålla.



Figur 1: Övergripande arkitektur av applikationen, skapad i Excalidraw

4.3 Databasdesign

Datamodellen är uppbyggd kring kurser, moduler och frågor. Ett ER-diagram av databasen illustreras i figur 2. En kursansvarig, som i databasen representeras av modellen Teacher, skapar kurser som består av flera moduler. Varje modul har ett fält för studiematerial och tillhörande frågor av tre typer: flerval, fritext och parbildning. Content-fälten i Module- och Question- tabellerna är lagrade som JSON (JavaScript Object Notation) eftersom innehållet kommer från TipTap. Moduler kan ha förkunskapskrav via en självrefererande många-till-många-relation. Förkunskapskraven är tänkt att bilda en DAG (Directed Acyclic Graph), valideringen av att inga cykler skapas görs i applikationslagret och beskrivs i avsnitt 4.7.



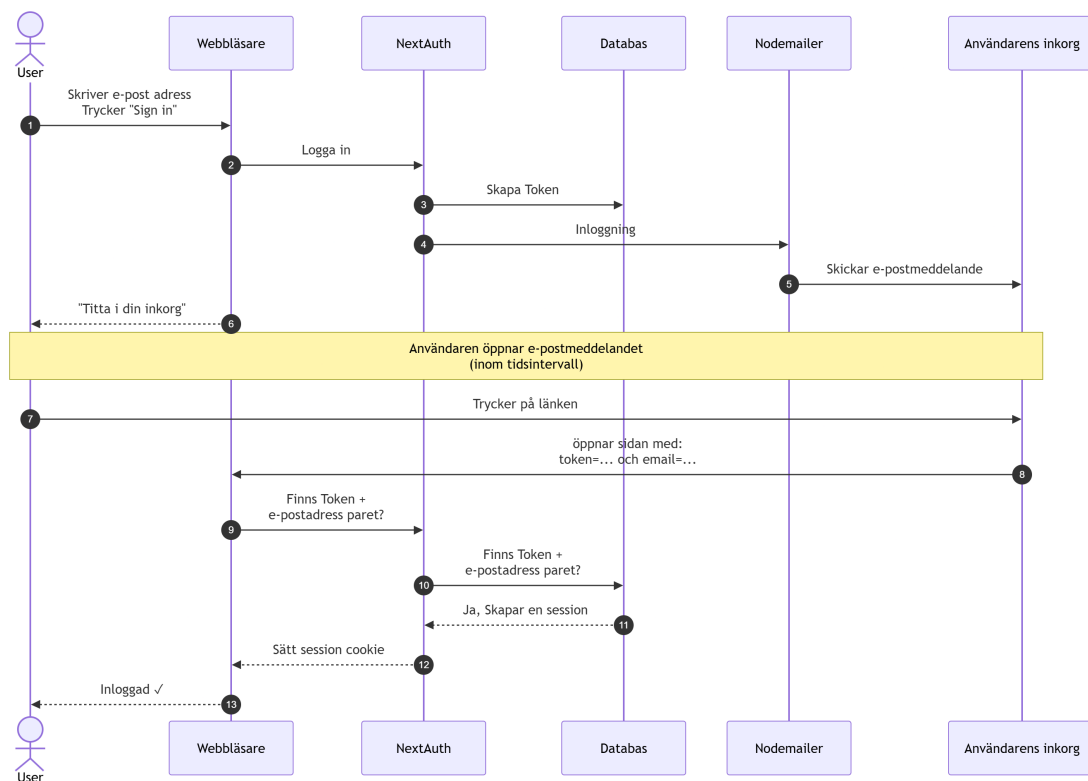
Figur 2: ER-diagram över systemets databasschema, skapat i Excalidraw

Databasen har stöd för studenter att gå med i kurser, avancera i moduler, och aktivera push-notifikationer. Studenter kopplas till kurser via StudentInCourse. Varje students progression i en modul spåras i ModuleProgress. Tabellen lagrar om studenten har läst studiematerialet, studentens nivå för modulen och nästa schemalagda repetitionstillfälle. PushSubscription lagrar Web Push-prenumerationer, de möjliggör för applikationen att skicka notifikationer via Web Push API:n till användarna. Databasen har även en uppsättning tabeller som används för inloggning och autentisering, dessa har utelämnats ur ER-diagrammet då de genereras automatiskt av T3 vid projektinitialisering och ej behandlats utöver detta.

4.4 Inloggning och autentisering

Inloggning på applikationen sker enbart via e-post. Användaren anger sin e-postadress för att logga in eller skapa ett konto. NextAuth genererar en tidsbegränsad token som hashas och lagras i databasen. Användaren får sedan ett e-postmeddelande via Nodemailer med den ohashade token i en länk.

Om användaren klickar på länken innan tidsbegränsningen är slut så matchas token och mejladress med en tabell i databasen. och om en matchande användare finns så loggas de in, annars skapas ett nytt konto. Vid inloggning skapas en session token som en cookie i användarens webbläsare. Detta gör att de förblir inloggade när de öppnar applikationen igen. Detta flöde beskrivs förenklat i figuren 3.



Figur 3: En simplifierad version av inloggningsflödet, skapad i Mermaid

Servern använder sessioner för att kontrollera åtkomst till skyddade endpoints. Servern verifierar att användaren är inloggad som rätt roll, student eller kursansvarig, innan åtkomst till skyddade endpoints tillåts.

4.5 Spaced repetition-algorithm

Systemets intervallberäkning bygger på SM-2. Som beskrivs i avsnitt 3.3 erbjuder SSP-MMC högre effektivitet i form av samma kunskapsbevarande med färre repetitioner, men med högre implementationskomplexitet. Den högre effektiviteten bedömdes inte motivera den extra komplexiteten för detta projekt. SM-2 använder en 6-gradig skala för att välja en EF mellan 1.3 - 2.5, då detta system använder binär bedömning av svar tillämpas en fast EF. Systemet har 11 nivåer av inläring en modul kan vara.

4.5.1 Inlärningsfas (nivå 0-3)

SM-2 saknar modell för den initiala inlärningsfasen. Dess första repetitionstillfälle är schemalagt för 24 timmar efter inläring. Ebbinghaus experiment visade dock att 24 timmar efter initialt inläringstillfälle är majoriteten av information bortglömt. För att kompensera för detta har nivåer för inlärningsfasen skapats.

Nivå 0 (1 timme) är den lägsta nivån och den initiala testningen som sker direkt efter att studenten läst modulens material. Nivå 1 (1 timme) och nivå 2 (8 timmar) placeras tidigt för att motverka den snabba initiala minnesförlusten som Ebbinghaus beskriver. Nivå 3 (1 dag) motsvarar SM-2:s första definierade intervall.

4.5.2 Mellansteg (nivå 4)

En fast EF erbjuder ingen möjlighet att kompensera med kortare intervall för svårare moduler. Därav introduceras ett mellansteg vid 3 dagar för att gardera mot att kunskap glöms bort innan det andra SM-2 intervallet.

4.5.3 SM-2 progression (nivå 5-10)

Från nivå 5 och framåt följer intervallen SM-2:s formel: $I(n) = I(n - 1) \times EF$. EF sattes till 2.0, ett mer konservativt val än standard 2.5 för att motsvara material av medelhög svårighetsgrad. För att resterande intervall ska vara mer naturliga kalendersteg av hela veckor så avrundades intervallet i nivå fem till en vecka i stället för sex dagar. Detta resulterar i en effektiv EF av ~ 2.33 vilket ligger inom SM-2:s normala intervall på 1.3–2.5.

4.5.4 Alla nivåer

Intervallen för alla nivåer blir som följande:

Nivå	Intervall
0	1 timme
1	1 timme
2	8 timmar
3	1 dag
4	3 dagar
5	1 vecka
6	2 veckor
7	4 veckor
8	8 veckor
9	16 veckor
10	32 veckor

Tabell 1: Intervall per nivå

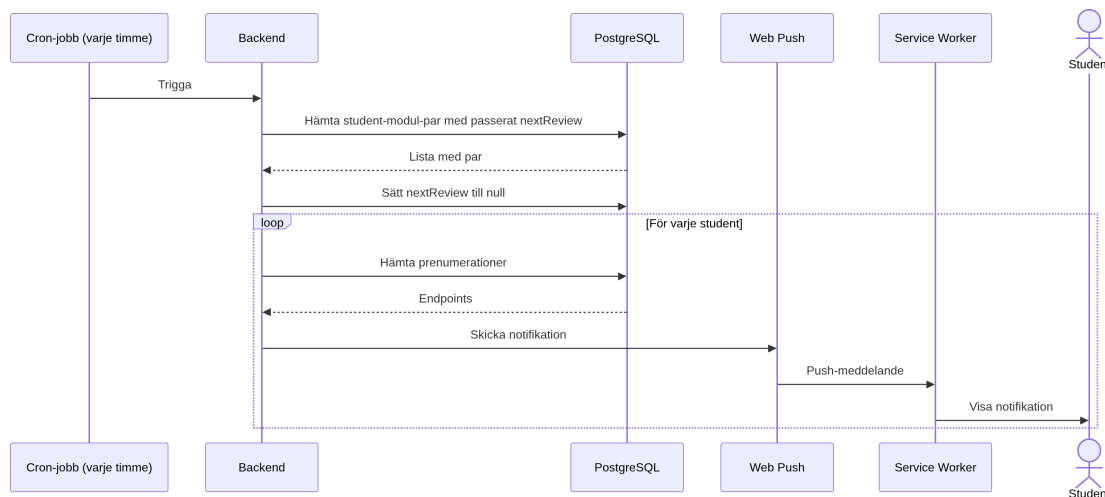
4.5.5 Hantering av felaktiga svar

Applikationen hanterar felaktiga svar annorlunda än SM-2. I SM-2 återställs progressionen till det första steget efter ett felaktigt svar. Istället reduceras SRS-nivån med ett steg, då urvalet av frågor kan ha en varierande svårighetsgrad.

4.6 PWA och notifikationer

För att underlätta för studenter att utföra repetitionstillfället på schemalagd tid är notifikationer ett användbart verktyg. För att implementera detta har applikationen utformats som en PWA (Progressive Web Application), vilket tillåter den att exekvera ett bakgrundsskript i användarens webbläsare. Web Push API:t används för att skicka notifikationer till användare genom deras webbläsare. Bakgrundsskriptet används sedan för att ta emot och visa push-notifikationerna. Detta fungerar även när applikationen inte är öppen i webbläsaren. När en student tillåter notifikationer sparas en push-prenumeration, vilket är vad som används för notifikationsutskick.

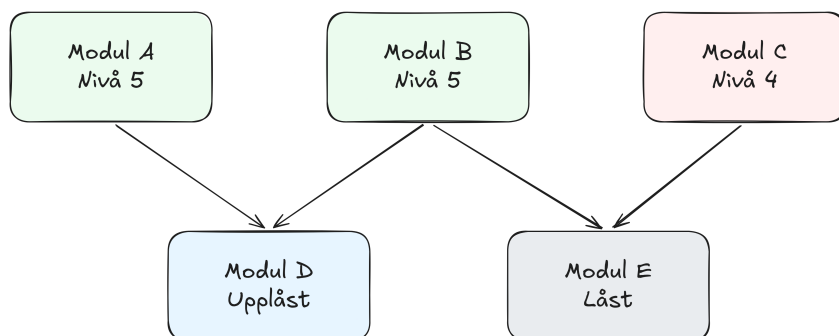
Ett cron-jobb har schemalagts att köra varje hel timme och hämtar alla par av studenter och moduler där det schemalagda repetitionstillfället har passerat. För studenter i dessa par skickas en notifikation att det nu finns moduler tillgängliga för repetition. En förklaring av denna sekvens visualiseras i 4. Som kan ses i tabell 1 så är det kortaste intervallet av repetitioner en timme, därav är frekvensen av cron-jobbet även satt till en timme.



Figur 4: Sekvensdiagram av cron-jobbet som skickar ut notifikationer, skapad i Mermaid

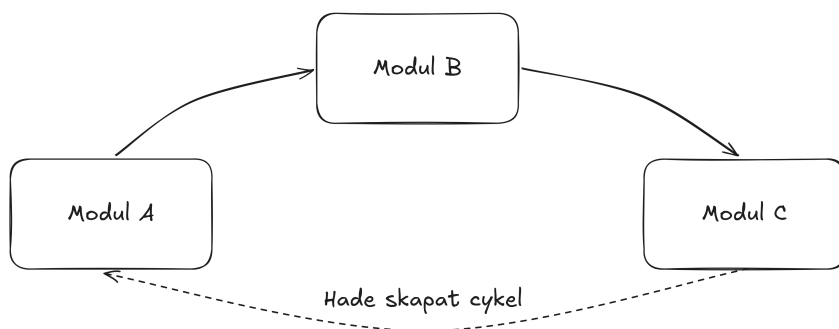
4.7 Förkunskapskrav och beroendehantering

Varje modul kan ha förkunskapskrav av andra moduler och anses då vara beroende på dem. En modul läses enbart upp när alla moduler som utgör dess förkunskapskrav är uppfyllda. Ett förkunskapskrav är uppfyllt när det nått åtminstone SRS-nivå fem och anses då inlärd nog. En illustration av detta visualiseras i 5. Kursansvarig kan skapa och ta bort dessa beroenden mellan moduler.



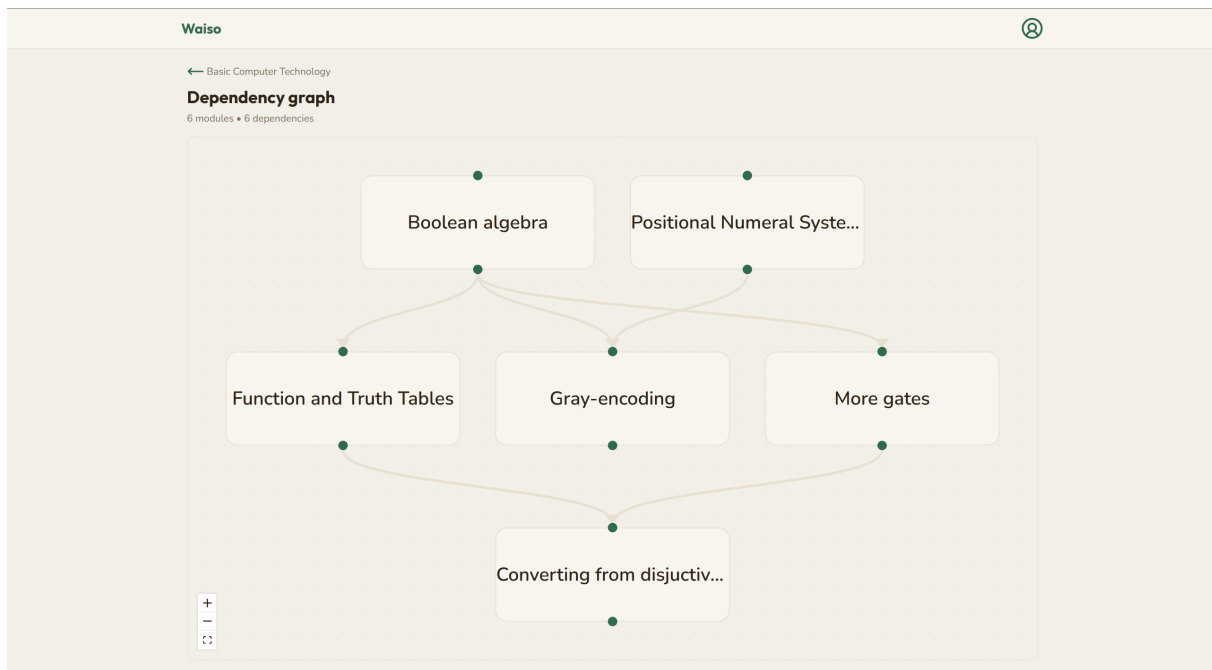
Figur 5: Modulrelationer, skapad i Excalidraw

Ett problem som kan uppstå är att cykler kan bildas vilket hade hindrat alla moduler i cykeln från att låsas upp. Ett sådant scenario visas i figur 6. För att motverka detta så modelleras beroendestrukturen som en DAG, där varje modul är en nod och beroende en riktad kant. När kursansvariga vill lägga till ett beroende byggs en graf upp från kursens moduler och deras befintliga beroenden. Med den modul som ska få ett nytt förkunskapskrav som startnod utförs en BFS (Breadth-First Search). Om den nod som ska bli förkunskapskrav går att nå hade en cykel skapats och beroendet nekas.



Figur 6: Problemet med cykel mellan moduler, skapad i Excalidraw

Beroendestrukturen visualiseras i en interaktiv graf där kursansvariga har möjlighet att se samt hantera förkunskapskrav. Grafen byggs upp genom att varje modul tilldelas en nivå baseras på dess avstånd från en modul utan förkunskapskrav. Moduler med samma nivå placeras horisontellt på samma rad. Kursansvariga kan skapa och ta bort beroenden direkt i grafen, varje ändring valideras med cykelkontrollen och nivåerna beräknas om automatiskt. Grafen renderas med React Flow, ett bibliotek för interaktiva grafvisualiseringar och visas i figur 7.



Figur 7: Vy med interaktiv graf

4.8 Funktionalitet och användarupplevelse

Kursansvariga och studenter har tillgång till olika vyer och funktionalitet av applikationen. Kursansvariga har ett gränssnitt designat för datorskärm med funktionalitet för skapande och redigering av kurser, medan studenters gränssnitt har designats för mobilanvändning med funktionalitet för inläring av kursmaterial. De gränssnitt som skapats för båda dessa användargrupper finns i helhet att beskåda i bilaga C.

4.8.1 Kursansvariga

Kursansvariga har tillgång till ett flertal funktioner för att möjliggöra skapande av kurser. För en skapad kurs kan moduler läggas till som ska omfatta ett koncept av kunskap. Varje modul kan sedan fyllas med det teoretiska innehållet för dess koncept, samt en uppsättning frågor i syfte att testa studenter på deras kunskap om konceptet. Förkunskapskrav för dessa moduler kan skapas både via modul gränssnittet samt i den interaktiva grafen. När en kurs är skapad har kursansvariga möjlighet att publicera den för studenter. Även efter publicering har de fortsatt möjlighet till redigering av dess innehåll.

4.8.2 Studenter

För en student är fokuset av applikationen istället inläringen av materialet i kurserna. De kan gå med i kurser och se vilka kurser de deltar i. I varje kurs har studenten möjlighet att läsa och repetera materialet för upplåsta moduler. När de läst materialet i en modul, kan de börja repetera material genom besvarande av dess frågor. Repetitionstillfällen tillgängliggörs utefter SRS-schemat samt deras nivå i modulen. Vid ett repetitionstillfälle går studentens SRS-nivå upp eller ner beroende på ifall de svarade rätt på första försöket. När en moduls förkunskapskrav är uppfylla låses den upp för studenten. Om ny-upplåsta moduler finns, kan studenter gå igenom deras material via en inlärningsknapp. På den

kursspecifika sidan kan studenter repetera material från den aktuella kursen, medan startsidan ger möjlighet att repetera material från alla kurser.

4.9 Designavvikelser

Då implementationen har genomförts med en gränssnittsdriven process finns enbart ett fåtal skillnader mellan den planerade designen och vad som i slutändan skapades. Den ursprungliga designen finns i bilaga A medans det slutgiltiga gränssnittet finns i bilaga C.

Den första av dessa avvikelser sker i hur repetitionerna visas för användarna. Ursprungligen planerades repetitionstillfällena visas som en lista av moduler. Detta ersattes med en repetitionsknapp för alla repetitionstillfällen samt en separat vy deras schemalagda tider.

Därefter finns en skillnad i hur modulers beroenden till varandra skapas. I den planerade modulvyn kunde förkunskapskrav enbart skapas för den aktuella modulen. Detta har kompletterats med möjligheten att även lägga till den aktuella modulen som förkunskapskrav till andra.

Slutligen finns det en skillnad i funktionaliteten av grafen över beroendestrukturen. Den ursprungliga planen var att grafen enbart ska visualisera relationerna utan någon interaktivitet. I den slutgiltiga applikationen har grafen kompletterats med funktionalitet att skapa och ta bort beroenden mellan modulerna.

4.10 Driftsättning för användartest

För driftsättning användes gratisplaner av ett flertal tjänster. Applikationen driftsattes på Vercel med en PostgreSQL-databas på Neon och bildlagring via UploadThing.

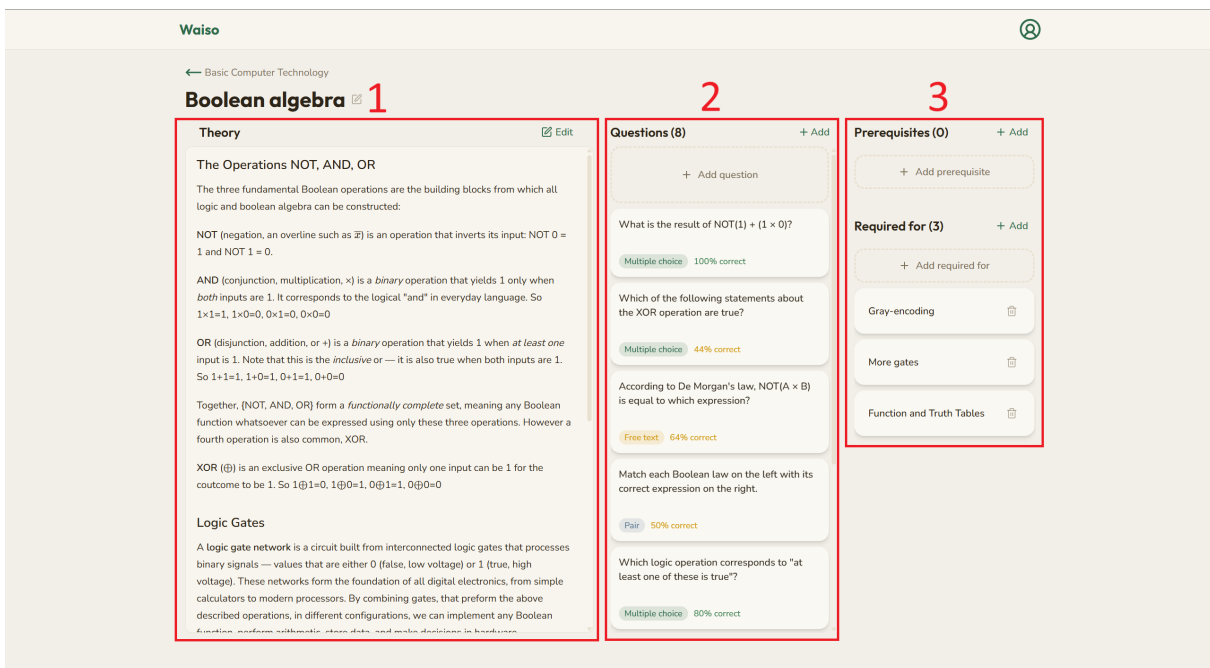
Applikationens notifikationslogik kräver ett cron-jobb som körs varje timme. Vercels gratisplan begränsar cron-jobb till ett minsta intervall på 24 timmar. Därav används den externa gratis tjänsten Cron-job.org, för att anropa systemets cron-endpoint. Anropet autentiseras med en hemlig nyckel för att förhindra obehörig åtkomst.

5 Resultat

Detta kapitel syftar till att presentera projektets resultat. Det förtydligar ifall all funktionalitet har implementerats samt beskriver resultaten av användarenkäten.

5.1 Applikationen

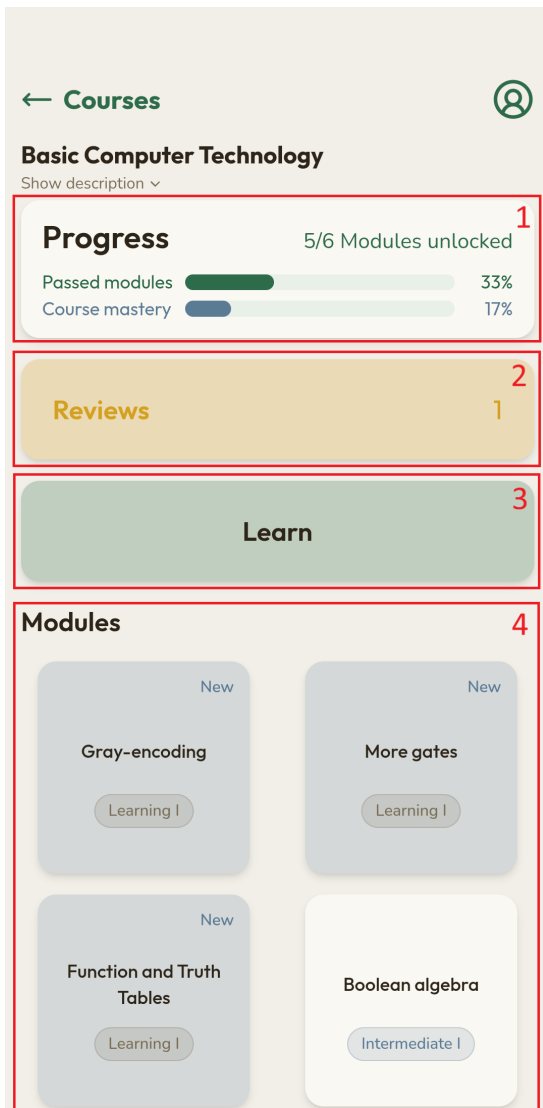
Den färdiga applikationen erbjuder ett flertal funktionaliteter. Kursansvariga har möjlighet att skapa, redigera och publicera kurser. Kurserna består av moduler där varje modul täcker ett koncept från kursen. Modulvyn för kursansvariga visas i figur 8 som är uppdelad i tre områden. I första området kan kursansvariga se och redigera teorin för konceptet. Vyn visar i det andra området upp en överblick av existerande frågor. Kursansvariga kan klicka på dem för att se och redigera, och de har även möjligheten att skapa nya frågor. Det kan även skapas förkunskapskrav mellan moduler vilket ger en relativ ordning för när de låses upp. Detta framträder i det tredje områden i form av två listor. I den övre listan visas förkunskapskraven för den aktuella modulen. I den nedre visas de moduler vars förkunskapskrav innefattar den aktuella modulen. Kursansvariga har även möjlighet att överblicka dessa modulrelationer i en grafvy, med en förmåga att förändra relationerna. Denna vy, samt applikationens resterande vyer finns i bilaga C.



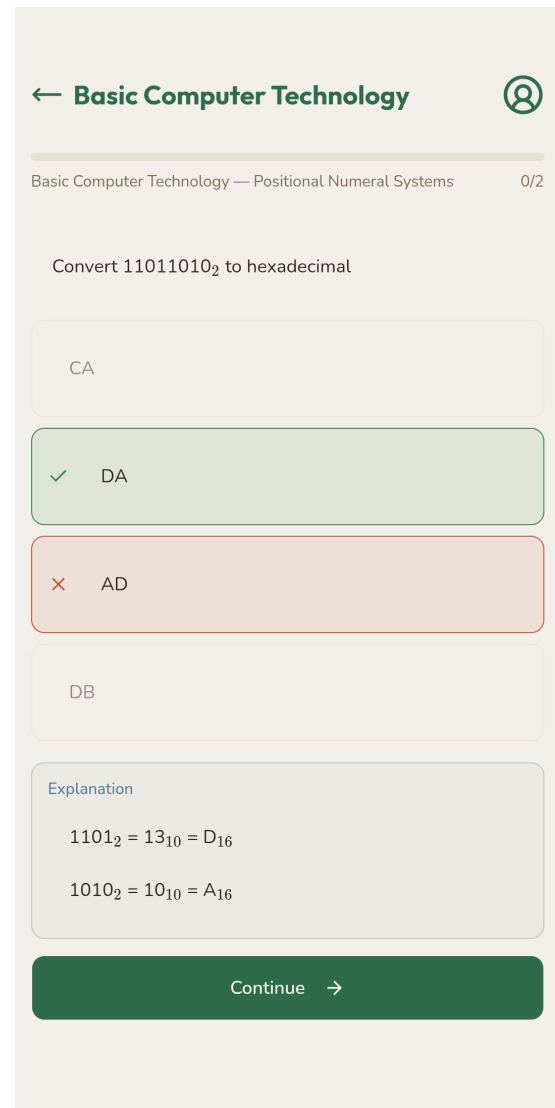
Figur 8: Kursansvarigas modulvy, uppdelad i: 1.Teori, 2.Frågor, 3.Förkunskapskrav. Redigerad i Paint.

Studenter har tillgång till andra delar av applikationen än kursansvariga. De kan se en översikt av kurser de deltar i och kan även gå med i nya kurser. I figur 9 visas studentens kursvy uppdelad i 4 områden. I det första området kan studenten se sin progression i kursen genom andel tillräckligt inlärd moduler samt andel av kursen totala nivåer som är avklarade, både presenteras som en progressbars. Repetitionsknappen, som finns i område två, kan klickas ifall det finns en tillgänglig repetition bland kursens moduler. Knappen tar då studenten till frågevyn där de kan besvara frågor om modulerna. Efter studenten

besvarat en fråga får de återkoppling på svarets korrekthet samt en förklaring, vyn för en besvarad fråga visas i figur 10. Om frågan är fel besvarad behöver studenten fortsatt svara på frågor från modulen tills de svarat korrekt. Om studenten har moduler med oläst teori framträder inlärningsknappen i det tredje området. Denna knapp leder studenten till en vy för att stega igenom de olästa modulernas teori. Alternativt har studenter möjligheten att läsa modulens teori genom att klicka på dem i kursvyns modulektion, som visas i område fyra. Repetitioner schemaläggs enbart för moduler vars teori har lästs.



Figur 9: Studenters kursvy, uppdelad i: 1. Progressionskort, 2. Repetitionsknapp, 3. Inlärningsknapp, 4. Modulektion. Redigerat i Paint



Figur 10: Studentvyn för en fel besvarad fråga

När en student går med i en kurs är enbart moduler utan förkunskapskrav tillgängliga. För att låsa upp en ny modul behöver studenten repetera de moduler som utgör dess förkunskapskrav tills de är uppfyllda, vilket sker när de nått åtminstone nivå 5 i dem alla. Nivån ökar när den första frågan från en modul besvaras korrekt vid ett repetitionstillfälle, annars sjunker nivån. I applikationen visas nivåerna som Learning I-V, Intermediate I-III, Advanced I-II och Mastered, där Intermediate I motsvarar nivå 5.

Repetitionsintervallen för varje nivå baseras på Ebbinghaus [1] forskning om forgetting curve samt SM-2 [16] algoritmen. En fullständig motivering av intervallen presenteras i avsnitt 4.5.

5.2 Enkäten

Ett användartest av applikationen utfördes med tillhörande enkät. Enbart studentupplevelsen har testats av 12 deltagare där 10 besvarade enkäten, vilket är en svarsfrekvens på ~83 procent. Några av de frågor som besvarats lyfts fram i detta avsnitt medan den fullständiga enkäten presenteras i tabell 2. Alla svar i enkäten är på en sexpoängsskala, där ett betyder dåligt och sex betyder bra. Undantaget till detta är Q_3 samt Q_5 där ett representerar för lite och sex för mycket.

	Fråga	μ	σ
Q_1	Hur skulle du betygsätta din övergripande upplevelse av applikationen?	4.7	1.1
Q_2	Hur enkelt var det att förstå hur applikationen fungerar utan förklaring?	5.1	0.7
Q_3	Hur upplevde du frekvensen av notifikationer? (1 är för lite, 6 är för mycket)	3.1	~1.44
Q_4	Hur hjälpsamma var notifikationerna för att få dig att repetera?	3.3	1.9
Q_5	Hur upplevde du tiden mellan repetitionerna? (1 är för sällan, 6 är för ofta)	3.4	~0.91
Q_6	Kände du att du kom ihåg modulernas innehåll bättre över tid tack vare repetitionerna?	5	~1.18
Q_7	Hur enkelt var det att navigera i applikationen?	5.2	~0.97
Q_8	Hur väl fungerade applikationen på din enhet?	6	0
Q_9	Hur motiverad kände du dig att fortsätta använda applikationen under testveckan?	3.9	~0.94
Q_{10}	Upplevde du att progressionsvisningen hjälpte dig att hålla motivationen uppe?	3.8	1.6

Tabell 2: Enkätresultat

Två frågor berörde användares interaktioner med notifikationssystemet för repetitions-tillfällen som blivit tillgängliga. Dessa frågor var Q_3 samt Q_4 vars resultat var $\mu = 3.1$ och 3.3 , $\sigma = \sim 1.44$ och 1.9 . För dessa frågor bör extra kontext tillföras att 20 procent av respondenter besvarat båda dessa frågor med 1, därtill lämnade de fritext svar om att de ej aktiverat notifikationer. En beräkning av Q_3 med dessa resultat exkluderat ger $\mu = \sim 3.6$ och $\sigma = \sim 1.11$ och för Q_4 med dem exkluderade en $\mu = \sim 3.8$ och $\sigma = \sim 1.69$.

Ännu en gruppering av frågor kan skapas med de fyra frågor som berör användarens upplevelse av applikationen. Frågorna Q_1 , Q_2 , Q_7 samt Q_8 undersöker alla vad användarna anser om applikationen efter användning. Med μ på 4.7, 5.1, 5.2, samt 6.0 som alla är

värden i den översta kvartilen av svarsintervallet. De tillhörande σ för dessa frågor var 1.1, 0.7, ~ 0.97 samt 0, vilket alla är starkt positiva svar.

En sista gruppering av frågor har skapats där Q_6 och Q_{10} båda undersöker användarnas upplevda förmåga att bättre bevara kunskap. Dessa frågor har $\mu = 5$ samt 3.8 och $\sigma = \sim 1.18$ samt 1.6. Som beskrivs i kapitel 3.1 är det primära syftet av spaced repetition att förbättra den långsiktiga förmågan att bevara kunskap, vilket den 1 vecka långa provotiden ej testat.

Det finns även ett flertal övriga fritextsvar som ej visas, men som vid relevans nämns i senare avsnitt.

6 Diskussion

Detta kapitel syftar till att diskutera resultaten av projektet. Huvudsaklig diskussion kommer centrera kring användarenkäten och huruvida kunskapsbevarandet förbättrades av applikationens användning. Enkätresultaten, se tabell 2, som diskuteras nedan baseras på tio svar efter en veckas användning och bör tolkas med detta i åtanke, svar är värden på en sexpoängsskala.

6.1 Modulbaserad kontra kortbaserad repetition

Många SRS applikationer är kortbaserade. Dessa testar användare på kort som ofta innehåller ord eller fraser för att förbättra studerandes kunskapsbevarande genom active recall av korten. Denna applikation testar istället koncept från moduler med hjälp av frågor. Risken med denna förändring är att studenter lär sig svaren på frågorna istället för de underliggande koncepten. För att motverka detta bör varje modul inkludera ett flertal frågor med förhoppningen att de lär sig koncepten istället för svaren. För att kunna avgöra om det faktiskt leder till bättre kunskapsbevarande än kortbaserad repetition hade det dock krävt en betydligt mer omfattande studie. Enkätresultaten tyder på att deltagarna upplevde en positiv effekt, fråga Q_6 om huruvida de kom ihåg modulernas innehåll bättre över tid tack vare repetitionerna fick ett μ på 5.0 samt σ på ~ 1.18 .

En förutsättning för att strategin ska fungera som tänkt är att studenter inte får samma fråga flera gånger i rad. Dessvärre kommenterade en deltagare i enkäten att samma fråga dykt upp två dagar i rad. Då systemet väljer frågor slumpmässigt finns det en risk för att detta sker, speciellt för moduler med få frågor. En förbättring vore att spåra de senaste frågorna en student besvarat i varje modul och exkludera dessa från nästa repetitionstillfälle, därav hade variation garanterats.

6.2 SRS-algoritm

Systemets implementation av SM-2 använder sig av en binär bedömningskala istället för den sexgradiga skalan i originalet. Detta har lett till att systemet saknar samma underlag som originalet använder för att tillämpa ett dynamiskt EF. Utan dynamiskt EF behandlar systemet alla moduler som lika svåra, däremot har användningen av en statisk EF förenklat implementationen. Då systemet sparar statistik för andel korrekta svar för frågor kunde använts för att sätta en svårighetsgrad för frågan. Denna svårighetsgrad hade kunnat användas för att tillämpa en dynamisk EF för schemaläggningen. Alternativ kunde den sexgradiga bedömningsskalan som originalet använder implementerats.

Enkätrespondenterna besvarade en fråga kring upplevd tid mellan repetitionstillfällena. Fråga Q_5 fick ett μ på 3.4 och σ av ~ 0.91 på en skala från för kort till för lång, vilket är en någorlunda låg spridning. Resultatet är nära 3.5, vilket är medelvärdet för poängskalan, som tyder på att intervallen upplevdes som rimliga av deltagarna.

6.3 Notifikationer

Systemet skickar notifikationer för att underlätta för studenter att följa SRS schemat. Notifikationernas frekvens upplevdes som lagom då Q_3 hade ett μ på 3.1 vilket är nära 3.5, men då spridningen är hög bör detta värde tolkas försiktigt. Frågan om notifikationernas

hjälpksamhet Q_4 fick neutralt μ med en mycket hög spridning. Som beskrevs i resultatet beror en del av spridningen på att deltagare utan aktiverade notifikationer gav lägsta betyg. Det tyder på att notifikationerna hade en viss positiv effekt för de som använde dem, men trots att applikationen uppmanar användaren att aktivera notifikationer vid första inloggning valde vissa deltagare att inte göra det.

Under driftsättningen observerades att cron-jobbet för notifikationer misslyckades vid anrop på hela timmar. Detta på grund av långa svarstider från databasen som överskred tidsgränsen för Vercels serverlösa funktioner. Serverlösa funktioner körs på begäran och har en begränsad maximal körtid, vilket innebär att funktionen avbryts om inte databasen svarar i tid. Detta orsakades troligen av att Neons databas går i viloläge efter fem minuter av inaktivitet på gratisplanen i kombination med hög belastning vid hela timmar. Problemet löstes genom att schemalägga cron-jobbet en minut före varje hel timme, vilket konsekvent resulterat i betydligt kortare svarstider och lyckade cron-jobb.

6.4 Motivation och progression

Flera observationer från enkäten pekade på att applikationen saknar transparens kring progressionen. Det framgick inte för användaren vad nivåskillnaderna innebär och det saknades information om när kommande moduler läses upp. Applikationen presenterar i nuläget ingen förklaring om hur algoritmen och intervallen fungerar för användarna, vilket kan ha påverkat upplevelsen av progression. Avsaknaden av denna förklaring kan vara en bidragande faktor till att resultaten på frågan om hur progressionsvisningen hjälpt motivera studenterna, Q_{10} , haft en stor spridning. En σ på 1.6 som är större än 40 procent av μ på 3.8, visar på en stor instabilitet i svaren från respondenterna. Den frånvarande förklaringen av syftet med applikationen kan även ha varit en faktor i ett potentiellt lägre μ på 3.9, samt $\sigma = \sim 0.94$, för fråga Q_9 som frågat hur motivationen under testveckan varit.

6.5 Användbarhet

Applikationen fick höga betyg på användbarhet. Fråga Q_7 om navigering samt Q_2 om intuitiv förståelse fick båda μ över 5 med någorlunda låg spridning i svaren. Alla deltagare gav högsta betyg på fråga Q_8 om hur väl applikationen fungerade på deltagarnas enhet. I fritextsvar framkom ett önskemål om anpassad layout för liggande skärm, vilket är förståeligt då studentvyn har designats med stående mobilvy i åtanke. Det tillkom även ett flertal önskemål om en flik för mer information om applikationens funktionalitet. De primära önskemålen, som tidigare nämnts i det föregående stycket, var kring hur progressionen av nivåer fungerade och vad syftet med applikationen var.

6.6 Enkätens begränsningar

Användartestet genomfördes med 12 deltagare under en veckas tid, av dem besvarade 10 enkäten. En veckas testperiod innebär att deltagarna som högst kunde nå nivå 5 om de klarade repetitionen efter intervallet på tre dagar. De längre intervallen, veckor och månader, är helt oprövade i utvärderingen. För att utvärdera applikationens faktiska effekt på kunskapsbevarande hade det krävts en betydligt längre studie med fler studenter, som exempelvis läser en riktig kurs. En sådan studie hade kunnat jämföra tentamensresultat

mellan studenter som använt och icke använt applikationen. Men framförallt genomföra ett uppföljningstest för att mäta långsiktigt kunskapsbevarande.

6.7 Vad hade gjorts annorlunda

Det var flera kommentarer i enkäten om presentationen av teorimaterialet. De upplevde att materialet presenterades som en vägg av text och att det hade behövt bättre formatering eller illustrationer. Teorimaterialet skapades enbart för användartestet och utnyttjade inte den fulla funktionen av textredigeraren, det finns därav redan möjlighet för kursansvariga att formatera deras material på ett mer pedagogiskt sätt. En mer grundläggande reflektion är om applikationen överhuvudtaget bör innehålla kursteori. I praktiken hade det möjligtvis varit bättre att begränsa modulernas teoridel till enbart läshänvisningar. Applikationen hade då kunnat fokusera fullständigt på repetitionsfrågorna istället för att försöka vara en plattform för både kursmaterial och repetition, vilket även hade minskat arbetet för kursansvariga.

Applikationen hade gynnats av riktlinjer vid kursskapande. Exempelvis hade rekommendationer om antal frågor per modul samt en moduls lämpliga omfattning varit gynnsamma. En modul bör motsvara ett enskilt koncept från kursen, inte ett helt kursavsnitt. Då ligger modulerna närmare det kortbaserade format som forskningen utgår från.

6.8 Etiska aspekter

Projektet syftar till att underlätta studier för studenter. Genom att erbjuda forskningsbaserad repetition kan applikationen bidra till bättre förutsättningar för studenter att lära sig, oberoende av deras förkunskaper om effektiva studiemetoder.

6.9 Miljöaspekter

Det finns även ett fåtal miljöaspekter som kan begrundas. Valet av att driftsätta applikationen för användartesting via en online-tjänst har medfört energikostnader. Även valet av vilken tjänst som använts för driftsättning har medfört en kostnad. Då Vercel ej erbjuder en databaslösning och en tredje part istället utnyttjats har medfört även detta en energikostnad. Användningen av hemsidan Website carbon [19] beräknar att en användning av applikationen skapar 0.03g av CO_2 , hemsidan har 12 maj 2026 haft 68 besökare vilket med Website carbons uppskattning hade resulterat i ett totalt utsläpp av 2.04g CO_2 . Vidare använder sig Vercel av serverlösa funktioner vilket resulterar i en lägre energikostnad jämfört med en dedikerad server som körs kontinuerligt.

7 Slutsats

Projektet har överlag varit framgångsrikt, med alla implementationsmål uppfyllda. De specificerade funktionaliteterna är testade av användare, men det går inte att dra några slutsatser om projektet faktiskt förbättrar kunskapsbevarandet av studenter. Detta beror på den begränsade testperioden inte lämnat rum för testning av långvariga minneseffekter. Under projektets gång har även ett flertal förbättringar till applikationen insetts.

7.1 Utvärdering av arbetsprocess

Den gränssnittsdrivna utvecklingsprocessen fungerade väl i praktiken. Genom att utgå från gränssnittet implementerades enbart de backend-endpoints och servicefunktioner varje vy faktiskt behövde. Det minskade även behovet att i efterhand justera redan implementerade endpoints, vilket annars kan uppstå när backend och frontend utvecklas separat. Processen underlättade även parallellt arbete då olika vyer kunde fördelas och utvecklas oberoende av varandra. En nackdel var att de detaljerade gränssnittsdesignerna var tidskrävande att ta fram innan implementationen kunde påbörjas.

7.2 Projektets styrkor och svagheter

Projektets största styrka är kombinationen av SRS med en kursbaserad struktur. Intervallen för SRS algoritmen är framtagna med stöd i forskning och nyttjas för att främja inlärandet av kursmaterial. Utöver detta har applikationens användbarhet och gränssnitt fått höga betyg från användare.

Projektet har dessvärre även svagheter. Den huvudsakliga svagheten är avsaknaden av information till användarna om applikationens funktion och syfte. Det saknas förklaringar om; vad spaced repetition är, varför moduler har nivåer, hur nivåer av moduler går upp, hur man låser upp nya moduler eller varför det hjälper med långvarig kunskapsbevaring. Denna information hade förbättrat användarnas upplevelse av applikationen, och var den mest kommenterade förbättringen i enkäten.

Slutligen kan kritik även riktas mot enkäten. Det primära målet med att använda sig av spaced repetition är att förbättra förmågan att minnas kunskap under en längre tid, för detta borde ett längre test av applikationen ha utförts. Användartestet enkäten baserats på varade endast en vecka, för kort tid för att testa det långvariga minneseffekterna. Enkäten består även endast av 10 respondenter, vilket är ett för lågt antal för att forma slutsatser.

7.3 Vidareutveckling

Det finns ett flertal förbättringar i applikationen som kan introduceras vid fortsatt utveckling. Den mest uppenbara av dessa är en sida som förklarar SRS algoritmen, progressionen samt hur moduler läses upp.

En mer effektiv algoritm för schemaläggning kan implementeras. SSP-MMC är en nyare algoritm än SM-2, vilket systemets SRS-algoritm är baserad på, som i simulering resulterat i markant färre repetitionstillfällen med samma kunskapsbevaring. SSP-MMC ansågs

inte vara i projektets omfattning men vid fortsatt utveckling hade implementation varit önskvärd.

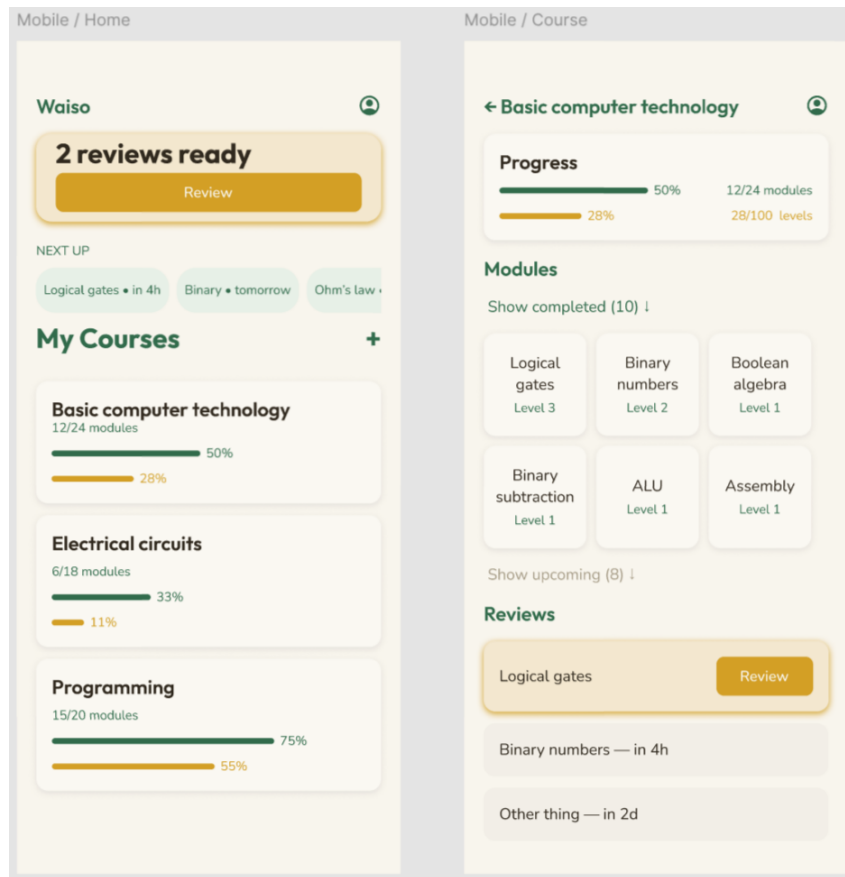
Slutligen kan en mer omfattande studie genomföras för att bättre testa funktionaliteten av applikationen. Studien bör pågå under en längre tid och inkludera fler studenter som är uppdelade i grupper som använder applikationen och som inte gör det. Mot slutet av studien bör studenterna testats på sin inläring, med ett uppföljnings test en lång tid efter studien för att testa kunskapsbevarandet. Kursansvarigas användarupplevelse har inte utvärderas, vilket även detta borde testas.

Referenser

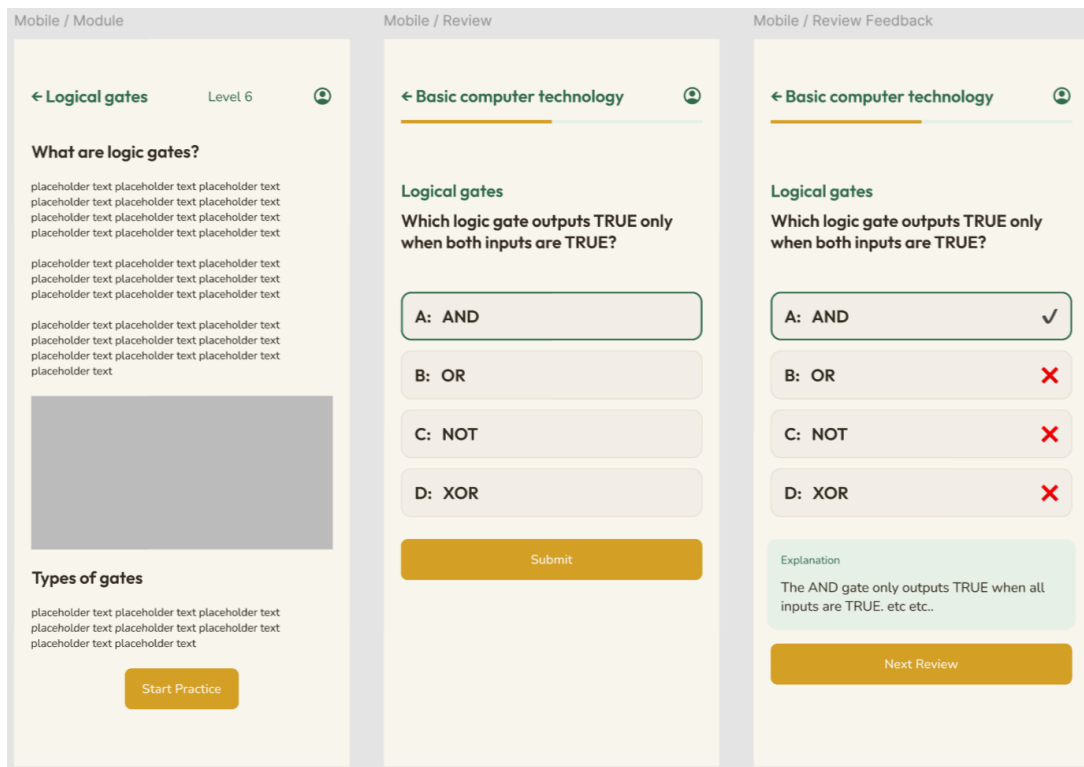
- [1] H. Ebbinghaus, *Memory: A Contribution to Experimental Psychology*, övers. av H. A. Ruger och C. E. Bussenius. Teachers College Press, 1913. DOI: 10.1037/10011-000.
- [2] S. H. McIntyre och J. M. Munson, "Exploring Cramming: Student Behaviors, Beliefs, and Learning Retention in the Principles of Marketing Course," *Journal of Marketing Education*, årg. 30, nr 3, s. 226–243, 2008. DOI: 10.1177/0273475308321819. eprint: <https://doi.org/10.1177/0273475308321819>. URL: <https://doi.org/10.1177/0273475308321819>.
- [3] S. H. K. Kang, "Spaced Repetition Promotes Efficient and Effective Learning: Policy Implications for Instruction," *Policy Insights from the Behavioral and Brain Sciences*, årg. 3, nr 1, s. 12–19, 2016. DOI: 10.1177/2372732215624708.
- [4] Create T3 App Contributors. "Create T3 App," hämtad 11 maj 2026. URL: <https://create.t3.gg/>.
- [5] Vercel. "Next.js," hämtad 11 maj 2026. URL: <https://nextjs.org/>.
- [6] tRPC contributors. "tRPC: Move Fast and Break Nothing. End-to-end Typesafe APIs Made Easy," hämtad 6 maj 2026. URL: <https://trpc.io/>.
- [7] "Prisma: Postgres, Perfectly Managed," Prisma Data, Inc., hämtad 6 maj 2026. URL: <https://www.prisma.io/>.
- [8] "Auth.js: Authentication for the Web," Better Auth Inc., hämtad 6 maj 2026. URL: <https://authjs.dev/>.
- [9] Z. contributors. "Zod: TypeScript-first schema validation," hämtad 11 maj 2026. URL: <https://zod.dev/>.
- [10] Tiptap. "Tiptap Editor," hämtad 11 maj 2026. URL: <https://tiptap.dev/>.
- [11] "Nodemailer," hämtad 11 maj 2026. URL: <https://nodemailer.com/>.
- [12] "Bun: A Fast All-in-One JavaScript Runtime," Anthropic, hämtad 6 maj 2026. URL: <https://bun.com/>.
- [13] N. J. Cepeda m. fl., "Distributed Practice in Verbal Recall Tasks: A Review and Quantitative Synthesis," *Psychological Bulletin*, årg. 132, nr 3, s. 354–380, 2006. DOI: 10.1037/0033-2909.132.3.354.
- [14] H. L. Roediger och J. D. Karpicke, "Test-Enhanced Learning: Taking Memory Tests Improves Long-Term Retention," *Psychological Science*, årg. 17, nr 3, s. 249–255, 2006. DOI: 10.1111/j.1467-9280.2006.01693.x.
- [15] K. A. Rawson och J. Dunlosky, "Optimizing Schedules of Retrieval Practice for Durable and Efficient Learning: How Much Is Enough?" *Journal of Experimental Psychology: General*, årg. 140, nr 2, s. 283–302, 2011. DOI: 10.1037/a0023956.
- [16] P. A. Woźniak, "Optimization of Learning: A New Approach and Computer Application," Supervisor: Prof. Zbigniew Kierzkowski. Available at <https://super-memory.com/english/ol.htm>, examensarb., University of Technology in Poznan, Poznan, Poland, 1990.

- [17] Ankitects Pty Ltd. "What Spaced Repetition Algorithm Does Anki Use?" Hämtad 6 maj 2026. URL: <https://faqs.ankiweb.net/what-spaced-repetition-algorithm>.
- [18] J. Su m. fl., "Optimizing Spaced Repetition Schedule by Capturing the Dynamics of Memory," *IEEE Transactions on Knowledge and Data Engineering*, årg. 35, nr 10, s. 10 085–10 097, 2023. DOI: 10.1109/TKDE.2023.3251721.
- [19] "Website Carbon Calculator," Wholegrain Digital, hämtad 6 maj 2026. URL: <https://www.websitecarbon.com/>.

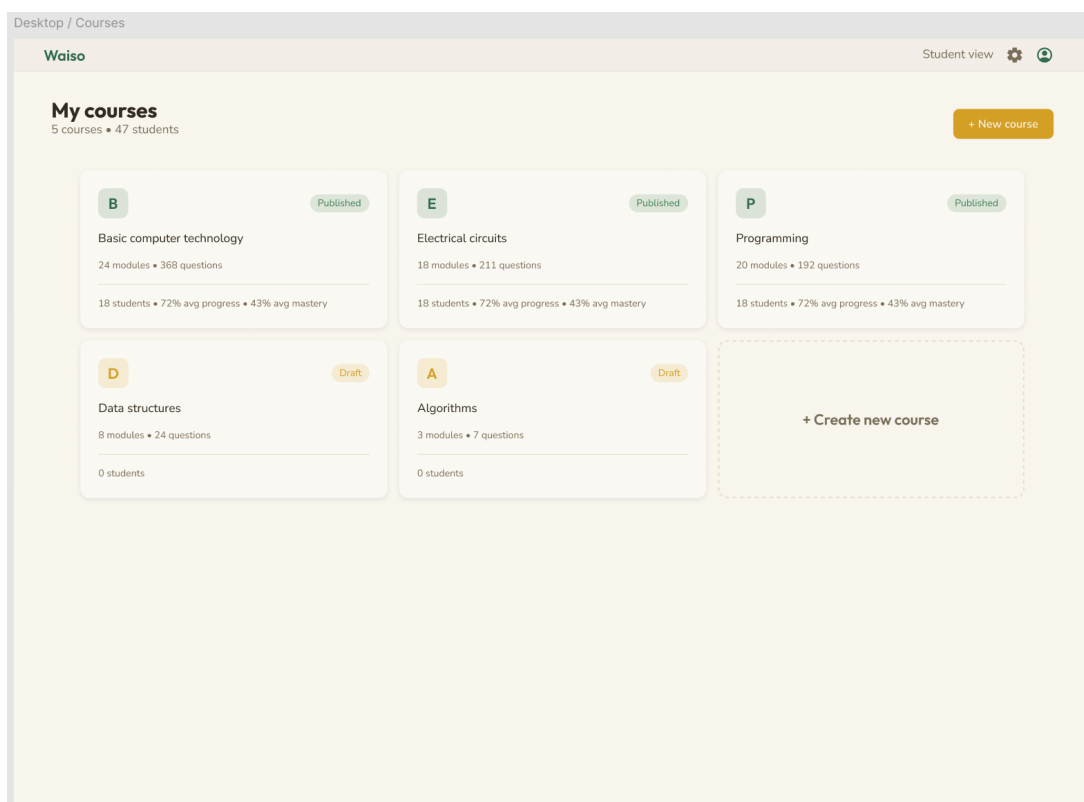
A Ursprunglig gränssnittsdesign



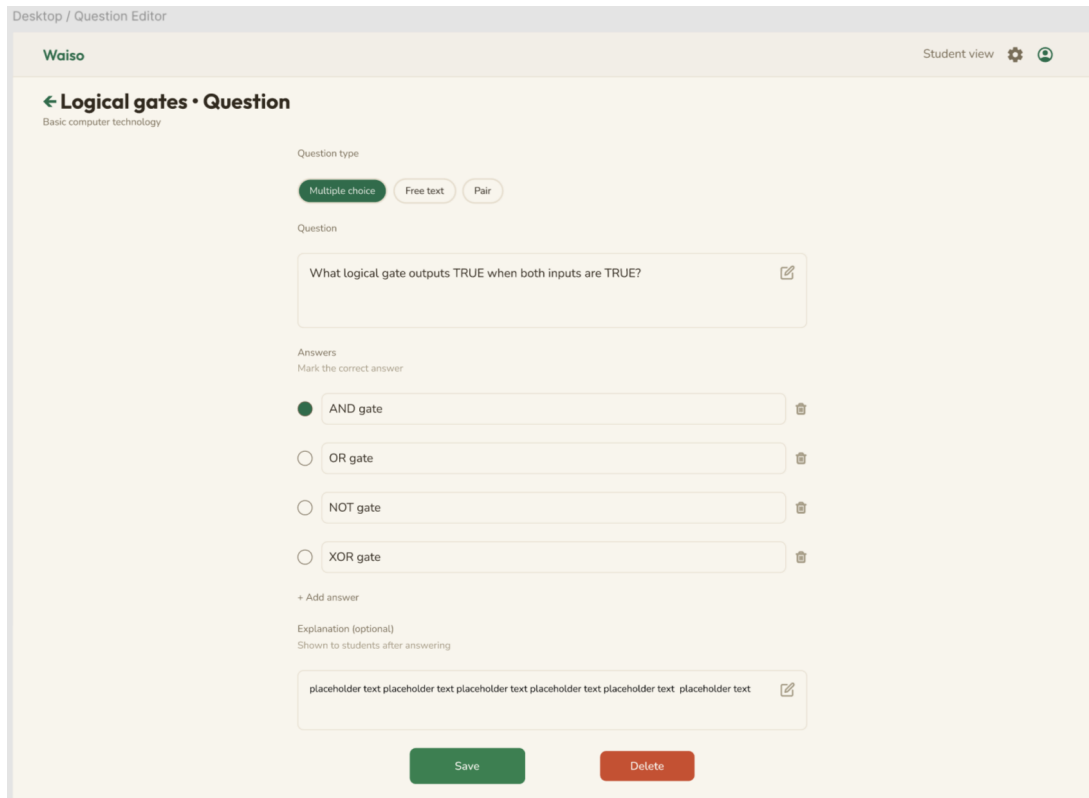
Figur 11: Hemskaerm och kursöversikt för studenter, skapad i Figma



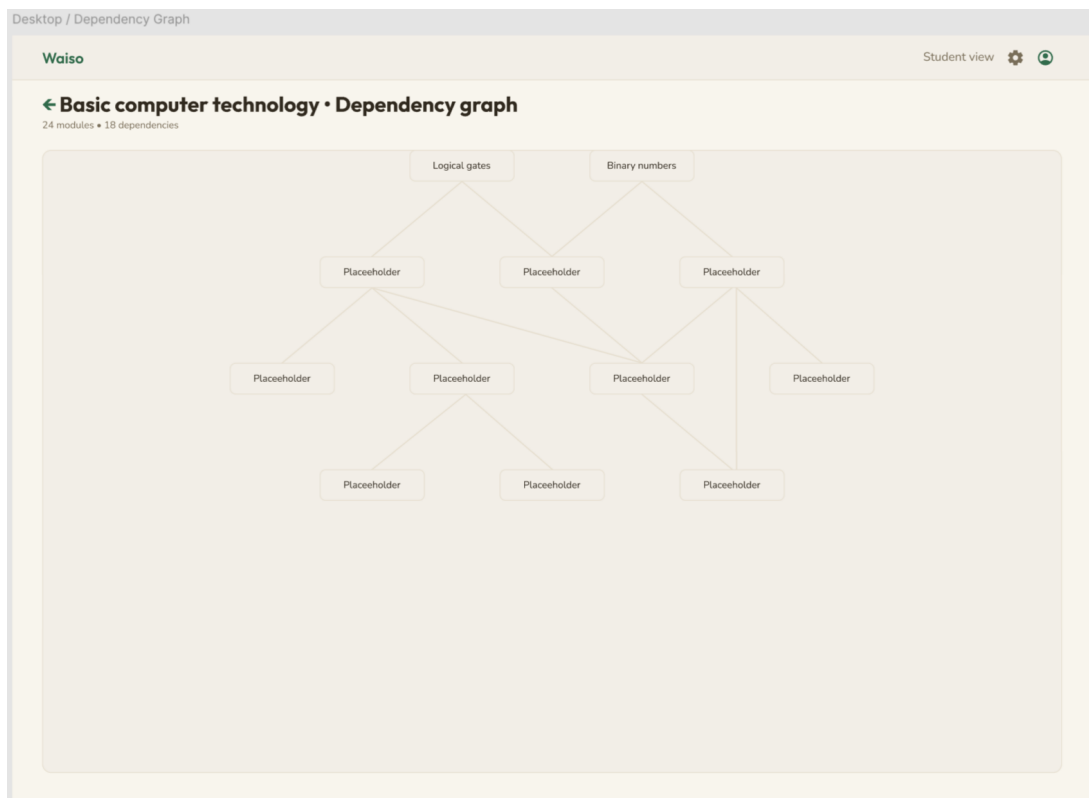
Figur 12: Modulöversikt och studeringsskärmar med, samt utan svar, skapad i Figma



Figur 13: Översikt av kurser skapade av kursansvarig, skapad i Figma

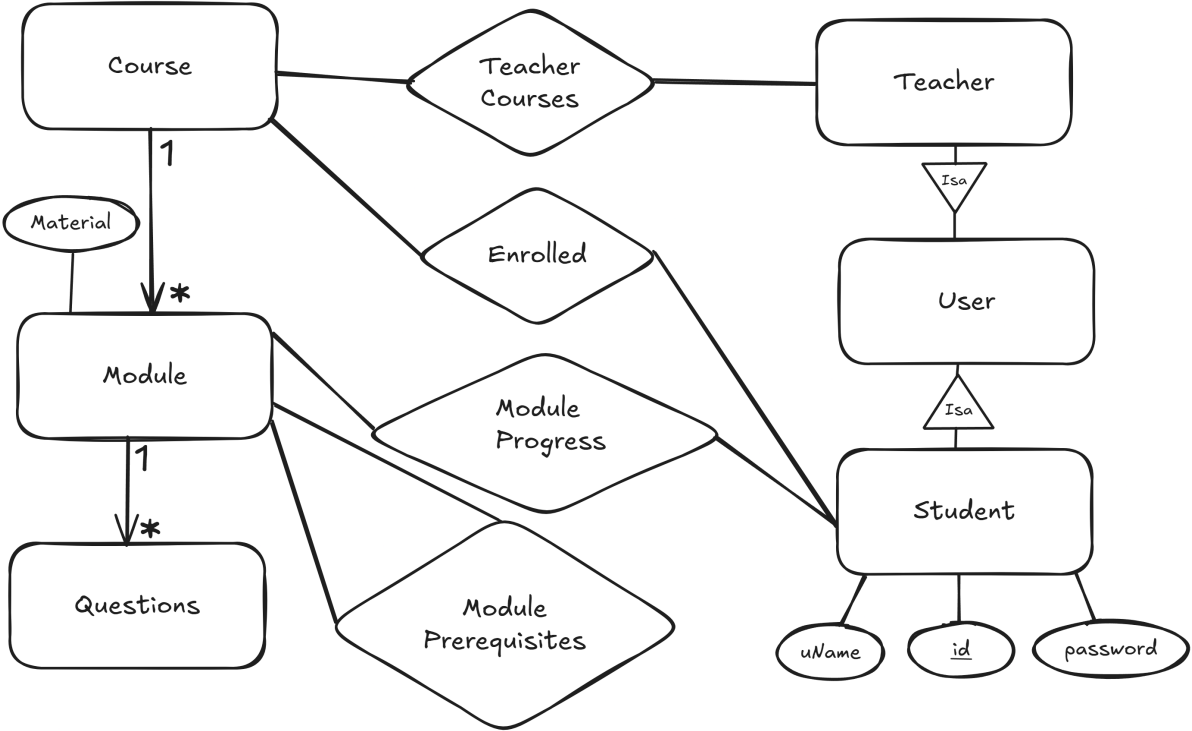


Figur 16: Frågevy för kursansvarig, skapad i Figma



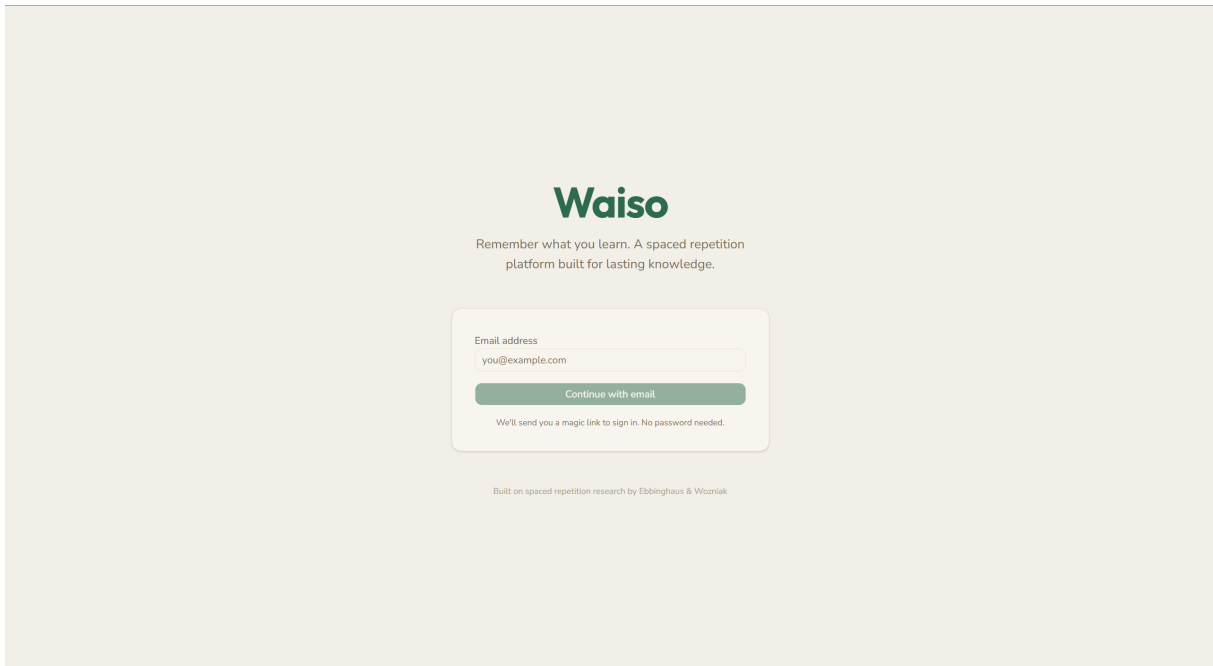
Figur 17: Graf av modulers beroende, skapad i Figma

B Ursprunglig databasdesign

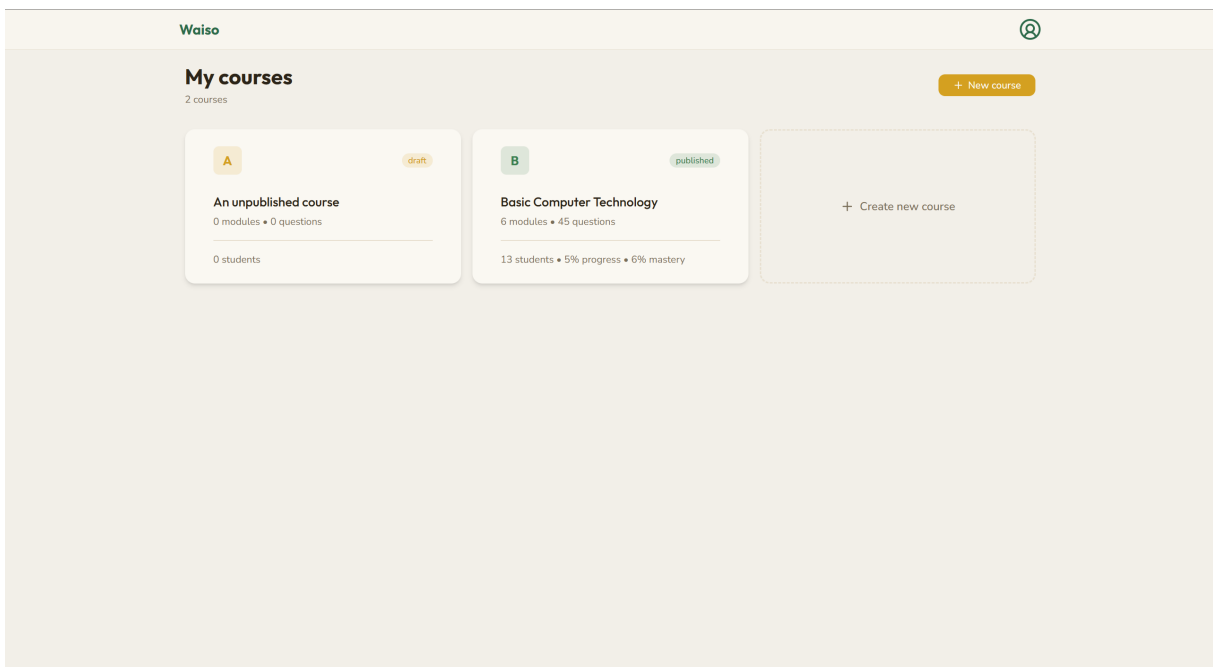


Figur 18: Ursprunglig databas design, skapad i Excalidraw

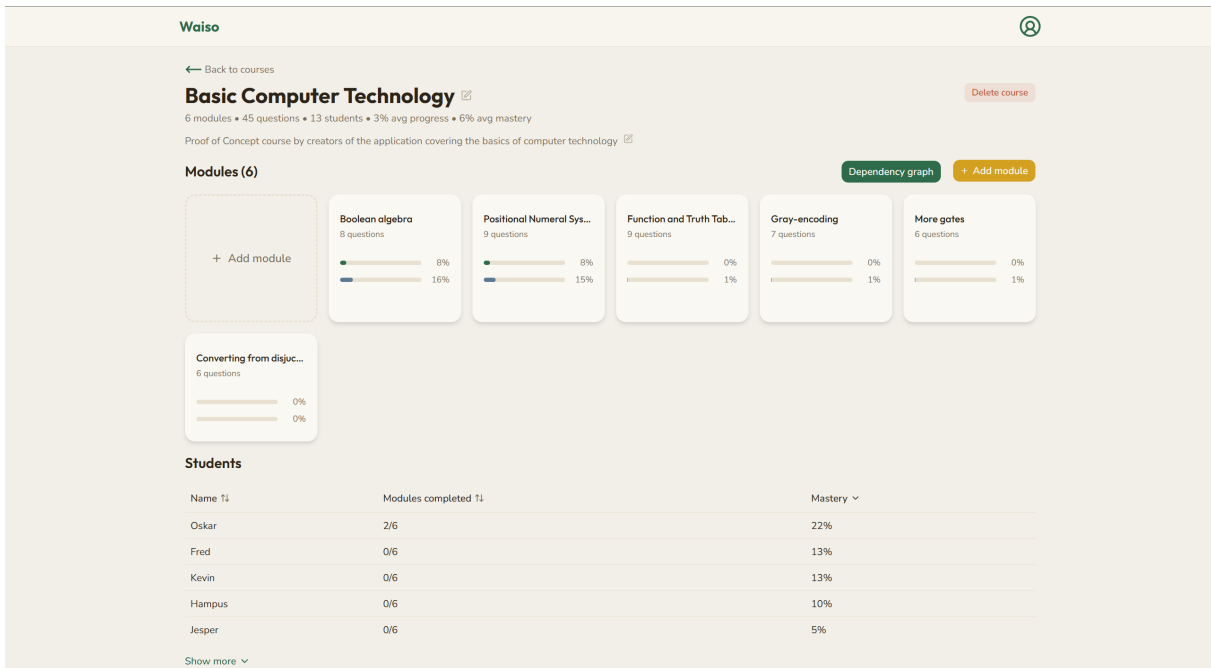
C Systemets utseende



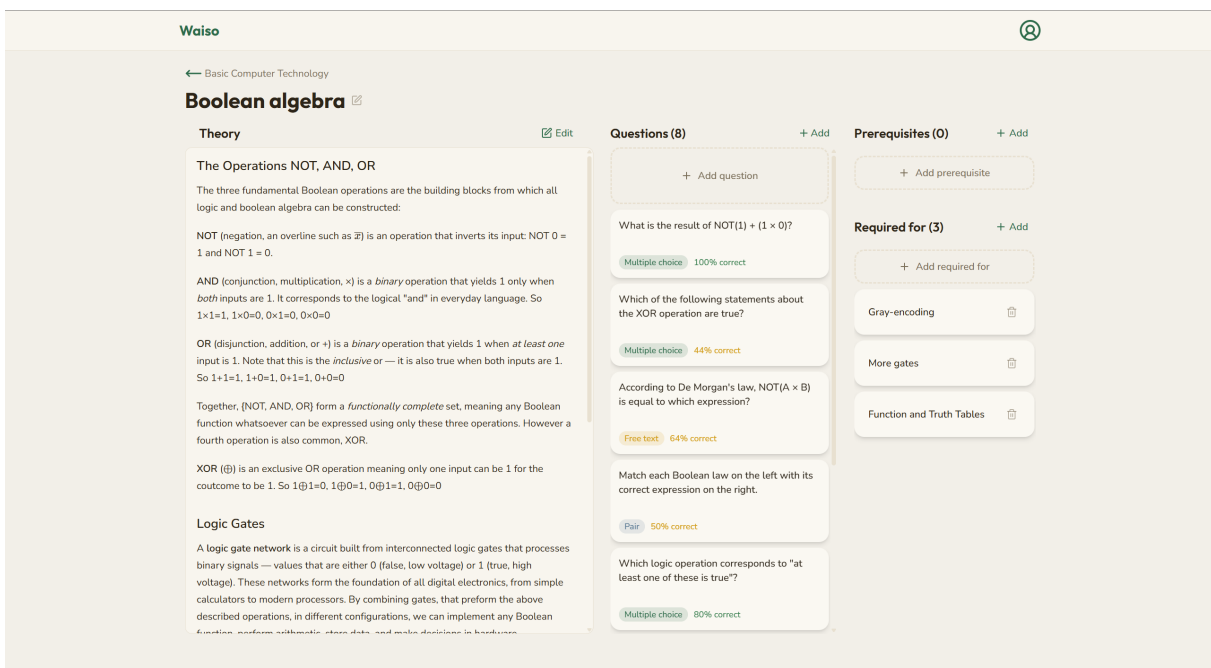
Figur 19: Startside för applikationen



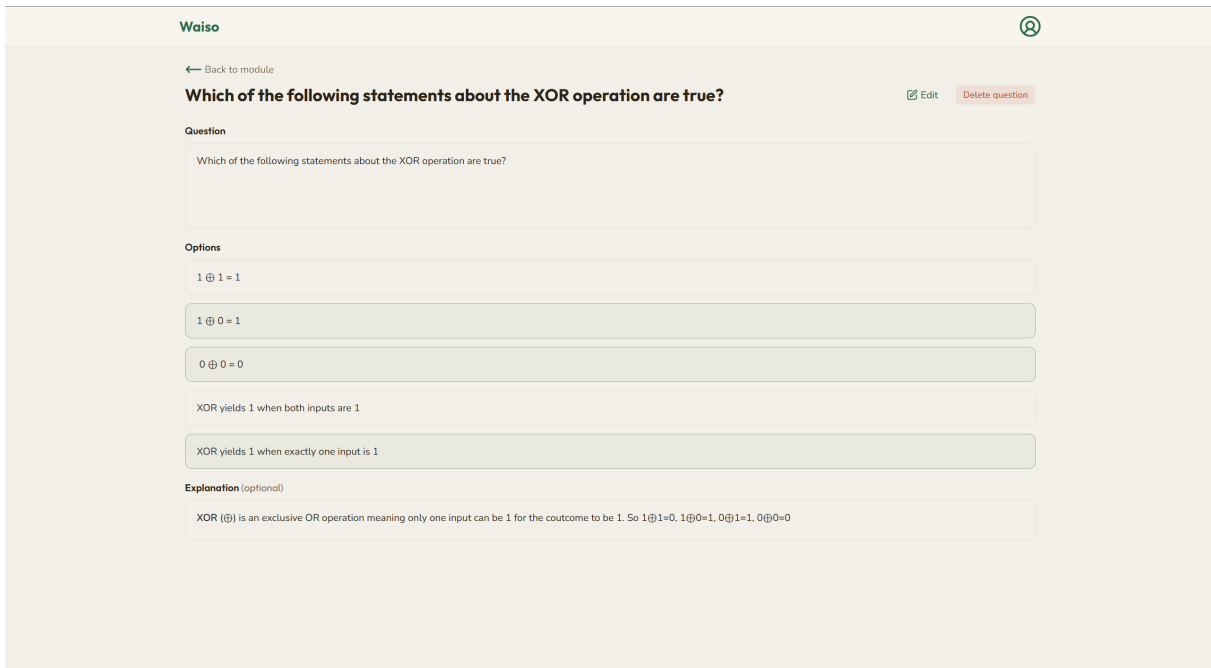
Figur 20: Översikt av de kurser en kursansvarig har skapat



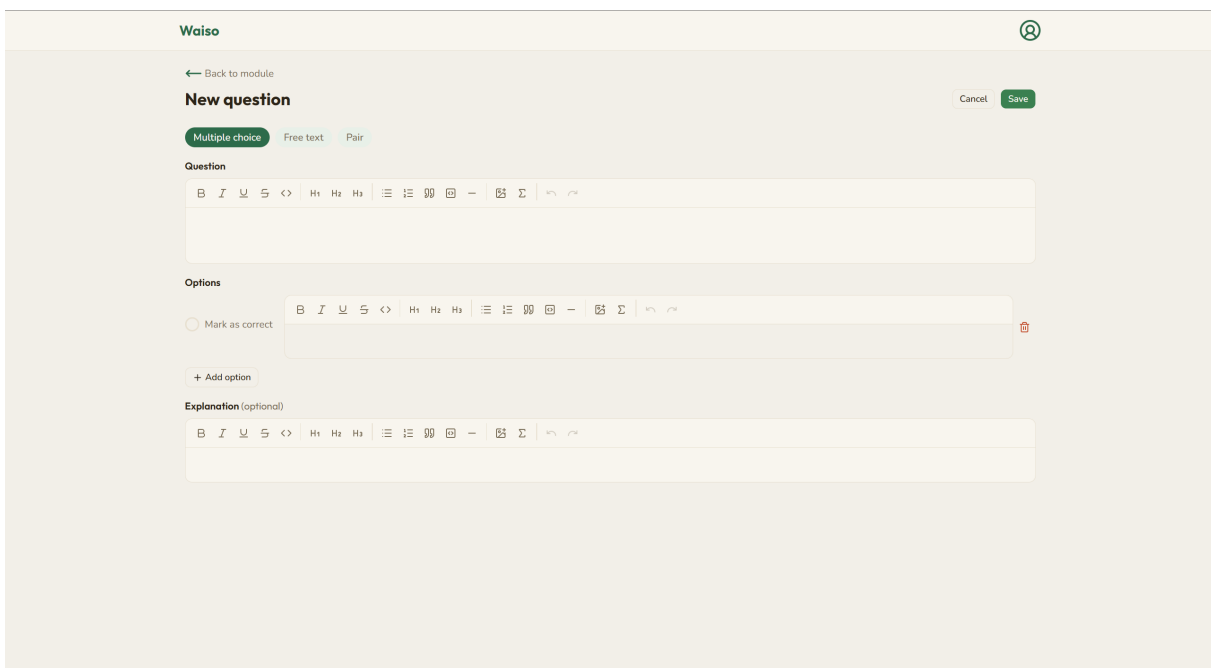
Figur 21: Kursöversikt för kursansvarig



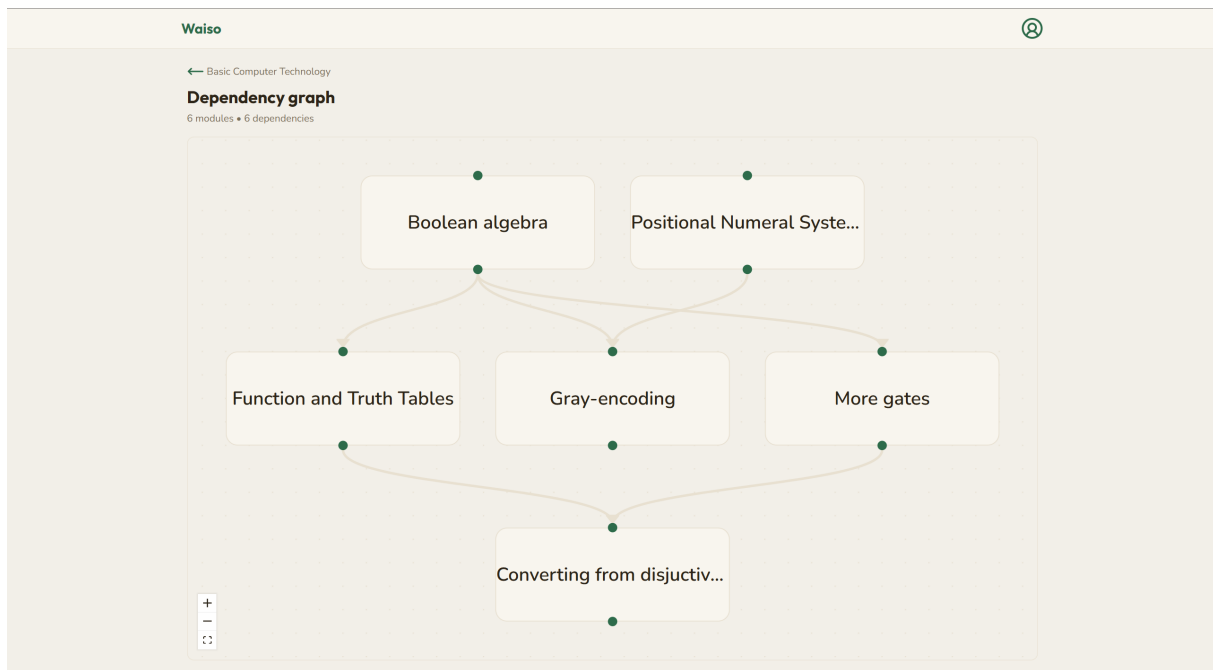
Figur 22: Modulöversikt för kursansvarig



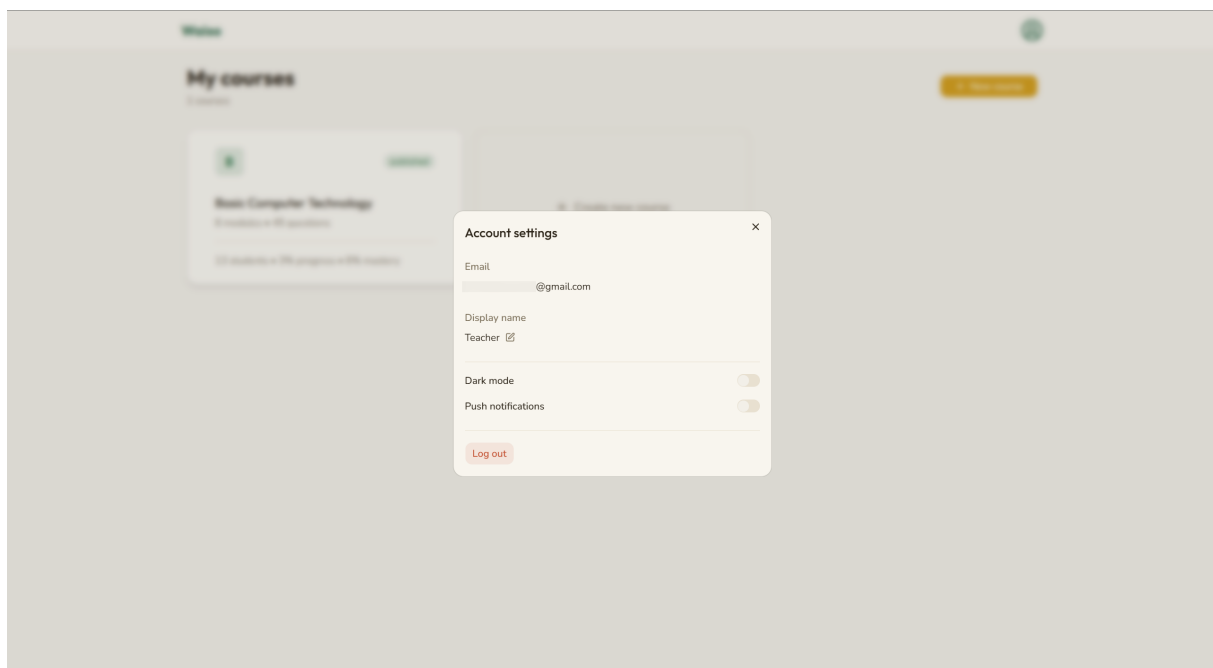
Figur 23: Exempel av en färdig flervals-fråga



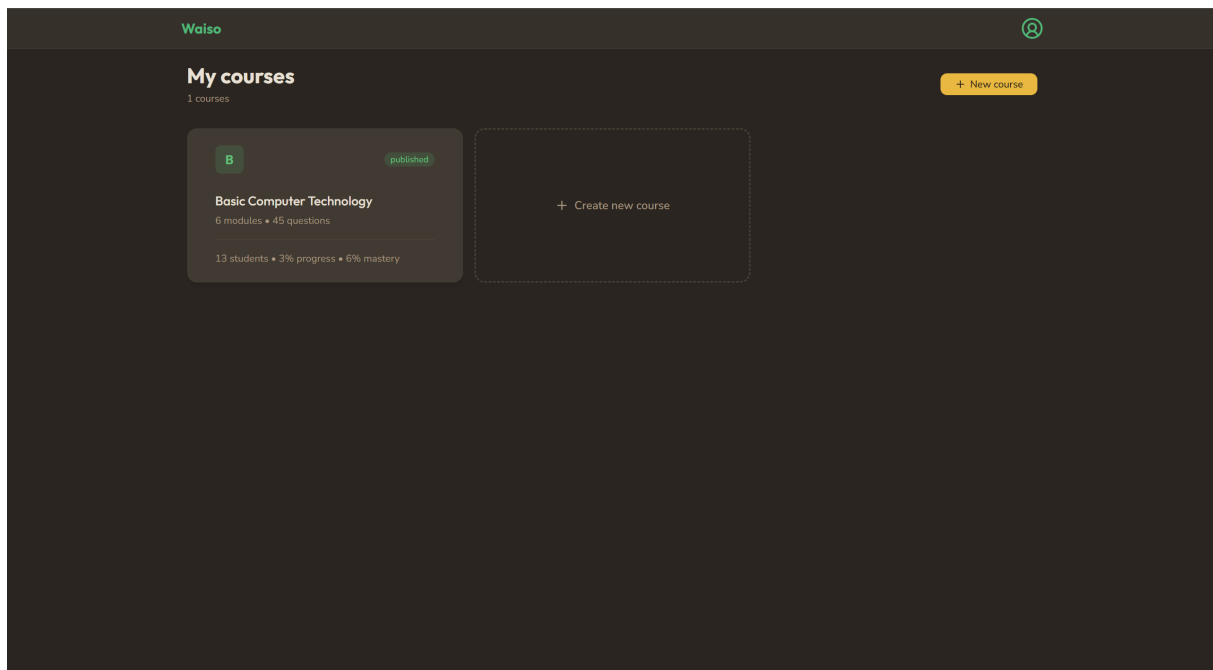
Figur 24: Exempel av en tom fråga



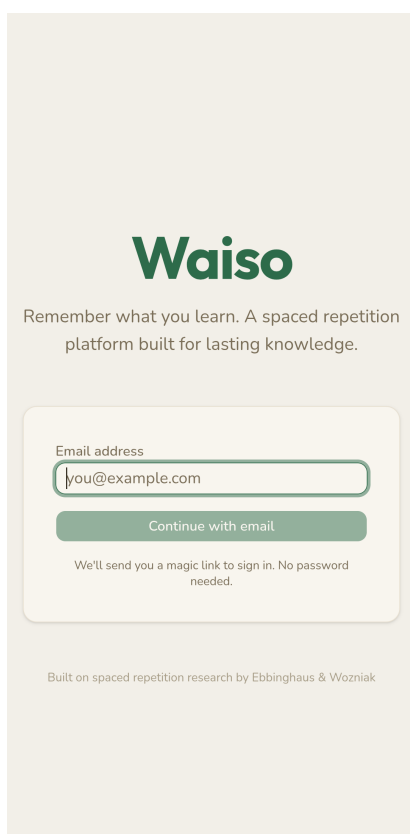
Figur 25: Graf över modul-relationer



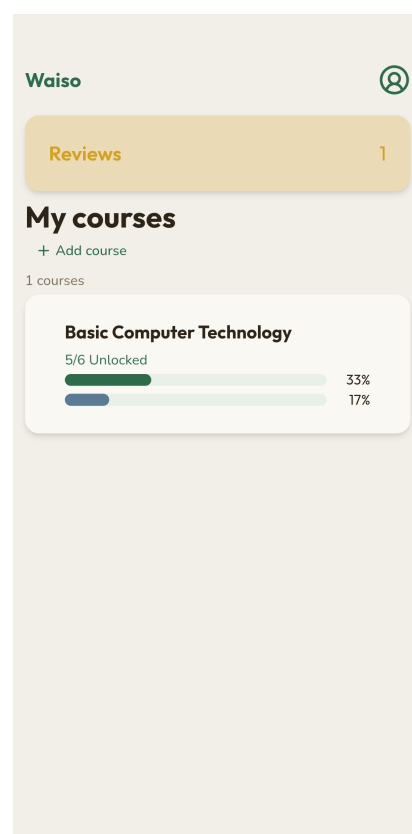
Figur 26: Användarinställningar för kursansvarig, redigerad med Gimp



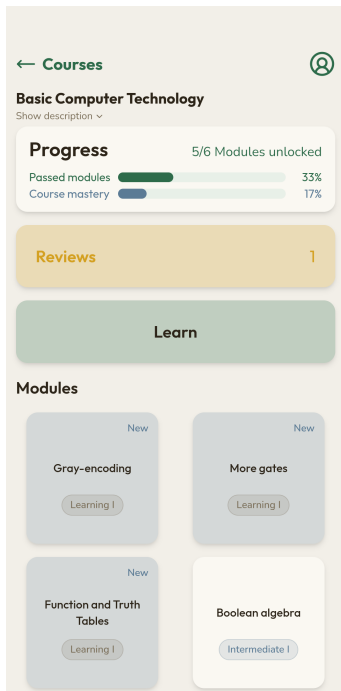
Figur 27: "Dark Mode" på kursöversiktssidan



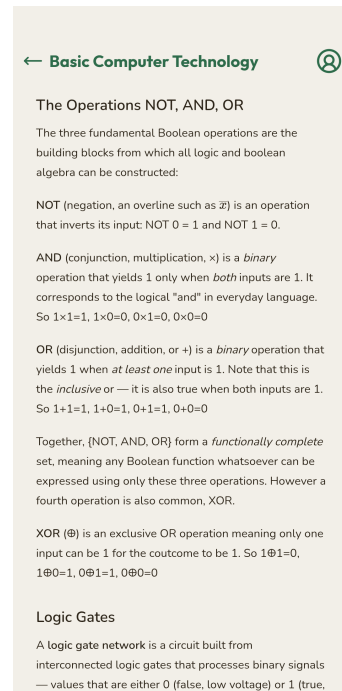
Figur 28: Inloggningskärm för mobiler



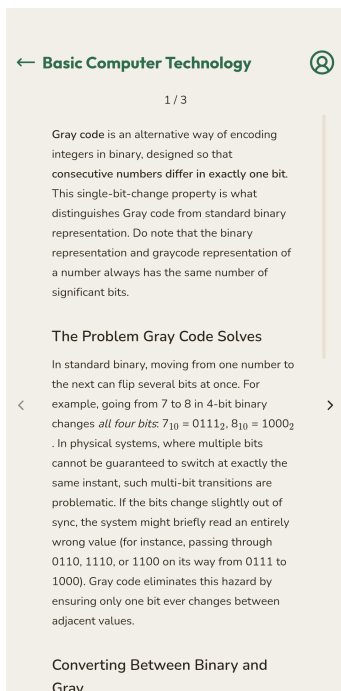
Figur 29: Översikt av kurser studenten är med i



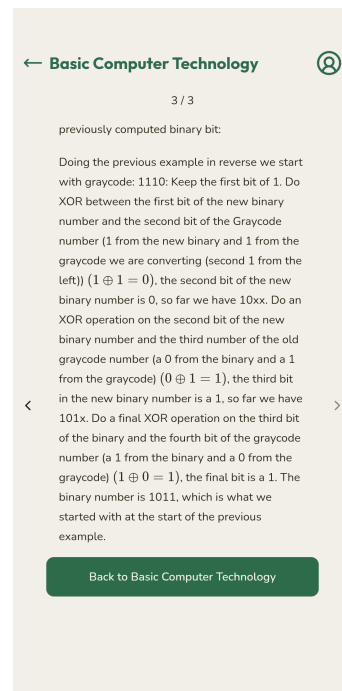
Figur 30: Studenters översikt av en kurs



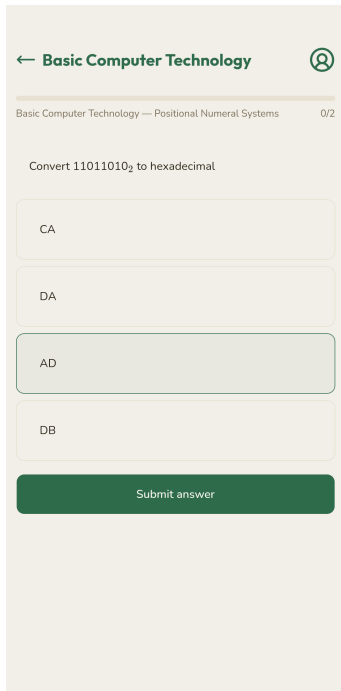
Figur 31: En öppnad modul



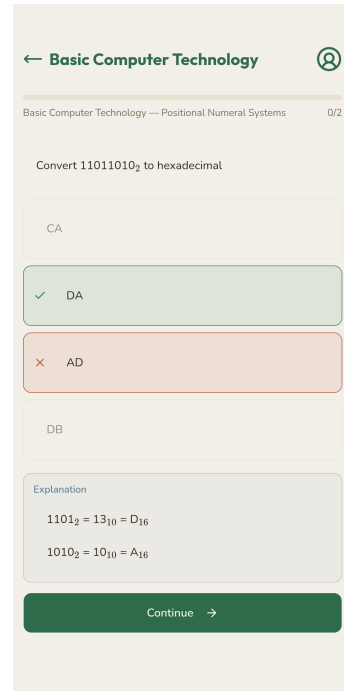
Figur 32: Inlärnings sidan för flera moduler



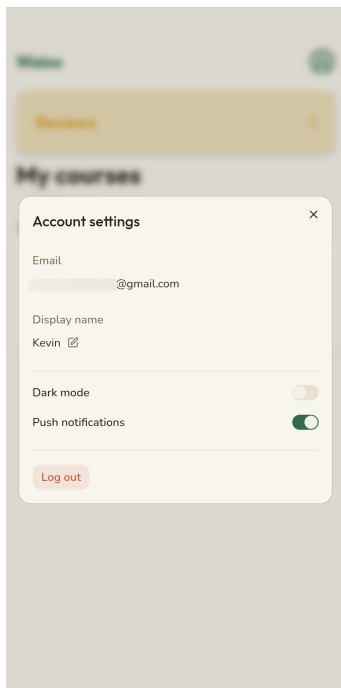
Figur 33: Sista modulen i en inlärnings-session



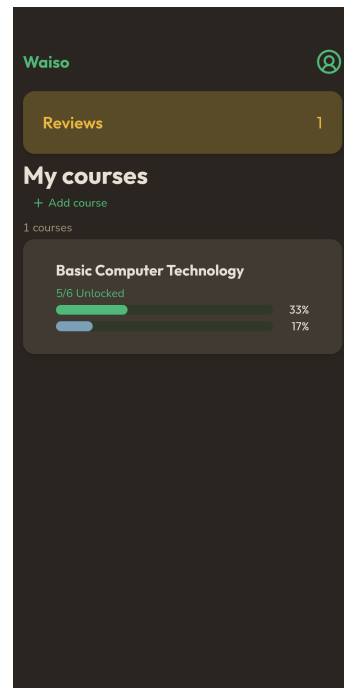
Figur 34: En fråga med svarsalternativ valt



Figur 35: En fel besvarad fråga



Figur 36: Studenters inställningsmeny, redigerad med Gimp



Figur 37: "Dark Mode" för en student på kursöversiktssidan