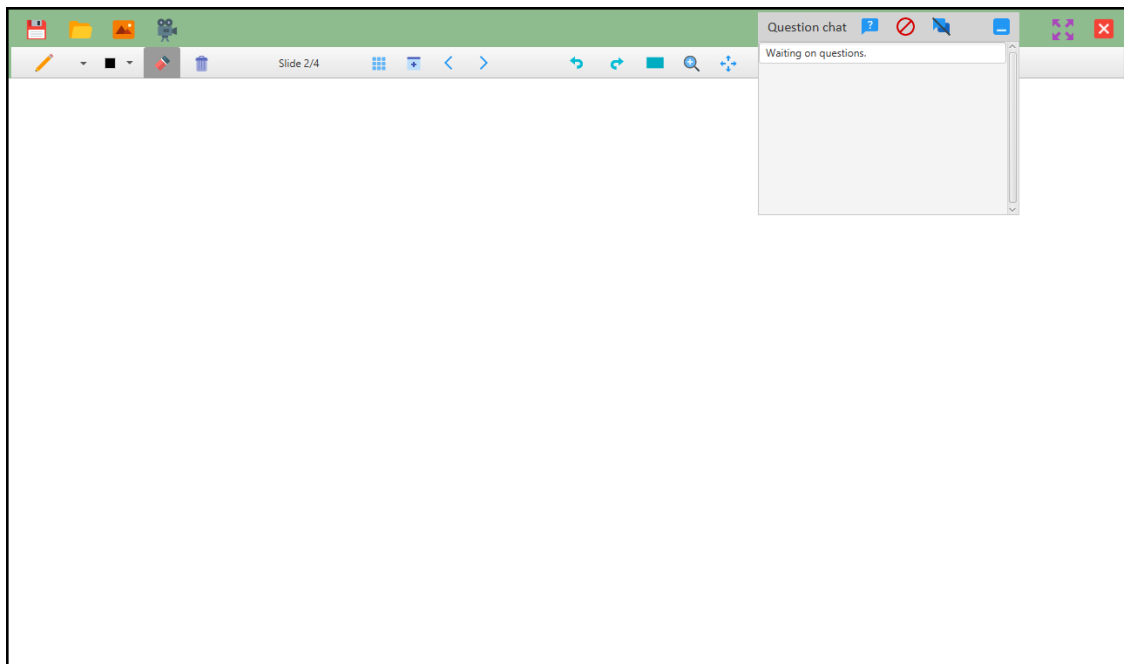




CHALMERS



Utveckling av en digital skrivtavla

Ett projekt i syfte att föra utvecklingen framåt inom ämnet utläring

Examensarbete inom Data- och Informationsteknik

AMANDA SJÖÖ

CHRISTER SONESSON

EXAMENSARBETE

Utveckling av en digital skrivtavla

Ett projekt i syfte att föra utvecklingen framåt inom ämnet utläring

AMANDA SJÖÖ
CHRISTER SONESSON



CHALMERS

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2019

Utveckling av en digital skrivtavla

Ett projekt i syfte att föra utvecklingen framåt inom ämnet utläring

AMANDA SJÖÖ

CHRISTER SONESSON

© AMANDA SJÖÖ, CHRISTER SONESSON, 2019.

Examinator: Peter Lundin, Data- och informationsteknik

Institutionen för Data- och Informationsteknik

Chalmers Tekniska Högskola / Göteborgs Universitet

SE-412 96 Göteborg

Telefon: 031 772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Omslag: Omslaget visar programmets huvudvy, vilket är vad föreläsaren kommer se när han eller hon arbetar med programmet

Institutionen för Data- och Informationsteknik

Gothenburg, Sweden 2019

Sammanfattning

Under senare år har allt mer nytänkande digitala system kommit ut på marknaden för undervisning med syftet att förenkla för landets skolor. Undersökningar visar att dessa kan leda till ökad effektivitet i utbildningen. Syftet med detta projekt är att designa ett sådant system som ska kunna effektivisera undervisningen ytterligare.

I det första steget utvärderades existerande system och baserat på utvärderingen togs ett nytt koncept fram. Den slutgiltiga lösningen kom att utgå från en ritplatta som kan placeras på ett podium kopplad till en projektor eller skärm. Med denna lösning kan föreläsaren bland annat vara vänd mot eleverna under undervisningen, något som syftar till att öka graden av publikkontakt. I nästa steg utvecklades mjukvaran till det program som ska användas på ritplattan. Huvudfokus för mjukvaran var att utveckla ett så enkelt och användarvänligt användargränssnitt som möjligt. Till sist anordnades användartester med föreläsare och andra anställda från Chalmers. Testarnas intryck av konceptet var positivt men samtidigt upptäcktes förbättringsområden hos ritprogrammet och dess funktioner.

Nyckelord: Digitalt utbildningsverktyg, pedagogik, interaktiv skrivtavla, ritplatta, ritprogram.

Abstract

In recent years, more innovative digital systems have arrived at the market for education tools in order to make things easier for schools around the country. The purpose with this thesis is to design such a system that should be able to make teaching more efficient.

In the first step of the project, existing systems were evaluated and based on the results, a new concept was formed. The final solution came to be concentrated around a drawing tablet that can be placed on a podium, connected to either a projector or screen. With this solution the teacher can face the students at all times which has many educational advantages. In the next step, the drawing program for the drawing tablet was developed. The main focus of the software was to develop as simple and user-friendly of a user-interface as possible. Finally, to evaluate the concept, user tests were performed with lecturers and other employees at Chalmers as the test subjects. The testers overall impression of the concept was positive, but at the same time areas of improvement were discovered with the main program.

Keywords: Digital education tool, pedagogy, interactive desk, drawing tablet, drawing program.

Förord

Detta examensarbete omfattar 15hp och skrevs på Institutionen för datateknik på Chalmers tekniska högskola. Det utfördes av Amanda Sjöo och Christer Sonesson som går dataingenjörsprogrammet, 180hp, på Chalmers tekniska högskola.

Ett stort tack till Sakib SisteK på Chalmers för ett spännande projekt och för bra handledning.

Vi vill även tacka de föreläsare som tog sig tid att utföra användartest med oss.

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	2
1.3	Mål	2
1.4	Avgränsningar	2
2	Teori	3
2.1	Utlärningens hörnstenar	3
2.1.1	Störningsmoment	4
2.1.2	Väck intresse och skapa motivativation	4
2.1.3	Pedagogik i projektet	5
2.2	Dagens utbildningsverktyg	5
2.2.1	Skrivtavlan	5
2.2.2	Overhead-apparaten	6
2.2.3	Projektor i kombination med laptop	6
2.2.4	Interaktiv skrivtavla	7
3	Metod	9
3.1	Tidigare arbete	9
3.2	Produktkrav	11
3.3	Nya lösningen	11
3.4	Designprocess	13
3.4.1	Val av testare	13
3.4.2	Testprocessen	14
4	Systemkonstruktion	15
4.1	Hårdvara	15
4.1.1	Val av ritplatta	15
4.2	Mjukvara	16
4.2.1	Ritfunktioner	16
4.2.2	Fönsterfunktioner	16
4.2.3	Slidefunktioner	17
4.2.4	Innehållsfunktioner	17
4.2.5	Chattfunktioner	18
4.3	Kodöversikt	18

4.3.1	PaintApp	19
4.3.2	PaintController	19
4.3.3	PaintModel	25
4.3.4	ImageObject	25
4.3.5	PictureHandler	25
4.3.6	PresenterViewController	25
4.3.7	Server	26
4.3.8	ClientHandler	27
4.3.9	ChatController	28
4.3.10	Client	28
4.3.11	Recorder	28
4.3.12	ImageToMovie	29
4.3.13	SoundToVideo	30
4.3.14	Sphinx4SpeechToText	30
4.3.15	UndoAndRedo	30
4.3.16	GraphicsObject	31
4.3.17	FXML-filerna	31
4.3.18	Android-klienten	31
5	Resultat	33
5.1	Testning	33
5.1.1	Pilottest	33
5.1.2	Användartester	34
5.1.3	Resultat i förhållande till målsättning	35
6	Miljö och Etik	37
6.1	Miljö	37
6.2	Etik	37
7	Slutsatser och Diskussion	39
7.1	Kritisk diskussion	40
8	Vidareutveckling av projektet	43
A	Checklista till testning	iii
B	Intervjufrågor	v
C	Funktionstabell	vii
D	Intervjer	xiii
D.1	Testare 1	xiii
D.2	Testare 2	xiii
D.3	Testare 3	xiv
D.4	Testare 4	xv
D.5	Testare 5	xvi
D.6	Testare 6	xvi

Terminologi

Externt bibliotek (jar):

En samling av öppet användbara kodsntuttar som tillhandahåller diverse funktioner för en programmerare. Tanken är att man inte ska behöva återuppfinna hjulet utan istället kunna använda externa bibliotek för att snabbt lösa redan kända och lösta problem med öppen tillgänglig kod.

JavaFX:

En mjukvaruplattform som bygger på Java som används för att skapa och leverera skrivbordsapplikationer som kan köras på en mängd olika plattformar [1].

JCodec:

Ett externt bibliotek som innehåller diverse funktioner för att i Java kunna manipulera videofiler.

Klient-server:

Klient-server är en arkitektur av IT-system med ett tydligt gränssnitt för kommunikation mellan de två modulerna klient och server. Klienten initierar kontakt med servern och ber om någon information, vilken servern då skickar till klienten. [2]

Trådar:

Trådar är en benämning på parallell exekvering av flera delar av en process (program), varje enskild parallell exekvering kallas då för en tråd.

Xuggle:

Även detta ett externt bibliotek som innehåller diverse funktioner för att i Java kunna manipulera videofiler.

Canvas:

En term i Java som innebär en vit rektangulär yta som användaren kan rita på, och som programmeraren kan modifiera via kod (för att lägga in exempelvis vad som ska hända vid en högerklickning respektive vänsterklickning med muspekaren på canvasen). Den här ytan är (till viss del) förprogrammerad och finns tillgänglig för användning i Java.

1

Inledning

Kapitel ett fungerar som en introduktion till examensarbetet, där bakgrund till problemet, syfte, mål för projektet samt avgränsningar för projektet presenteras.

1.1 Bakgrund

Den teknologiska utvecklingen går ständigt framåt, och detsamma gäller även utvecklingen inom mer 'mjuka' ämnen, och av dessa är ämnet pedagogik av speciellt intresse för den här rapporten. Nya teknologiska lösningar dyker upp på marknaden, och nya pedagogiska teorier kräver utvärdering och jämförelse med gamla beprövade metoder, för att se om utbildningssystemet kan förbättras och effektiviseras.

Utbildningssystemet som är vanligast i Sverige är det traditionella systemet som har varit på plats länge, där en föreläsare står i centrum och har fullt ansvar för det som lärs ut till eleverna under själva föreläsningarna. Även de utlärningsverktyg som föreläsare använder har i många fall varit på plats länge (laptops undantagna), då blackboards och whiteboards ofta står i centrum för föreläsningar. Men det finns nya metoder som skulle kunna bli normen i framtiden om de visar sig effektiva, ett exempel på en utveckling inom pedagogiken är så kallade 'flipped classrooms', och ett exempel inom den mer teknologiska utvecklingen är 'SMART-boards'. 'Flipped classroom' är ett försök att använda modern teknologi mer i dagens undervisning, då hemläxor ersätts av webb-baserade genomgångar av utlärningsmaterialet som i vanliga fall hade gått igenom av föreläsaren under en föreläsning, och själva lektionstiden kan då utnyttjas till mer interaktiva diskussioner med och mellan eleverna om det som eleverna har gått igenom online, eller möjligen till en mer djupgående genomgång av materialet. Detta ställer krav på eleverna i och med att de måste arbeta igenom materialet mer aktivt på egen hand istället för att passivt lyssna på en föreläsare. Det här systemet är under utveckling och bearbetning, och även om det finns studier som pekar på att systemet har en låg effekt, så tyder det ändå på försök till att förbättra utbildningen från en teoretisk vinkel [s.vi, 3].

En av de senaste teknologiska utvecklingarna är så kallade 'SMART-boards' och övriga produkter från företaget NetSmart, de har utvecklat både en digital skrivtavla som kan ersätta en whiteboard eller blackboard, och då erbjuder möjligheten för en föreläsare att genom mjukvarubaserade program vara väldigt flexibel med vad som visas på den här tavlan. Dessa nya digitala lösningar öppnar upp för stora möjligheter och visar på en möjlig ökad effektivitet av utläringen inom diverse ut-

bildningar [s.1, 4]. Trots detta ser vi att dessa lösningar till synes innehåller något som hindrar dess allmänna spridning i klassrum runt om i landet [s.23, 5]. Kanske är de för dyra för massinköp, eller kanske är de för svåransvända. Det verkar finnas element hos dessa mer uppkopplade och digitala verktyg som gör att institutioner och föreläsare föredrar mer beprövade verktyg. Det ser ut att finnas en bra möjlighet till att försöka förbättra de digitala verktygen i ett försök att skapa en ännu bättre lösning som har en högre chans att spridas och användas. Kanske kan ett sådant här system i framtiden användas ihop med metoden 'Flipped classroom', och i symbios uppnå en starkt positiv effekt på elevers inlärningsförmåga.

1.2 Syfte

Syftet med detta projekt är att utveckla en prototyp till ett enkelt och användarvänligt IT-verktyg för läraren/föreläsaren som kan bidra till att förbättra och effektivisera undervisningen och därmed underlätta lärandet för eleverna. Fokus för systemet ska vara enkel användning och stöd för bättre undervisning.

1.3 Mål

Målet med projektet är att utveckla och evaluera en mjukvara till en ritplatta. Syftet ska uppnås genom funktioner i programmet, i samspel med ritplattan. Fokus ligger på att göra programmet lättanvänt och lättförståeligt.

I det första steget kommer existerande utbildningssystem utvärderades utifrån pedagogiska grunder. Med hjälp av utvärderingen samt resultat från ett tidigare liknande projekt kommer ett nytt koncept tas fram. Därefter är nästa delmål att påbörja utvecklingen av en prototyp. Till sist ska en evaluering av prototypen och konceptet utföras där prototypen kommer utvärderas och diskuteras.

1.4 Avgränsningar

Under projektet kommer det endast ske utveckling av mjukvara, den hårdvaran som behövs för prototypen kommer att införskaffas på annat håll och inte utvecklas i projektet. Den mjukvara som utvecklas kommer dessutom bara utvecklas till att fungera på datorer. Mjukvaruapplikationen ska kunna köras på både Windows och MacOS. Andra operativsystem kommer det inte tas hänsyn till.

2

Teori

Följande kapitel kommer inledas med att förklara de olika grundpelarna inom pedagogik. I delsektionerna kommer det skrivas mer om de pedagogiska strategier som finns för effektivt lärande. Till sist kommer det kort presenteras om de vanligaste utbildningsverktygen som används idag samt för och nackdelar med dessa från ett pedagogiskt perspektiv.

2.1 Utlärningens hörnstenar

Människor lär sig bäst på olika sätt, något som gör jobbet som föreläsare komplicerat. Trots detta är ändå grunden till att lära sig densamma för alla. Olle Bälter, grundare av forskningsgruppen för Teknikförstärkt lärande, skriver följande i sin artikel *Tips för presentationer* [6]:

”All presentation går ut på att få mottagarna att förstå budskapet genom att koncentrera sig på detta, fundera över det, och diskutera det under så lång tid som möjligt. Hur? Genom att avskärma åskådarna från störningsmoment”.

Psykologen och föreläsaren Dennis Sale skriver i sin bok *Creative Teaching* [s.22, 7] att **motivation** och **uppmärksamhet** är de två hörnstenarna i effektiv inläring hos elever, och att motiverade elever med högre sannolikhet kommer att ge den uppmärksamhet som krävs samt överkomma möjliga problem i undervisningen.

Med den här informationen som bakgrund så kan följande påstående göras: för att effektivt lära sig något så måste en elev vara så koncentrerad som möjligt på ämnet. Detta görs genom att hålla en elev intresserad/motiverad samtidigt som denna utsätts för en så låg grad av störningsmoment som möjlig. Intresse behövs för att börja koncentrera sig på ämnet och för att bibehålla koncentrationen. Störningsmomenten gör det lätt för eleven att tappa koncentrationen på ämnet och börja fokusera på annat. Motivationen är en faktor som främst kommer inifrån eleverna själva [s.25, 7] men det finns metoder som kan användas av föreläsare för att försöka väcka intresse och skapa motivation hos eleverna. Det intressanta för en pedagog är då att veta hur man kan öka motivationsnivån samt vad ett störningsmoment är och hur de kan undvikas.

2.1.1 Störningsmoment

Ett störningsmoment är i detta fall alla faktorer som tar uppmärksamhet från undervisningen. Oftast delas de upp i interna och externa störningsmoment. Till externa störningsmoment hör saker som störande ljud i omgivningen, en föreläsare med utstickande dialekt, ett för varmt eller för kallt rum o.s.v. Interna störningsmoment beror på en elevs egna tillstånd. Detta kan vara saker som en elevs trötthet eller andra tankar som tävlar om elevens uppmärksamhet. Oavsett hur bra en föreläsare är på att presentera så kan externa och interna störningsmoment påverka elevernas koncentration och därmed också deras inlärningsförmåga. Alla sorters störningsmoment bör minimeras i så hög grad det går för att se till att det är så lätt som möjligt för eleverna att bibehålla koncentration. I de fallen med de interna störningsmomenten kan det vara svårt för en föreläsare att påverka då problemet handlar mer om elevens sinnestillstånd.

I fallet med de externa störningsmomenten bör en föreläsare försöka göra föreläsningen så lätt att följa som möjligt och hålla en så effektiv presentationsteknik som möjligt så att eventuella störningsmoment blir lättare att ha överseende med. Med bra presentationsteknik kan det även vara möjligt för en föreläsare att återfå elevens koncentration efter att ha förlorat den. Det är även viktigt för en föreläsare att fundera kring vilka störningsmoment ett utbildningsverktyg kan bidra till så att föreläsaren kan undvika dessa. Bland annat kan fläkten hos en overhead-apparat i kombination med ett mörkt rum ha sövande effekt på eleverna. Ett annat exempel är de tillfällen då skrivtavlan behöver suddas, eller under diverse andra avbrott i undervisningen.

Det är också väldigt viktigt att tänka på att om en elev har svårt att följa med i det som presenteras, antingen på grund av ämnets komplexitet eller på grund av bristande presentationsteknik, så behövs det inte mycket i form av störningsmoment för att en elev ska helt tappa fokus och behålla väldigt lite av det som lärs ut.

2.1.2 Väck intresse och skapa motivation

För att väcka ett intresse och skapa motivation hos eleverna är det viktigt att föreläsaren själv verkar intresserad och entusiastisk i det han eller hon försöker lära ut. Detta avläses främst ur föreläsarens sätt att prata och kroppsspråk. För att verka mer entusiastisk kan en föreläsare exempelvis variera sina rörelser, ändra avstånd från åhörarna, ändra tonläge i rösten, ha en bra hållning, använda gester samt bibehålla ögonkontakt med åhörarna.

Dennis Sale har skapat konceptet SHAPE som syftar till öka just elevernas intresse för den föreläsning som ges, SHAPE står för Stories, Humour, Activities, Presentation style, Examples [s.114, 7]. Han menar att väl berättade historier ökar elevernas emotionella anknytning till det som lärs ut, och att framstående föreläsare ofta använder historier som på något sätt kopplas till det som lärs ut för att öka elevernas intressenivå. Nyckelkomponenterna i att berätta en historia väl är enligt Sale att man har en tydlig och livlig presentationsstil, bra timing och extra beto-

ning på nyckelmoment i historien som ska anknyta till det som lärs ut, relevans för ämnet, och att man involverar studenterna i det som händer genom att hålla bra ögonkontakt med sin publik och att ha ett involverande och öppet kroppsspråk. Vidare så menar han att en bra användning av humor, aktiviteter som knyter an till ämnet samt involverar studenterna, och målande exempel också bidrar kraftigt till att hålla elevernas intressenivå hög.

Genom att använda Sales koncept SHAPE är det alltså möjligt att väcka ett intresse även hos elever som kanske inte egentligen är intresserade av ämnet och på så vis skapa motivation. Denna typ av teknik är extra viktig att använda i grundskolan där eleverna själva inte väljer vad de ska lära sig men där de fortfarande behöver lära sig alla ämnen.

2.1.3 Pedagogik i projektet

För att skapa ett så bra stödsystem som möjligt bör man finna en lösning som ger så hög grad av publikkontakt som möjligt samt minimerar antalet störningsmoment som skapas av verktyget. Ifall verktyget uppmuntrar till hög grad av publikkontakt blir det lättare för en lärare att hålla en god presentationsteknik vilket i sin tur kan leda till mer motiverade och intresserade elever. Med minskade störningsmoment från verktyget kan förhoppningsvis elever lättare hålla koncentrationen uppe och ge det fokus som krävs för att eleverna ska lära sig innehållet.

Förutom detta bör även det digitala verktyget tillhandahålla lärare med funktioner som kan hjälpa i syfte att förklara innehåll på ett mer tydligt sätt. Exempel på detta kan vara att kunna importera in en bild i programmet eller kunna stryka under något med en annan färg.

2.2 Dagens utbildningsverktyg

I denna sektion kommer det beskrivas lite om de olika aktuella eller allmänt intressanta utbildningsverktyg som finns. De verktyg som kommer beskrivas är den klassiska krittavlan eller dess efterföljare whiteboard, projektor i kombination med laptop, overhead-apparaten och till sist de nyare interaktiva skrivtavlor.

2.2.1 Skrivtavlan

Skrivtavlan i sina olika former (svarta tavlan eller whiteboard) är simpel, effektiv och billig. Konceptet är gammalt och man kan se spår av likande verktyg som användes redan på 1100-talet i Indien[8]. Efter det har den genomgått många förändringar där man bland annat ändrat färgen på svarta tavlan från svart till grön, och tagit fram en mer dammfri kalkpenna[8].

Fördelar: De främsta fördelarna hos dessa tavlor är att de är billiga och användarvänliga. En till fördel med skrivtavlor är att tempot naturligt blir långsammare än exempelvis vid användning av powerpoint. Detta kan göra det lättare för elever att

hinna fundera kring materialet och förstå bättre.

Nackdelar: Då en skrivtavla ofta är monterad på en vägg som är vänd mot publiken innebär detta att en föreläsare som skriver på tavlan delvis eller helt kommer vara vänd från publiken. Detta kan bland annat göra det svårare för elever att höra vad en föreläsare säger och det kan bli svårt för föreläsare att hålla kontakt med sina elever. Förutom detta så bidrar lösningen till flera opraktiska situationer som kan bli störningsmoment. Bland annat är det omöjligt att flytta om anteckningar på en skrivtavla utan att innehållet förstörs, det blir en paus i föreläsningen när föreläsaren behöver suddas tavlan samt att föreläsaren kan riskera att blockera tavlan medans han eller hon skriver.

2.2.2 Overhead-apparaten

Overhead-apparaten tillåter en föreläsare att snabbt visa förberedda bilder på en yta (ofta en vit projektorduk eller en white-board), bilderna är tryckta eller ritade på en tunn plastskiva i A4-storlek. En bra jämförelse är att kalla overhead-apparaten för en analog version av power-point presentationer.

Fördelar: föreläsaren kan vara vänd mot publiken samt gestikulera fritt vilket tillåter en hög grad av kontakt med publiken. Det är enkelt att förbereda inför ett undervisningstillfälle då plastskivorna kan göras i ordning innan.

Nackdelar: Vid användning av overhead rekommenderas att rummet har en låg ljusnivå. Det nedsläckta rummet i kombination med fläktljudet från overhead-apparaten bidrar med en sövande effekt. Detta kan göra det svårt att hålla långa föredrag på overhead. Det är en gammal teknik och den är inte särskilt flexibel då man är bunden till att använda genomskinliga plastpapper för att visa någonting.

2.2.3 Projektor i kombination med laptop

En modern (i förhållande till de tidigare nämnda systemen) och vanlig lösning är att ha en presentation på datorn gjord i exempelvis powerpoint, och sedan visa denna på en projektorduk med hjälp av en projektor ansluten till laptopen via HDMI-kabel. På detta sätt kan även videoklipp spelas upp, eller vad som helst annat som föreläsaren kan visa via sin laptop.

Fördelar: Detta är ett mycket interaktivt och brett system som kan visa nästintill vad som helst. Det har alla fördelar som de tidigare beskrivna systemen har, förutom skrivtavlans extremt höga användarvänlighet och låga kostnad.

Nackdelar: Den största nackdelen är att det inte finns något bra sätt att använda systemet som skrivverktyg, vilket är en viktig egenskap för föreläsare. Det finns tillgång till mjukvara som gör det möjligt att rita i en presentation men det är svårt att skriva text med en datormus. Föreläsaren kan även projicera powerpoint-slides på en whiteboard och sedan anteckna på whiteboarden, men då erhålls även de nackdelar

som finns hos en whiteboard, exempelvis när det gäller suddning, samt att föreläsaren skymmer det som visas av projektorn under tiden han eller hon antecknar eftersom det som ska visas kommer att projiceras på föreläsarens kropp. Förutom detta finns inga tydliga nackdelar utom att systemet i sig självt inte erbjuder några funktioner, alla funktioner som erbjuds görs det via diverse mjukvaruprogram som tillhandahåller en viss funktionalitet. En möjlig nackdel är att föreläsare som har en låg kunskap inom användandet av datorer kan ha det svårt att förbereda och visa presentationer, samt att ansluta laptopen till projektorn.

2.2.4 Interaktiv skrivtavla

Det mest moderna alternativet av skrivverktyg som ökar i användande på marknaden är de interaktiva skrivtavlor, där företaget SMART är det mest etablerade. SMART erbjuder främst två sorters skrivtavlor, den fristående tavlan SMART-BOARD och lösningen med en mobil ritplatta, SMART-PODIUM. SMART-BOARD är en produkt som fungerar som en digital skrivtavla med en rad olika funktioner. SMART-PODIUM är en ritplatta som man ska ha på ett podium, ritplattan kopplas sedan till valfri skärm. Föreläsaren står då vid podiumet och använder ritplattan som input till skärmen.

Fördelar: Många studier på de interaktiva skrivtavlor har gjorts i Storbritannien där de används på bred front. Dessa undersökningar visar många fördelar med att använda interaktiva skrivtavlor. Man har sett att tavlor ökar elevernas närvaro samt höjer deras kunskapsnivå [4]. Ett annat tydligt resultat visar på högre effektivitet under undervisningen. Med hjälp av det digitala förarbetet hinner föreläsaren undervisa mer under en lektion, samtidigt som man kan fånga elevernas intresse bättre än under traditionell undervisning[9].

SMART-PODIUM har förutom ovanstående också fördelen att föreläsaren kan vara vänd mot åhörarna under hela undervisningen. föreläsaren kommer inte heller vara i vägen för innehållet och blockera texten för eleverna som vid användning av en vanlig skrivtavla eller en SMART-BOARD.

Nackdelar: De största nackdelarna med dessa typer av produkter är att de är väsentligt mycket dyrare än de andra lösningarna som presenterats ovan. Priset är mellan 14500-63000 sek för SMART-BOARD beroende på modell. Priset för SMART-PODIUM är ekvivalent med en motsvarande ritplatta, men på detta så måste man prenumerera på deras mjukvarupaket för att få tillgång till det, vilket medför en månatlig kostnad så länge man vill använda systemet. De är inte heller lika användarvänliga eftersom användaren måste lära sig det digitala gränssnittet och alla funktioner för att kunna använda hjälpmedlet på ett bra sätt. Det har visats att även föreläsare med utbildning i liknande verktyg har svårt att komma igång och använda produkterna då de är komplicerade att sätta sig in i. SMART-BOARD har även samma nackdel som den vanliga skrivtavlan med att föreläsaren är vänd mot tavlan (en nackdel som ej delas av systemet SMART-PODIUM).

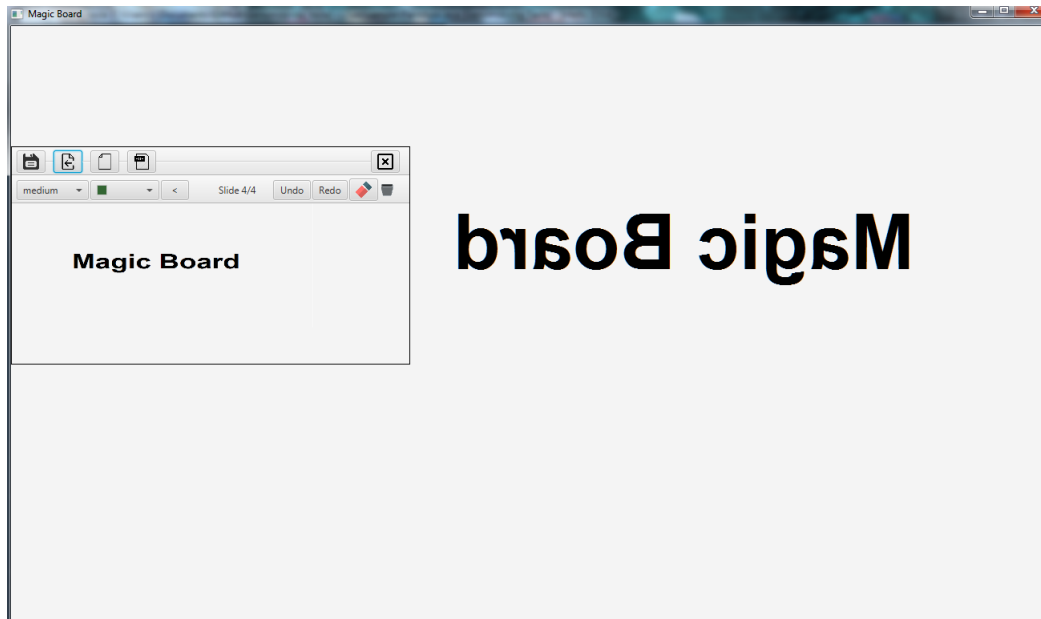
3

Metod

I detta kapitel beskrivs processen och tankegången i hur en prototyp utvecklades. I första delen av kapitlet beskrivs resultatet och slutsatsen från ett tidigare projekt med samma syfte. Därefter förklaras de produktkrav som tog fram för konceptet samt vad det slutgiltiga konceptet kom att bli. Till sist förklaras den metod som använts för att genomföra användartester i projektet.

3.1 Tidigare arbete

Detta projekt är en vidareutveckling av ett mindre tidigare projekt som utfördes under läsperiod två i höstterminen 2018 på Chalmers treåriga datatekniksprogram. Det äldre, redan genomförda, projektet kommer nu beskrivas. Det tidigare projektet grundade sig i idén att en föreläsare som kan hålla en maximal grad av ögonkontakt med publiken troligtvis också får en bättre publikkontakt, och därmed också uppnår en högre kvalitet på sina föreläsningar. Systemet som planerades för att uppnå detta kretsade kring att föreläsaren skulle ha tillgång till en transparent pekskärm, som bör stå längst fram i klassrummet. Föreläsaren ska helst kunna stå bakom denna tavla och skriva på den med hjälp av skärmens touch-funktion, men för det behövs specialiserad mjukvara som bland annat spegelvänder texten på ett naturligt sätt så att både föreläsare och elever lätt kan följa med i vad som skrivs och ritas. Att skärmen är transparent kan troligen leda till en högre grad av publikkontakt eftersom föreläsaren konstant kan vara vänd mot eleverna, oavsett om tavlan nyttjas för tillfället eller ej. Det tidigare projektet slutfördes med ett försök till en mjukvara som skulle köras på en sådan transparent tavla och uppnå det som beskrivits. Det programmet nådde dock aldrig längre än en prototyp-fas anpassad mot datorer mer än genomskinliga tavlor, detta eftersom det är svårt att få tillgång till den dyra teknologi som krävs (transparenta och digitala skrivtavlor).



Figur 3.1: Den tidigare projektgruppens skapade mjukvara för en transparent pekskärm

Den tidigare mjukvaran utvecklades i programmeringsspråket Java då många i den projektgruppen redan hade erfarenhet av det språket, gruppen valde att använda sig av verktyget JavaFX för att rita upp programmets grafik. Prototypen som skapades var ett program som kan liknas vid en simplare version av microsoft paint, men som även spegelvände det som ritades i programmet.

Under projektets gång försökte gruppen att köpa in en transparent pekskärm som skulle användas till den prototyp som höll på att utvecklas, men denna typ av hårdvara är relativt dyr. Av den anledningen stannade utvecklingen av projektet i prototyp-stadiet, och programmet som egentligen skulle ha körts på en dyr transparent pekskärm är istället fortfarande designat för att köras enbart på en dator.

I slutskedet av projektet valdes att skapa en enkät som förklarade konceptet samt ställde frågor om värdet av idén. Enkäten skickades ut till en mängd studenter på Chalmers och gruppen fick tillbaka svar från 119 elever. Slutsatsen av projektet kom till stora delar från just dessa svar. Bland annat fanns slutsatsen att föreläsaren blir svår att höra när personen är placerad bakom en skärm, ett problem som togs upp av många studenter som svarat på enkäten. Detta problem och många mer togs upp av enkäten och visade på att just lösningen med en transparent pekskärm kanske inte är den optimala. Det var också svårt att fortsätta utvecklingen med en transparent skärm då mjukvaran inte kunde testas på rätt hårdvara, och det skulle troligtvis bli svårt att få tag på rätt hårdvara även i ett senare skede. I diskussionsdelen av rapporten presenterades förslaget att vidareutveckling av projektet kunde ske med en ritplatta som fokus istället för en transparent pekskärm. Denna skulle ha samma fördelar med att en föreläsare kan ha ögonkontakt med åhörare samtidigt som man undviker nackdelarna med en pekskärm.

3.2 Produktkrav

I skapandet av konceptet utarbetades ett antal produktkrav. En viktig del av en föreläsares presentationsteknik kommer från att föreläsaren kan interagera med eleverna. Därför sätts produktkravet att föreläsaren och eleverna måste kunna se varandra hela tiden, mängden tid en föreläsare är vänd bort från publiken ska minimeras.

Som forskning visar kan en digital lösning leda till effektivisering av utläringen inom utbildningar [s.1, 4]. Av denna anledning ser vi stora möjligheter i att klara projektets mål ifall vi använder oss av en digital lösning. Därav har kravet att lösningen måste vara digital satts, det är mjukvara som ska stå för lösningen.

En nackdel med de interaktiva skrivtavlor som finns i dag är att de är väldigt innehållstäta och onödigt komplicerade att förstå, speciellt för en användare med låg grad av teknisk kunskap. För att se till att föreläsare använder produktens funktioner fullt ut ska systemet fokusera på ett fåtal funktioner som gör stor nytta hellre än många funktioner som tar över användargränssnittet och gör programmet svårt att använda. På grund av detta sätts produktkravet att användargränssnittet måste vara enkelt och tydligt att förstå.

En annan negativ sak med de interaktiva skrivtavlor är att systemen är mycket dyrare än andra lösningar. Det är positivt om produkten kan nå ut till så många användare som möjligt eftersom syftet är att hjälpa till med utläringen av kunskap vilket gynnar samhället som helhet, därmed sätts kravet att systemet ska vara gratis/billigt och av typen open-source.

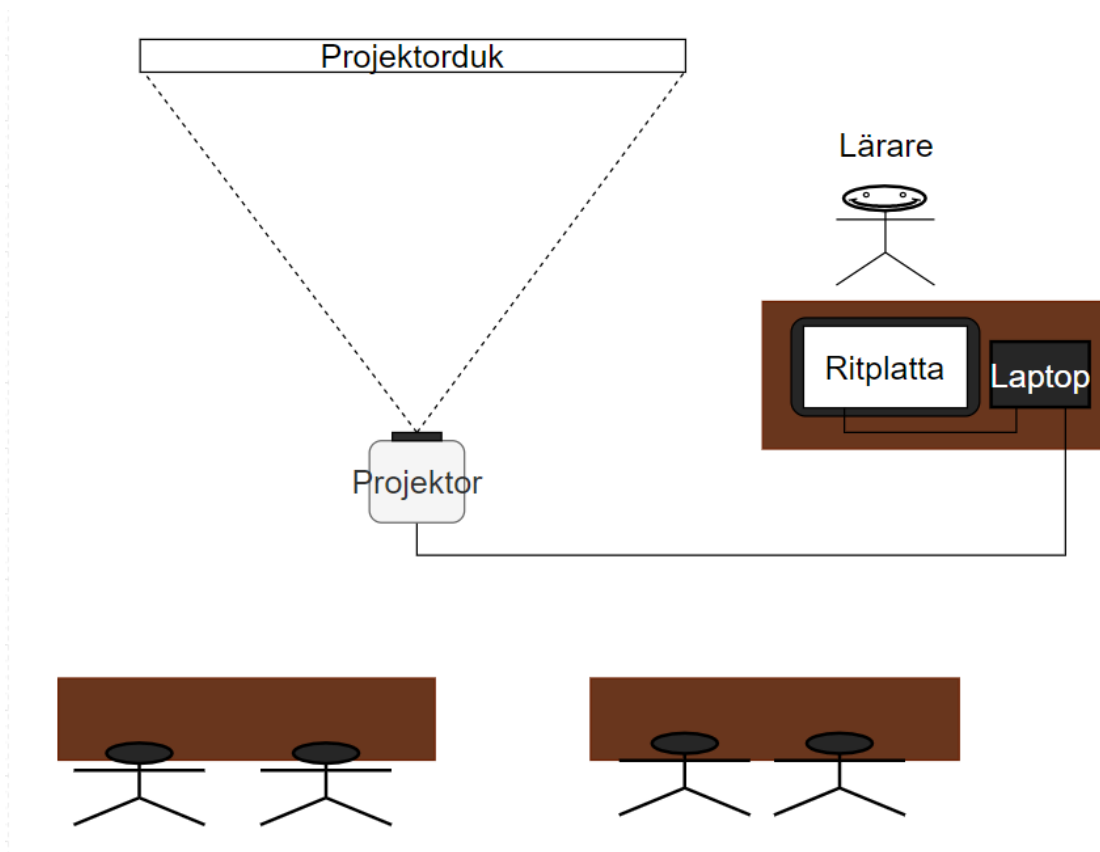
I slutändan fick vi fyra olika krav på produkten enligt nedan:

- Minimera tiden föreläsare är bortvänd från publiken.
- Lösningen ska vara digital
- Enkelt och användarvänligt användargränssnitt.
- Billigt system med open source mjukvara.

3.3 Nya lösningen

Lösningen kommer baseras kring en ritplatta som kopplas till föreläsarens egna laptop. Laptopen kommer i sin tur kopplas till en projektor eller en skärm. Ritplattan fungerar som inputsystem till laptopen och har ingen eget fungerande operativsystem, all input till ritplattan går direkt in som input till laptopen, ingen mjukvara behöver utvecklas för att fungera som ett gränssnitt mellan laptop och ritplatta. Det är viktigt att både ritplattan och projektorn utvidgar laptopens skärm och inte duplicerar. Tanken är att föreläsaren ska se en programvy som innehåller alla de funktioner som föreläsaren kan använda, medans det som projiceras för eleverna är en annan vy som enbart innehåller det som är viktigt för eleverna (det som ritas upp av föreläsaren, men ej alla menyknappar). Detta uppnås genom att föreläsaren använder laptopens existerande 'extend'-funktion som gör att en del av program-

met kan dras ut till en av de skärmar som används, och då är då enbart en av dessa skärmar som projiceras på väggen (där laptop och ritplatta alltså innebär två separata skärmar i systemet). När programmets startas kommer huvudfönstret för programmet öppnas på ritplattan (se bild 1 a) och ett andra fönster kommer öppnas på laptopen och projiceras (se figur 1 b). Ritplattans fönster innehåller en stor vit yta och ett menyfält i övre delen av fönstret. Användaren kan dra på den vita skrivytan med den tillhörande ritplattepennan och på så vis rita eller skriva. Projektorns fönster består endast av den stora vita ytan som kommer vara en avbild av den på ritplattan.

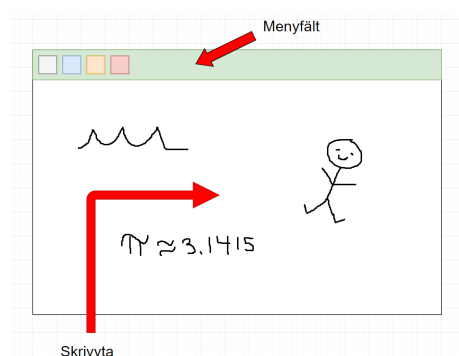


Figur 3.2: Konceptbild sedd ovanifrån.

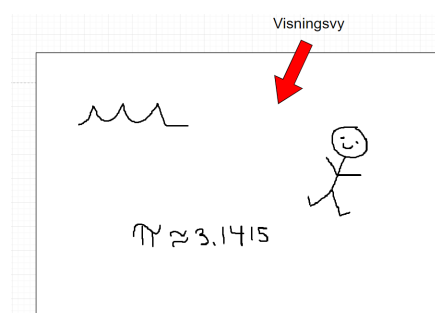
Tanken är att varje lärosal innehåller en egen ritplatta och en skärm eller projektor. Föreläsaren tar med sig sin egen laptop och kopplar in den till resterande system på plats i salen. För bästa pedagogiska förutsättningar ska en föreläsare placera sig vid ett bord eller kateder med kroppen vänd mot eleverna, på en plats där denna inte är i vägen för projektorduken.

Denna lösning har flera fördelar. Systemet är relativt billigt. Många lärosalar är redan utrustade med projektor eller skärm och många föreläsare har redan en egen laptop. Det enda som behöver köpas in är ritplattor. Eftersom systemet är relativt mobilt är det lätt för en föreläsare att placera sig på ett sätt så att han eller hon får bra kontakt med åhörarna samtidigt som att han eller hon inte står iväg för tavlan. Då alla föreläsare enligt det här systemet ska ha mjukvaran på sin laptop så är

det lätt för en föreläsare att förbereda en presentation och planera upplägget av en lektion i förväg. Föreläsaren kan exempelvis förbereda ett antal slides i powerpoint och sedan under lektionen importera hela powerpoint-presentationen till programmet, och därefter åtnjuta de extrafunktioner programmet erbjuder (i detta fallet främst högre grad av interaktivitet med powerpoint-presentationen och möjligheten att spela in föreläsningen). Om inte powerpoint-slides används så kan även text importeras in i programmet, bilder kan importeras, eller så kan föreläsaren teckna fritt. Föreläsaren kan fritt teckna över allting som visas på skärmen, zooma in, samt klippa ut eller kopiera delar av det som visas, oavsett vad det är som importerats till programmet. Detta kan vara användbart, till exempel så kan man enkelt stryka under en viktig del av ett importerat textstycke, eller ringa in det viktiga i en importerad bild.



Figur 3.3: Användargränssnitt till ritplattan.



Figur 3.4: Användargränssnitt till projektor eller skärm.

3.4 Designprocess

Med det mål som satts för detta projekt, att utveckla ett koncept som ska underlätta lärande, behövs ett sätt att evaluera konceptet för att kunna utvärdera huruvida målet med projektet är nått eller inte. För att göra detta valdes att utföra användartester av mjukvaran samt intervjuer med frågor angående mjukvara och koncept. Härnäst beskrivs mer detaljerat de val som gjordes under designprocessen.

Första steget är att hitta potentiella testare. För att testet skulle ge ett någorlunda rättvist resultat sattes en gräns på att minst fem tester behövde genomföras. För att vara säkra på att tillräckligt många genomför testet bestämdes att minst 20 föreläsare skulle kontaktas. Innan de riktiga testerna påbörjades kommer först en pilotstudie av testet att genomföras. Pilotstudien kommer gå till på samma sätt som det vanliga testet och är till för att hitta småfel och andra delar som kan modifieras inför den riktiga testomgången.

3.4.1 Val av testare

När testare valdes bestämdes att testgruppen skulle bestå av föreläsare från Chalmers, och då från olika sektioner inom Chalmers. Detta val gjordes för att få en

variation av ämnen och kunna fånga olika behov som finns inom de olika ämnesgrupperna. Pågrund av tidsbrist valdes inte att kontakta föreläsare från exempelvis gymnasiet eller lägre årskurser. Något som skulle kunna vara bra att göra för att fånga upp alla tänkta användares åsikter.

Det krav som ställs på testaren är att denna ska jobba med undervisning eller ha pedagogik som ämnesområde. Testarna kommer slumpmässigt plockas från Chalmers hemsida och därefter kontrolleras så de uppnår kraven.

3.4.2 Testprocessen

Testet delas upp i tre delar: först en snabb bakgrund till projektet och vad resultatet blivit. Därefter får testaren testa att använda mjukvaran på ritplattan samtidigt som denna ska försöka utföra vissa valda funktioner från en checklista (Appendix A). Efter det att testaren känner sig nöjd, eller då vi är nöjda, kommer testaren att få svara på några frågor angående användarupplevelsen och mer generella frågor om koncept samt mjukvarulösning (Appendix B).

4

Systemkonstruktion

I detta delkapitel kommer det beskrivas mer i detalj om den mjukvaran som utvecklas och om hårdvaran till konceptet. Då hårdvaran inte var en del som utvecklades kommer hårdvarudelen mest innehålla beskrivning och krav på den hårdvaran som behövs för konceptet. Mjukvarudelen kommer först gå in på de funktioner som skapats och vilken nytta dessa har till projektet. Därefter kommer det beskrivas mer djupgående om vilka klasser som skapats och hur dess fungerar.

4.1 Hårdvara

Hårdvaran som behövs för att alla funktioner ska fungera som tänkt är en laptop installerad med mjukvaruprogrammet, en ritplatta som fungerar som en skärm, någon hårdvara att visa upp det stora fönstret på (antingen projektor eller en större skärm) och en mikrofon till datorn (för ljudinspelning). Under utvecklingen har projektet använt en ritplatta av modellen Wacom Cintiq 16 som utveckling gjorts mot. Resterande hårdvara som använts är projektmedlemmarnas egna laptops. Den ritplatta som användes var en ritplatta vars huvudsyfte är konst, och den har därmed inget eget operativsystem. Den kopplas helt enkelt in i datorn, sedan installerar användaren drivrutiner från Wacoms hemsida, och efter detta fungerar den som en utökning av datorskärmen, men nu med möjlighet för en högre grad av olika inputs (finkänsligt tryck-input som kan användas ihop med den penna som följer med plattan, samt en större yta än vad en laptops skärm brukar erbjuda).

4.1.1 Val av ritplatta

Anledningen till att just modellen Wacom Cintiq 16 köptes in berodde på flera faktorer. För det första är företaget som tillverkar ritplattan ett av de ledande företagen inom branschen av ritplattor, med många kunder inom animation och digital konst. På så vis visste vi att deras plattor, oberoende av vilken modell, skulle ha den funktionalitet som räcker för det jobb vi vill att den ska göra. Företaget har både ritplattor med eller utan skärm, men för den prototyp vi tänkt oss behövde vi kolla på de modeller med egen skärm. Av dessa finns ett urval av plattor från 13 tum till 32 tum. Vissa modeller kommer också i pro-versioner där ritplattan har en ännu mer finkänslig ritfunktion. Sedan finns också alternativet för HD-skärm för vissa modeller.

Wacom Cintiq 16 valdes just till skillnad från de andra större modellerna eller de

bättre pro-modellerna då vi ansåg att denna var tillräckligt stor (så att användaren kan teckna och skriva på den på ett bekvämt vis), och innehöll de funktioner vi letade efter till ett billigare pris än pro-modellerna. Förutom det har mycket tanke hos ritplattan lagts på ergonomi där pennan skapats för att ligga bra i handen och skärmen har en bred ram till för att vila handen på när man ritat. Wacom Cintiq 16 har också ett friktionsfilter på skärmen samt en mjuk topp på pennan som tillsammans ger känslan av att rita på papper vilket vi tror kommer uppskattas av användare vid längre användning.

4.2 Mjukvara

Från det tidigare projektet utvecklades en bra kodgrund där de flesta grundfunktionerna redan var implementerade (rita, sudda, ändra pennstorlek, ta bort allt, o.s.v). Enda problemet var att denna lösning var speciellt anpassad för att fungera på en transparent pekskärm. Detta betydde att det föreläsaren ritade var tvunget att vändas för att eleverna skulle kunna läsa texten. Mjukvaran från det tidigare projektet löste detta genom att dela upp användargränssnittet i två olika fönster. Ett mindre som föreläsaren skriver i som har ett eget menyfält och ett fönster som tar upp resterande yta och kommer speglavända föreläsarens fönster. Första steget blev därför att omarbete programmet så att användargränssnittet endast har en rityta och ett menyfält.

Nästa steg var att börja utveckla funktioner som syftar till att erbjuda funktioner en föreläsare kan tänkas vilja ha. Dessa bestämdes utefter egna tankar om vad som skulle vara bra med återkoppling från vår handledare. Mjukvaran har implementerats med en mängd funktioner som kan delas upp i fem olika delgrupper av verktyg. Dessa är: ritfunktioner, fönsterfunktioner, chatt-funktioner, bildfunktioner och innehållsfunktioner. För att se mer beskrivning om varje enskild funktion se appendix C.

4.2.1 Ritfunktioner

Projektet går ut på att skapa en digital form av en skrivtavla. Ritfunktionen blir den viktigaste funktionen som måste finnas för att lösningen ens ska kunna efterlikna en skrivtavla. I användargränssnittet består all yta, förutom menyfältet längst, upp av en vit rityta. Ritfunktionen gör det möjligt för användaren att helt fritt kunna uttrycka sig i bild och text för att lättare kunna förmedla ett budskap. För att tillföra ännu mer till denna funktion har även funktionerna ”ändra storlek på pennan” och ”ändra färg på pennan” lagts till.

4.2.2 Fönsterfunktioner

Dessa funktioner hanterar programmets applikationsfönstret. Med de olika funktionerna kan man: stänga av applikationen, minimera/maximera fönstret, flytta på ett minimerat fönster samt ändra storlek på ett minimerat fönster. Dessa finns för att

en användare ska kunna få möjlighet att ha andra fönster uppe utan att de blockerar varandra.

4.2.3 Slidefunktioner

En nackdel hos traditionella skrivtavlor är att det finns begränsat med utrymme och det inte går att spara informationen på ett bra sätt. För detta projekt har det skapats ett bildsystem som gör det möjligt att spara tidigare bilder. En användare skapar under ett tillfälle en hel presentation där alla bilder sparats i ordning och kan återbesökas. Detta gör det lätt att återkoppla till tidigare material som användaren gått igenom. Funktionerna som lagts till för att hantera bilder är: skapa en ny bild, gå till föregående bild, se en översikt över alla bilder som skapats under tillfället och hoppa direkt till en av dem, ta bort en bild samt spara aktuella bilden eller alla bilder. Användargränssnittet är också utrustat med en notis om det totala antalet bilder och vilken bild i bildspelet som är öppen för tillfället. Bildsystemet i sin helhet gör det enkelt för en användare att referera tillbaka till tidigare material. Spara hela bildspels-funktionen gör det möjligt för en användare att spara hela föreläsningar och enkelt ladda upp dessa på en plats där elever kan få tag i dem. På så vis behöver inte en lyssnare anteckna allt material utan kan helt koncentrera sig på innehållet.

4.2.4 Innehållsfunktioner

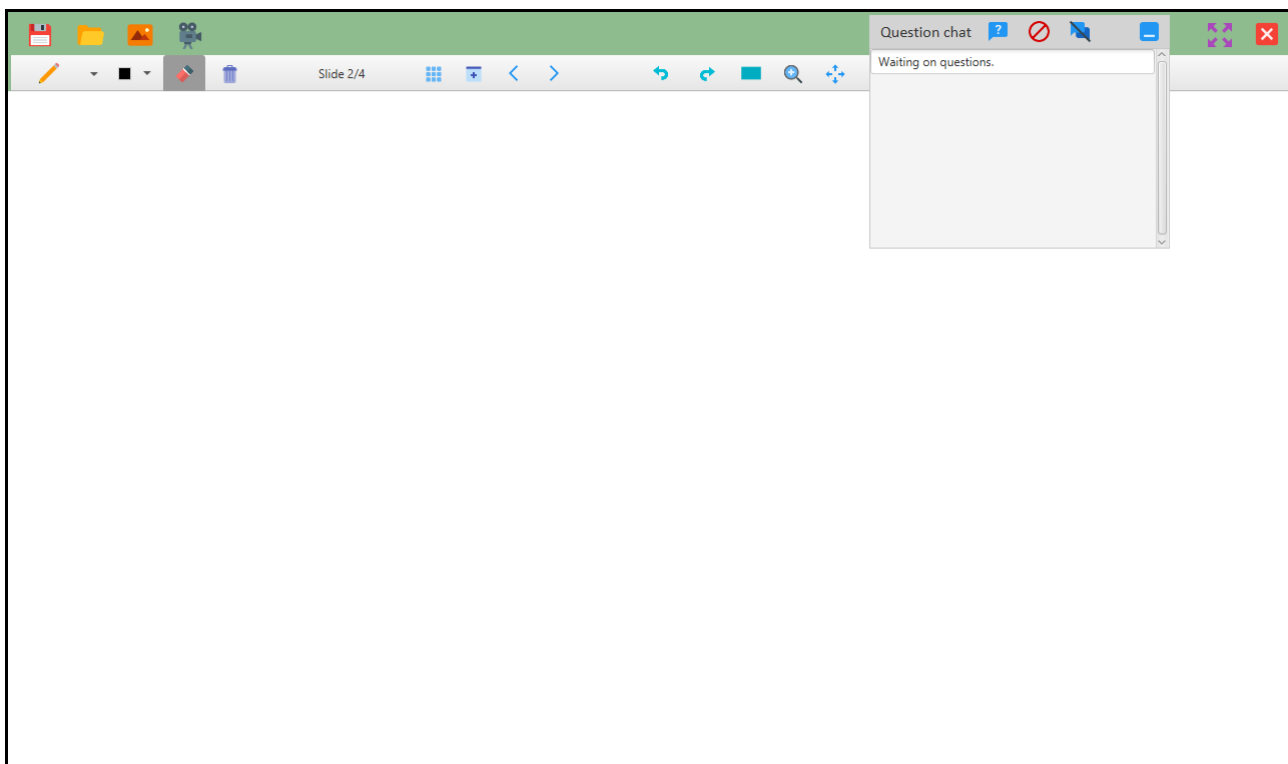
Denna grupp syftar till funktioner som modifierar innehållet eller lägger till innehåll på skrivytan. När det gäller modifiera innehållet finns funktionerna sudda, ångra och göra om och till slut select. Suddfunktionen gör det möjligt att sudda småfel. Med ångra tar programmet bort de senaste ändringarna. Med "göra om" går det att ta tillbaka ändringar som har ångrats. Med funktionen "select" markerar användaren ett område. Det markerade området kan kopieras, raderas, flyttas och ändras storlek på. Alla dessa funktioner är till för att användaren ska kunna hantera skrivytans innehåll på ett snabbt och smidigt sätt.

De funktioner som lägger till innehåll på skrivytan är importera bild-funktionen samt importera fil-funktionen. Den förstnämnda används för att importera en bild som kan placeras på skrivytan. Det är också möjligt att ändra storlek på bilden. Med importera fil-funktionen väljer användaren en fil och beroende på filsort kommer programmet öppna filen (om det är ett filformat som stöds av programmet). Det går bland annat att öppna text-filer, word-filer, pdf-filer och powerpoint-filer.

Ifall en användare vill zooma in på en specifik del av bilden finns också en funktion för just detta. Till sist finns också en funktion som används för att filma programmets innehåll. Med funktionen kan en hel föreläsning spelas in och sparas på användarens dator. Om användaren har en mikrofon inkopplad kan också ljudinspelning ske för filmen.

4.2.5 Chattfunktioner

Chattfunktionen är en funktion som gör det möjligt för elever att skicka in frågor till en föreläsare anonymt via en app (som även utvecklats under projektet) till android-telefoner. Användargränssnittet har ett eget chattfönster där dessa frågor dyker upp. Funktionen är till för att öka kommunikationen mellan föreläsaren och eleverna. Med chatten kommer flera olika funktioner: en banna funktion, som låter användaren bannlysa ett meddelande och radera detta. Det finns också en ”kopp-la bort chatt/återanslut chatt” funktion som stänger eller öppnar chatten och en ”göm/visa” funktion som döljer eller visar chattfönstret för användaren.



Figur 4.1: Användargränssnitt från mjukvaran.

4.3 Kodöversikt

Här kommer alla programmeringsdelar från huvudapplikationen att förklaras, och relevanta delar av kod kommer visas. Fokus kommer läggas på kod som är extra intressant, eller speciellt svår att förstå.

Koden till ritapplikationen är skriven i Java med tillägget JavaFX. De externa biblioteken Xuggle, JCodec, PNG-decoder (underlättar importen av bildfiler), Tika (underlättar importen av diverse textfiler) och Sphinx4SpeechToText (syftar till att låta en använda speech-to-text, men då den funktionen fungerade dåligt så används enbart bibliotekets ljudinspelningsfunktion för ljudinspelning) har även använts.

4.3.1 PaintApp

Den här klassen innehåller main-funktionen och är därmed den klass som körs för att starta programmet. I övrigt så gör den några initiala inställningar som syftar till att presentera allting på rätt sätt för användaren, men är en kort klass på bara 70 rader kod, och är till stor del återanvänd från det tidigare projektet. Den laddar upp FXML-filen som hanterar layouten för huvudvyn via följande kodsnuitt:

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("
    paint.fxml"));

Parent root = loader.load();
```

Med följande kod hämtar programmet skärmens höjd och bredd för att sätta den storlek på det minimerade fönstret. Därefter sätts storleken på fönstret till att anpassas automatiskt till att ta upp hela skärmen med metoden `setMaximize()`.

```
Rectangle2D screenBounds = Screen.getPrimary().
    getVisualBounds();
double width = (screenBounds.getWidth());
double height = (screenBounds.getHeight());
stage.setHeight(height - 200);
stage.setWidth(width - 400);"
final Scene scene = new Scene(root, width - 200, height - 200);

stage.setMaximized(true);
```

Följande kod sätter muspekaren till en svart cirkel, motsvarande den storlek på "pennan" man kan rita med, genom att använda de inbyggda JavaFX-metoderna `snapshot` och `setCursor` (`snapshot` ger tillbaka en bild av den sak som kallar på `snapshot`, och `setCursor` ändrar helt enkelt windows-muspekaren till en bild som man ger funktionen):

```
Circle circle = new Circle(2, Color.BLACK);
circle.setStroke(Color.BLACK);
SnapshotParameters sp = new SnapshotParameters();
sp.setFill(Color.TRANSPARENT);
Image image = circle.snapshot(sp, null);
scene.setCursor(new ImageCursor(image, 16, 16));
```

Sista delen av klassen kopplar ihop FXML-klassen med JavaFX-stage för att sedan göra den synlig med följande kod:

```
stage.setScene(scene);
stage.show();
```

4.3.2 PaintController

`PaintController` är kontrollklass till huvud FXML-filen `paint.fxml`. Detta innebär att denna klass innehåller alla metoder som anropas av de gui-element som FXML-

filen har. Detta är en ganska lång klass på nära 2000 rader, så enbart de viktigaste delarna kommer gås igenom här. Klassen är delvis återanvänd från det tidigare projektet, de delar som återanvänts är de delar som hanterar grundfunktionerna i att rita på en Canvas med muspekaren (dock är även dessa omarbetade till en viss grad).

Metoden *Initialize()* är en metod som kallas på när den tillhörande FXML-filen laddas i applikationstråden. Denna innehåller en mängd initieringar av diverse fält, men det viktigaste den gör är att sätta upp de metoder som FXML-objekten kopplas till i FXML-filen. Denna koppling mellan ett FXML-objekt och en metod i kontrollklassen triggas av en valfritt *event* som FXML-objekten har tillgång till. Event för en knapp är exempelvis *onMouseClicked*, som gör att den kopplade metoden kallas på när knappen blir klickad. Det går också att skapa lyssnare av events i kontrollklassen ifall denna har tillgång till ett GUI-objekt. Exempel på detta kan se ut såhär:

```
presenter_view.setOnMousePressed(e -> {
// innehåller en mängd if/else-satser för att fånga
// olika inställningar
// som användaren kan göra som påverkar vad som ska
// hända då musen klickas
})
```

Huvuddelarna av klassen är iallafall att hantera och tolka de olika events som sker på den skrivytan av skärmen för att kunna rita på denna. Den andra delen är att hantera de metदानrop som FXML-filen gör från dess olika knappar. Följande metod returnerar en bild av det som visas på canvasen just nu:

```
public Image getSnapshot()
{
    Image snapshot;
    snapshot = presenter_view.snapshot(null, null);
    return snapshot;
}
```

Följande metod ändrar muspekaren till någonting annat, beroende på funktion. Enbart den delen som ändrar muspekaren till en valfri bild kommer visas, resten av metoden innehåller kod som är snarlik det som visades i PaintApp, relaterat till muspekaren. Som ses så ändras muspekaren till en valfri .png-bild om man har aktiverat zoom-funktionen, i det fallet då man aktiverat drag-funktionen så ändras muspekaren till en handsymbol som redan fanns tillgänglig i JavaFX, och den visas som stängd eller öppen beroende på om man kallar setCursor med OPEN_HAND eller CLOSED_HAND:

```
public void onUpdateCursor()
{
    if (zooming)
    {
```

```

        Image image = new Image("/icons/zoomCursor.png")
            ;
        scene.setCursor(new ImageCursor(image, 16, 16));
        return;
    }
if (dragging)
{
    if (openHand)
    {
        scene.setCursor();
        scene.setCursor(Cursor.OPEN_HAND);
    }
    else if (closeHand)
    {
        scene.setCursor();
        scene.setCursor(Cursor.CLOSED_HAND);
    }
    return;
}
...

```

Följande metod tar en bild av det som visas på den ritbara delen av skärmen, och lägger den bilden på en stack. Stacken representerar föregående ritade bilder och dessa kan återgå till genom att trycka på en knapp som representeras av en vänsterpekande pil. Sedan rensas skärmen på allt som är ritat och undo-and-redo-historiken rensas även. Den här metoden skapar alltså ett nytt "blad" som användaren kan rita på utan att föregående blad försvinner, precis som att vända till en ny sida i ett block:

```

public void onNewSlide()
{
    Image image = getSnapshot();
    prevStack.push(image);
    g.clearRect(g.getCanvas().getLayoutX(), g.getCanvas()
        ().getLayoutY(),
    g.getCanvas().getWidth(), g.getCanvas().getHeight());
    ;
    prevButton.setVisible(true);
    setSlideLabel();
    currentSlideImage = getSnapshot();
    undoAndRedo.clearPaintHistory();
}

```

Följande metod får representera majoriteten av alla "on-X"-metoder som ligger i den här klassen, eftersom de fungerar på väldigt liknande sätt. Först stängs de boolean-fält som representerar att en annan funktion är aktiv av (sätts till falska), och sedan antingen stängs den funktionen som klickats på av eller sätts på, beroende om den redan var av eller på. De här metoderna kallas på när användaren klickar på en knapp

för att aktivera eller av-aktivera en viss funktion i programmet. "Button.setStyle"-delarna ändrar färg på knappen för att visa om den är intryckt eller ej.

```
public void onSelect()
{
    if (zooming) {onZoom();}
    if (dragging) {onDrag();}
    if(eraser) {onErase();}
    if(mark) {onMark();}
    if(selectionMode)
    {
        selectionMode = false;
        onUpdateCursor();
        selectButton.setStyle("-fx-background-color:
            transparent");
    }
    else
    {
        selectionMode = true;
        onUpdateCursor();
        selectButton.setStyle("-fx-background-color: #9a9a9a
            ");
    }
    updateDraw();
}
```

Följande metod används när användaren försöker importera en fil (ej bildfil då de hanteras genom en annan funktion), om det finns innehåll i stacken nextStack så innebär det att den nuvarande vyn är ett "papper" i mitten av ett skapat "block", och programmet meddelar användaren att allt importerat filinnehåll kommer läggas i nya "blad" som kommer läggas sist i "blocket". Detta sker genom att programmet bläddrar fram till sista "sidan" genom *onNext()*-metoden.

```
public void onOpenFile()
{
    if (!nextStack.empty()) {
        Alert alert = new Alert(Alert.AlertType.CONFIRMATION
            , "Do you wish to continue?",
            ButtonType.YES, ButtonType.NO, ButtonType.CANCEL);
        alert.setTitle("Add resources to presentation");
        alert.setHeaderText("Each page will be added after
            the last slide.");
        alert.showAndWait();
        if (alert.getResult() == ButtonType.YES)
        {
            while (!nextStack.empty())
            {
                onNext();
            }
        }
    }
}
```

```

    }
    onOpenFileHelper ();
    ...

```

Följande funktion är en av funktionerna som används för att importera text till programmet från ett antal olika filformat. De filformat som valdes var de som var i vanligast användning, vilket avgjordes genom att undersöka vilka filformat det gick att spara till i de största tillgängliga textredigeringsprogrammen (Google Documents och Microsoft Word). För att hämta själva texten användes det externa biblioteket Apache Tika, som är skapat av Apache Software Foundation och är gratis under Apache License, Version 2.0 (se bilaga X). Detta externa bibliotek har lagts till i programmet och det är genom dess funktioner som text hämtas ur diverse textfiler, vilket ger ett bra resultat från flera olika filtyper. Metoden nedan kollar vilken filtyp "inputFile" består av, och kallar sedan på en annan metod för att extrahera texten, en av dessa metoder kommer visas härnäst. Ifall filen är en MS-Word-fil (.docx) så behövs det även möjligtvis extraheras bilder från dokumentet, om filen är en sådan fil så sätts en boolesk variabel till sann för att signalera detta, och ett annat externt bibliotek (Apache Poi) används istället, på samma sätt som Tika (men med möjlighet för lättare bildextrahering).

```

private void openTextFile(File inputFile)
{
    BodyContentHandler handler = new BodyContentHandler();
    Metadata metadata = new Metadata();
    FileInputStream inputstream = null;
    try
    {
        inputstream = new FileInputStream(inputFile);
    }
    catch (FileNotFoundException e1)
    {
        e1.printStackTrace();
    }
    ParseContext pcontext = new ParseContext();
    String textString = null;
    boolean docx = false;

    if (inputFile.getName().substring(inputFile.getName().
        length()-4).equals("docx") ||
        inputFile.getName().substring(inputFile.
            getName().length()-3).equals("doc"))
    {
        textString = extractStringDocx(inputstream,
            handler, metadata, pcontext);
        docx = true;
    }
    else if (inputFile.getName().substring(inputFile.getName

```

```

        ().length()-3).equals("txt"))
    {
        textString = extractStringTxt(inputStream, handler,
            metadata, pcontext);
    }
    else if (inputFile.getName().substring(inputFile.getName()
        ().length()-3).equals("odt"))
    {
        textString = extractStringOdt(inputStream, handler,
            metadata, pcontext);
    }
    else if (inputFile.getName().substring(inputFile.getName()
        ().length()-3).equals("rtf"))
    {
        textString = extractStringRtf(inputStream, handler,
            metadata, pcontext);
    }

    writeStringToScreen(textString);
    if (docx)
        extractImagesFromDocx(inputFile);

    onNewSlide();
    onPrev();
}

```

Här är ett exempel på en av funktionerna som kan kallas på från den förra funktionen för att extrahera text från ett RichText-dokument. Som kan ses så används Tika-klassen `RTFParser` (Rich Text File Parser) för att hämta texten ur Rich Text Dokumentet.

```

private String extractStringRtf(FileInputStream inputStream,
    BodyContentHandler handler,
    Metadata metadata, ParseContext pcontext)
{
    RTFParser rtfparser = new RTFParser ();
    try
    {
        rtfparser.parse(inputStream, handler, metadata,
            pcontext);
    }
    catch (IOException e)
    {e.printStackTrace();}
    catch (SAXException e){e.printStackTrace();}
    catch (TikaException e){e.printStackTrace();}
    return handler.toString();
}

```

4.3.3 PaintModel

Detta är en mycket kort klass som fungerar som hjälpklass till PaintController. Den innehåller enbart två funktioner, en metod som hjälper användaren att välja en fil från en katalog på datorn. Denna används av de metoder som ska importera antingen en fil eller en bild in i programmet. Metoden gör detta genom att skapa en nytt fönster av som öppnas av en ny initiering av en *FileChooser*. Med hjälp av argumenten byggs fönstret upp med titel och tillåtna filer. Därefter returneras den valda filen från användaren.

Den andra metoden denna klass har är metoden `GetTime()`. Denna metod returnerar aktuella datum och tid.

4.3.4 ImageObject

Den här klassen representerar en importerad bildfil och innehåller den grundinformation som krävs för att rita upp och manipulera bildobjektet på ritskärmen. Standardklassen PaintController har bland annat en lista med alla bildobjekt som existerar på skrivytan. De metoder klassen har är `draw`. Där den ritas sig själv på den canvas som fås som in parameter, `changeImage` som tar hjälp av `PictureHandler` för att kopiera, ta bort, flytta eller ändra storlek på en bild, `deleteImage` och `checkposition` som kontrollerar huruvida bilden befinner sig på koordinaterna som fås som in parameter.

4.3.5 PictureHandler

Detta är en klass som bidrar till arbetet att hantera de bilder som importerats till programmet, och kallas från en `ImageObject`. Denna klass har endast metoden `drawPicture()` som tar en `ImageObject` och en canvas som in parameter. Det första den gör är att skapa ett nytt fönster som placeras framför huvudapplikationens fönster som bilden från `ImageObject` läggs in i. Detta fönstret implementeras sedan med ett flertal lyssnare som kontrollerar ifall musen drar fönstret eller ej, ifall musen befinner sig i kanten av fönstret o.s.v. Dessa lyssnare gör det möjligt att flytta fönstret och ändra storlek på fönstret. Fönstret har också två knappar, `copy` och `delete`. Vid `delete` kallas `ImageObjects delete` funktion och bilden tas bort. Vid `copy` skapas en ny `ImageObject` med samma parametrar som den aktuella.

4.3.6 PresenterViewController

Den här klassen innehåller funktionalitet för att utöka skärmen så att programmet kan användas på två skärmar. Detta gör det möjligt för föreläsaren att ha ett eget fönster innehållande alla menyer och knappar medan lyssnare kan ha ett eget fönster som visar det som finns på skrivytan. Denna klass körs på en egen tråd då den hela tiden uppdateras.

Functionen `run()` uppdaterar sin egen vy genom att ta en skärmdump på skrivytan

från huvudfönstret. Detta görs genom klassen Robot som returnerar en BufferedImage som görs om till en WritableImage. Koden ser ut som följande:

```

Robot robot = null;
    try
    {
        robot = new Robot();
    }
    catch (AWTException e)
    {
        e.printStackTrace();
    }
    screenFullImage = robot.createScreenCapture(screenRect);
    img = SwingFXUtils.toFXImage(screenFullImage, null);

    if (!pc.getFullscreen())
    {
        PixelReader reader = img.getPixelReader();
        img = new WritableImage(reader, (int) pc.
            presenter_view.getLayoutX(), (int) pc.
            presenter_view.getLayoutY(), (int) pc.largePane.
            getWidth(), (int) pc.largePane.getHeight());
    }
    iw.setImage(img);

```

4.3.7 Server

Denna klass öppnar en serversocket på den aktuella datorn. Klassen går sedan in i en loop där den lyssnar efter klienter som försöker ansluta sig till socketen. När en ny anslutning till en klient hittas så skickas denna vidare för att hanteras av klassen clientHandler. Varje ny clientHandler skapas i en ny tråd för att se till att servern kan hantera många klienter samtidigt. Allt detta sker enligt följande kod:

```

while (running)
{
    // wait for a client to connect
    try
    {
        clientSocket = servSocket.accept();
        if (bannedSockets.contains(clientSocket.
            getInetAddress().getHostName()))
        {
            clientSocket.close();
        }
    }
    else
    {

```

```

        Thread t = new ClientHandler(this, clientSocket,
            clientSocket.getInputStream());
        t.start();
    }
}
catch (IOException e){}
}
System.out.println("st nger av");

```

Server är själv en trådklass för att kommunikationen inte ska ta prestanda från huvudprogrammet. Klienthanteraren som skapas för en klient får medskickat en instans av servern för att hantera inkommande meddelanden. Användaren som kallar på funktionen är tänkt att vara anonym då endast frågor skrivs ut i UI och inget namn. Dock i syfte att kunna kontrollera så att chatten inte fylls av spammeddelanden finns en funktion där man kan banna ett specifikt meddelande. Den IP-adress som skickat meddelandet sparas i en lista som servern själv har och ifall en IP-adress som är med i banna-listan försöker koppla upp sig mot servern igen kommer kontakten brytas.

Ett problem som upptäcktes var att en metod som inte körs på samma GUI-tråd som kontrollklasserna till FXML-filerna kommer inte få uppdatera GUI-elementen. För att lösa så att meddelanden lades till när servern fick ett meddelande kommer istället GUI-klassen för chatten, ChatController, kontrollera ifall server har något nytt meddelande. Därför lades metoden getMessage() som returnerar alla meddelanden från en lista. För att inte samma meddelande ska skickas igen kommer server hålla koll på hur många rader denna har skickat redan. Koden ser ut såhär:

```

public synchronized String getMessage()
{
    if(questions.size() <= 0)
    {
        questions.add("Waiting on questions.\n");
        return null;
    }
    else if(rowsSent < questions.size())
    {
        rowsSent++;
        return questions.get(rowsSent-1);
    }
    else return null;
}

```

4.3.8 ClientHandler

Denna klass är en privat klass inom Server-klassen. Det är en trådklass som skapas av varje ny klient-förfrågan på serverSocket och sparar kopplingen till denna genom en vanlig Socket. När objektet skapats går den in i en loop som hämtar data från

Socket och skickar meddelandet till Server för att skrivas ut i chatten. ClientHandler kontrollerar också ifall meddelandet från klienten är olämpligt och kommer då lägga till denna på serverns ban-lista.

4.3.9 ChatController

Denna klass är controller-klassen till FXML-filen Chat-FXML. Funktionen skapar och initierar server-objektet därefter går klassen in i en loop där denna hämtar meddelanden från server klassen med hjälp av metoden getMessage().

4.3.10 Client

Klassen client är en testfunktion som är till för att testa server-client funktionen utanför android miljön. Den försöker skapa en koppling till en viss IP-adress och en port. Vid koppling kommer det gå att skicka meddelanden till servern.

4.3.11 Recorder

Detta är basklassen för videoinspelning av skrivytan. Den påbörjar alla funktioner som har att göra med att spela in bild och ljud, och kallar på andra klasser för att skapa en videofil.

Följande klass startar en ny videoinspelning, och baserat på en boolesk variabel så spelar den även in ljud via klassen Sphinx4SpeechToText", som kommer förklaras mer senare. Tråden som skapas via metoden "createNewThread"är satt att upprepas x-antal gånger per sekund, och skapar en bildfil vid varje upprepning. Det är dessa bildfiler som sedan vävs ihop till en video.

```
public void startRecording(String path, boolean
    isThereAMicrophone)
{
    filePath = path;
    isThereAMic = isThereAMicrophone;

    isRecording = true;
    createNewThread();
    repeatingThread.start();

    if (isThereAMic)
        Sphinx4SpeechToText.start();
}
```

Följande metod tar en skärmbild via java-klassen Robot och returnerar sedan bilden.

```
private static BufferedImage getCanvasCapture()
{
    BufferedImage image = null;
    try
```

```

    {
        image = new Robot().createScreenCapture(new
            Rectangle(Toolkit.getDefaultToolkit().
                getScreenSize()));
    }
    catch (HeadlessException e1)
    {
        e1.printStackTrace();
    }
    catch (AWTException e1)
    {
        e1.printStackTrace();
    }
    return image;
}

```

4.3.12 ImageToMovie

Den här klassen använder JCodec för att skapa en videofil utifrån en mängd bildfiler.

```

public void createVideoFromBufferedImages(String path,
    boolean isThereAMic) throws Exception
{
    filePath = path;
    SeekableByteChannel out = null;
    try
    {
        out = NIOUtils.writableFileChannel(path);
        fileSequence++;
        AWTSequenceEncoder encoder = new
            AWTSequenceEncoder(out, Rational.R(2, 1));

        int i = 0;
        int j = 0;
        while (new File("bufferedImages/capture" + i++ +
            ".jpg").exists())
        {
            BufferedImage image;
            image = ImageIO.read(new File("
                bufferedImages/capture" + j++ + ".jpg"));

            encoder.encodeImage(image);
        }
        encoder.finish();
    }
}

```

```

        finally
        {
            if (isThereAMic)
            {
                File videoFile = new File(filePath);

                filePath = videoFile.getName().
                    replace('.', 'b');
                filePath = filePath + ".mp4";
                videoFile.renameTo(new File(filePath
                ));
                File deleteFile = new File(filePath)
                ;
                deleteFile.deleteOnExit();
            }
            NIOUtils.closeQuietly(out);
        }
    }
}

```

4.3.13 SoundToVideo

Den här klassen använder Xuggle för att lägga in ljud till en tyst videofil. Den tar in en videoström och en ljudström och är baserad på demokod från Xuggles hemsida som syftade till att lägga en mediafil bakom en annan mediafil (så att exempelvis två ljudfiler kunde klippas ihop till en ljudfil). Detta omarbetades så att en videofil och en ljudfil kunde läggas ihop till en videofil med ljud.

4.3.14 Sphinx4SpeechToText

Sphinx4 är ett gratis kodpaket som finns att hämta på nätet och som syftar till att transkribera en talad ljudfil ner till textfil, det vill säga tal till text". Den här funktionen experimenterades med under utvecklingen av projektet, men kvaliteten på transkriberingen blev inte bra nog för att vara värd att implementera. Dock så används den här klassens system för att spela in ljud fortfarande för att spela in ljud i projektet, även om klassens huvudfunktion ej används (eller syftas till att användas).

4.3.15 UndoAndRedo

Den här koden är återanvänd från det tidigare projektet och sköter funktionaliteten med undo- och redo-funktionen av uppritade föremål i programmet. Den är uppbyggd på två olika listor. En lista som innehåller många sammankopplade GraphicsObjects, alltså linjer. Så länge användaren ritan kommer GraphicsObjects lägga till i listan men när användaren väl lyfter pennan kommer det nya nya linjen läggas till. När en linje är klar läggs den också till i en lista kallad paintHistoryList som är en lista med alla linjer i kronologisk ordning.

UndoAndRedo har en funktion kallad redraw som varje gång användaren klickar på undo-knappen kommer rensa skrivytan och rita om alla linjer i historielistan fram till ett visst index. Därefter finns metoderna redo och undo som ändrar indexet för hur många linjer från historielistan som ska ritas upp vid nästa redraw.

4.3.16 GraphicsObject

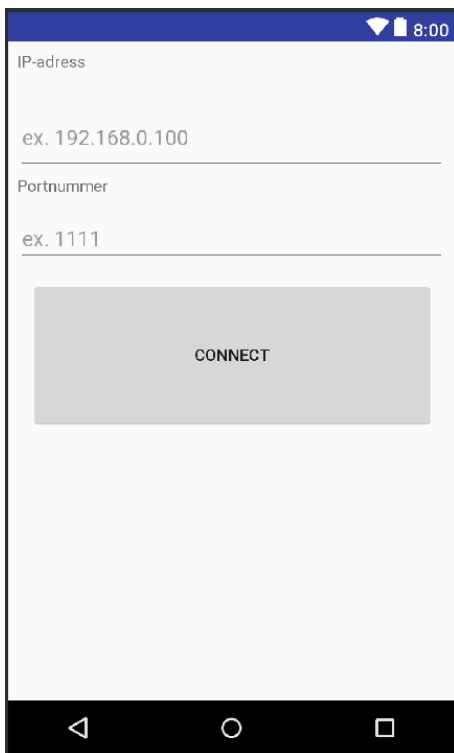
Den här klassen representerar och sköter logiken för ett grafiskt object. Ett grafiskt objekt, till exempel en linje, ritas upp på skärmen med hjälp av inbyggda JavaFX-funktioner som man kallar på genom att skriva bland annat "beginPath()", "moveTo(x,y)", och liknande metदानrop.

4.3.17 FXML-filerna

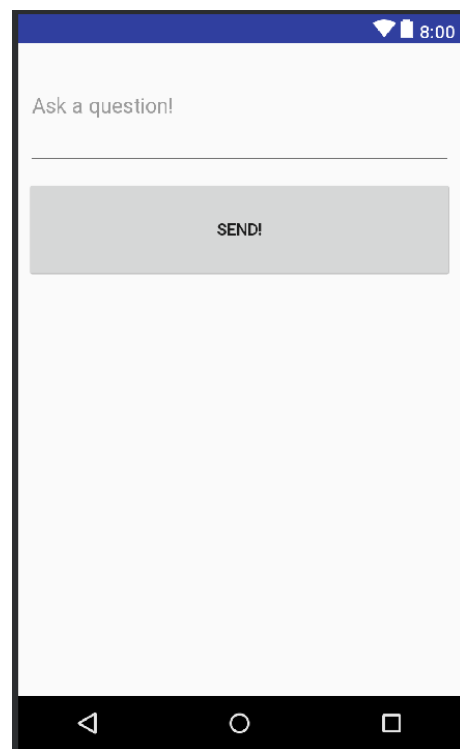
FXML-filernas syfte är att organisera programmets grafiska presentation, dvs placeringen av knappar i fönstret och liknande detaljer. Det är inget som behöver gås igenom närmare. Just nu finns tre olika FXML-filer som används dessa är: paint.fxml som är användargränssnittet för hela basapplikationen, Chat.fxml som är användargränssnittet för chatten och dess knappar. Denna läggs till i paint.fxml och tillhör detta användargränssnitt också. Och sist PresenterView.fxml som är det egna fönstret lyssnarna ser som ska fungera som en avspegling av paint.xmls skrivytan.

4.3.18 Android-klienten

Som nämnts tidigare finns en chat där frågor kan skickas in anonymt till föreläsaren. För att eleverna ska ha möjlighet att skriva frågor utvecklades en Android-app för just detta syfte. Det är en väldigt simpel app med två olika gränssnitt, ett startgränssnitt där man fyller i IP-adress och porten man vill ansluta till (se figur 3). Vid anslutning går appen vidare till nästa gränssnitt där man skriver in och skickar en fråga.



Figur 4.2: Gränsnittet vid start av applikation.



Figur 4.3: Gränsnittet vid anlutning.

5

Resultat

Här presenteras resultaten från projektet. De delar som kommer presenteras är hur pilottestet blev och resultatet från användartesterna och intervjuerna, samt en utvärdering av resultatet i förhållande till målet.

Projektet resulterade i en prototyp av ett program som körs på en föreläsares laptop. Programmet kan liknas vid Microsoft paint, men med funktioner som är specifikt anpassade till att stödja föreläsare i deras undervisning. Det finns möjlighet att rita med olika färger, sudda, zooma, bläddra mellan en mängd olika bilder, importera bilder, texter från diverse textfiler, och powerpoint-slides, samt möjligheten att spela in allt som sker i programmet med både bild och ljud. En mobilapplikation utvecklades även som kan användas tillsammans med programmet så länge både mobilen och programmet är uppkopplade på samma lokala nätverk (samma subnät), man kan då anonymt skicka frågor till programmet som visas för föreläsaren när han trycker på en ikon för att se mottagna meddelanden. Programmet har ej nått ett felfritt slutstadium som är redo för bruk då det innehåller diverse buggar, samt att vissa funktioner behöver mer utvecklingstid för att fungera på ett bra sätt.

5.1 Testning

Testprocessen kom att ändras en del från det som bestämdes från början. Bland annat ansåg vi det inte nödvändigt att ta tid medan testerna utfördes då vi insåg att vi ändå inte kunde jämföra tiderna med något. Istället valde vi att fokusera på att fånga tankar och känslor som testarna uttryckte under testprocessen genom att anteckna vad de sa. Denna metod visade sig inte vara optimal då testarna i vissa fall pratade för snabbt för att allt skulle hinna antecknas. Därav blev anteckningarna en tolkning av vad testaren sa och inte sparad exakt.

5.1.1 Pilottest

Innan de riktiga användartesterna började så genomfördes en mindre pilotstudie med enbart en deltagare, för att undersöka om testprocessen skulle ge värdefull information. En mängd värdefull information kunde antecknas medan testsubjektet utforskade programmet.

Pilottestet visade sig vara väldigt nödvändigt, främst då en rad olika buggar samt problem med användargränsnittet kunde identifieras. Det hittades bland annat att

vid användningen av ”minimera fönster”-funktionen kan ritfunktionen för skrivytan sluta fungera och ibland vid användandet av funktioner så kan muspekaren för skrivytan försvinna och kommer inte tillbaka igen. Dessa var de buggarna vi ansåg vara störst då man måste stänga ner och starta om programmet för att lösa dem. Ett problem som fanns med användargränssnittet var att testaren misstog en bild av en penna (som finns i användargränssnittet) för en knapp och klickade flertalet gånger på denna i försök att få tillbaka vanliga ritfunktionen, efter det att en annan funktion hade använts. Detta görs korrekt genom att av-markera den markerade funktionen genom att klicka på funktionens knapp igen. Detta åtgärdades genom att vi gjorde om ikonen till att vara en knapp, dock används knappen för att ändra storlek på pennan och inte för att få tillbaka ritfunktionen (men det är ändå mer informativt i och med att något händer snarare än inget).

Frågorna ansåg vi, efter pilottestet, vara bra formulerade och ge informativa svar, därför använde vi oss av samma frågor utan omformuleringar även i de riktiga testerna. Vi valde dock att lägga till några fler frågor då det fanns viktigaste områden vi glömt att fråga om.

5.1.2 Användartester

Det skickades ut mail till 24 anställda på Chalmers. Totalt blev det sex anställda som ställde upp och utförde testet. Fem av de sex deltog specifikt i vår målgrupp, det vill säga föreläsare, medans den enda som ej var en föreläsare arbetade med liknande frågor om digitalt lärande och därför ändå var en passande användartestare. Användartesterna var mycket informativa då de, förutom att stöta på redan kända buggar, gav en utomstående synvinkel baserad på stor praktisk erfarenhet inom just föreläsning.

Från intervjuerna kunde urskiljas att åsikterna som skildes åt mellan användartesterna revolverade mest kring hur viktig funktionen med att kunna importera power-point slides var (eller inte var), där en testare ansåg att funktionen borde tas bort helt och hållet eftersom programmet då bara blir en annan version av de många powerpoint-system som redan finns (samt att utlärande i allmänhet bör röra sig bort från att skifta mellan en rad powerpoint-slides), och där andra tyckte att det var programmets mest viktiga funktion, då funktionen erbjuder allt en vanlig powerpoint-baserad presentation kan erbjuda, men med ytterligare möjligheter för interaktivitet (enkel tillgång till ritverktyg, bild- och text-importering, skapandet av videofiler, etc).

De negativa aspekterna som enhetligt pekades ut var främst det flertalet buggar som just nu existerar i programmet, samt klumpigheten hos vissa grundläggande rit-och-zoom funktioner när man jämför med mer professionellt utvecklade program. Den slutgiltiga inställningen hos samtliga testare varierade mellan svagt och mycket positiva. Även de som initialt var skeptiskt inställda blev mer positivt inställda efter att de provat programmet och sett hur enkelt det är att använda, och hur enkelt det är att få en översikt över de få, men noga utvalda, funktioner som är implementera-

de. Den funktion som uppskattades mest överlag var funktionen som gör det möjligt för elever att skicka frågor till föreläsaren, och de flesta ville se den här funktionen utökad med fler möjligheter. Möjligheten att spara frågor för senare besvaring och en möjlighet för föreläsaren att snabbt kunna ställa ja-och-nej-frågor till eleverna genom sitt program, som eleverna kan besvara genom mobil-applikationen, och där resultatet kan visas upp i programmet (exempelvis i form av ett tårt-diagram).

5.1.3 Resultat i förhållande till målsättning

Det mål som sattes upp har uppnåtts till en tillfredsställande grad. Mjukvaruprogrammet som utvecklades innehåller en rad funktioner som syftar till att underlätta för föreläsare och effektivisera föreläsningsspassen. Att deras funktion har uppfyllt har bekräftats av nära på samtliga användartestare. Detta gäller även för målet med att göra programmet lättförståeligt, vilket specifikt pekades ut som en styrka i programmet av en testare.

6

Miljö och Etik

I den här delen kommer etiska och miljömässiga överväganden med projektet beskrivas. Eftersom det är ett programmeringsprojekt som nästan uteslutande berör mjukvara så kommer den här delen att bli kort och översiktlig.

6.1 Miljö

Den fulla konceptidén som tagits fram består av tre hårdvarudelar som bygger upp systemet (laptop, projektor eller skärm, och ritplatta). Tanken med vårt koncept är att laptopen ska tas med av föreläsaren själv, då de flesta föreläsare redan har egna laptops kommer inga nya laptops behöva köpas in specifikt för vårt system. Detsamma gäller för projektorer, de har redan en hög spridningsgrad. På exempelvis Chalmers har alla större rum redan en projektor monterad. Det som behöver köpas in är en stor mängd ritplattor, en för varje klassrum. Detta medför en ökning av produktionen och cirkulationen av elektronik. Ritplattan som vi använt oss av fungerar som en skärm som behöver vara inkopplad i eluttag vid användning och har därmed inga miljöskadliga batterier. Den bärbara pennan som tillhör ritplattan är batterilös och får ström genom EMR (Electro-Magnetic Resonance) teknik [10]. Ritplattorna innehåller fortfarande olika sorters metaller som inte bör komma ut i naturen.

Det går inte att förneka att systemet, med all hårdvara som behöver köras samtidigt, kommer dra mer ström än många andra lösningar. I synnerhet svarta tavlan och whiteboarden. I Sverige är detta inte en stor nackdel då mycket av elkraften som produceras kommer från förnybara källor [11]. Men i ett scenario där ett annat land, där mycket av elen kommer från exempelvis kolkraft, bestämmer sig för att all undervisning ska ske enligt vårt system skulle detta potentiellt innebära en negativ inverkan på miljön.

6.2 Etik

Det finns uppenbara positiva fördelar med produkten (främst effektiviserad utläring), dessa kommer inte gås igenom här. Finns det då några möjliga negativa konsekvenser den här produkten kan ha för samhället? Svaret är nej om vi håller oss till direkta konsekvenser, men en effektiviserad utläring kan självklart användas för negativa utläringssyften, exempelvis utläringen av samhällsskadlig eller kriminell kunskap. Detta är dock ingenting som går att undvika, och inget man kan

förhindra som skapare av produkten. Produkten är ett verktyg som kan användas både för gott och ont, liksom alla produkter som syftar till att stödja olika former av undervisning.

7

Slutsatser och Diskussion

I början av arbetet var det svårt att uppskatta hur stort möjligt genomslag ett sådant här system kan ha. Även om man som teknik och mjukvaruintresserad ser möjligheter och har ett intresse för en viss typ av system så är det möjligt att ens intresse förblindar en för de problem som tekniken kan ha, samt att man sopar undan brister på grund av att någonting är intressant. För att ett sådant här system ska fungera så är det viktigt att den huvudsakliga användarkåren också har intresse för det, och att de finner att det kan hjälpa dem i deras arbete så som det var tänkt. Lyckligtvis så pekar våra användartester på en allmän positiv uppfattning om systemet och dess möjligheter, där även potentiella brister, såsom en väldigt simpel design, uppfattades som styrkor (simpel design = enkel att förstå vid ett ögonkast). Dock måste man ha i åtanke att ifall mer övergripande användartester gjordes så skulle säkert negativa åsikter dyka upp, man kan inte avfärda möjligheten att just Chalmers lärarkår är mer öppna för ny teknik än vad andra lärarkårer möjligtvis är. Det var dock varierande åsikter om sälja detaljerna som ett sådant här system bör innehålla, vissa undervisare uttryckte en önskan att komma bort från powerpoint-slide-systemet som blivit så vanligt i undervisning och ville istället till en mer interaktiv och powerpoint-fri undervisningsform (och antydde därför att möjligheten att importera slides kanske borde tas bort), medans andra upphöjde möjligheten att importera slides som programmets mest viktiga funktion, och ansåg istället att möjligheten att importera och dynamiskt anteckna i och manipulera slidesen som det viktigaste programmet erbjöd. Samtliga undervisare var dock positivt inställda till att använda detta, eller ett liknande system, i framtiden, och därför tror vi att digitalt assisterad undervisning i en liknande form kommer att bli mer populärt i framtiden. Med det kan vi klart säga att det finns ett stort intresse av att denna typ av produkt ska komma att kunna användas på Chalmers och därför anser vi att det här konceptet är något som är värt att vidareutvecklas inom Chalmers, eller eventuellt på någon annan plats.

En viktig del i att ett verktyg som detta kommer fungera bra ligger i att lärarna själva har ett intresse av produkten och att de själva ser möjligheter till att förbättra sin undervisning med det. Detta är ett svårt jobb för en produkt då det går att se att olika föreläsare föredrar olika sätt att lära ut på. Redan bland de sex testarna gick det att se olika behov från denna produkt. En lösning på detta är att skapa en modifierbar öppen lösning där produkten kan anpassas med antingen nedladdningsbara funktioner eller kod som kan skrivas om efter behov. På så vis skulle en föreläsare helt kunna välja vilka funktioner denna vill ha i sitt verktyg. Att göra mjukvaran anpassningsbar på detta sätt skulle fungera väldigt bra med

tanke på att mjukvaran kommer köras från en föreläsares egen dator. Detta skulle till och med kunna leda till mer diskussion bland föreläsare om vilka funktioner just de gillar, och på så sätt inspirera varandra till att testa nya pedagogiska möjligheter.

Under projektets gång har alla slutgiltiga beslut om konceptet och specifika funktioner tagits av gruppens deltagare. Besluten har främst utgått från antaganden om pedagogiska fördelaktigheter och lite grundläggande teori om pedagogik och lärande. Detta arbetssätt har utvecklats då gruppens mål varit att ta fram en utarbetad prototyp snarare än bara ett utarbetat koncept. Arbetssättet har lett till att vissa beslut om funktioner kanske inte blivit ordentligt genomtänkta innan de implementerats, då bättre liknande lösningar kan ha funnits istället. Å andra sidan är det samtidigt lättare att få en uppfattning om en funktions värde samt utvecklingsmöjligheter när funktionen existerar. Med detta sagt hade troligtvis det pedagogiska värdet av lösningen varit bättre ifall projektet hade använt sig av en expertgrupp eller liknande. Men samtidigt hade vi troligtvis inte hunnit implementera tillräckligt många funktioner för att med prototypen visa på alla de möjligheter en digital lösning faktiskt har. Ett alternativt sätt skulle ha varit att skicka ut enkäter till lärare eller föreläsare för att fråga om vilka funktioner de skulle använda i ett liknande program.

Syftet med projektet var att, från en startplats av låg kunskap i ämnet digitalt lärande, läsa på och ta fram ett koncept om ett mjukvaruprogram som skulle effektivisera undervisningen och hjälpa föreläsare under sina föreläsningar. Detta uppnåddes med goda resultat om man ser till reaktionerna under användartesterna, men man kan inte förneka att slutprodukten har en mängd brister och fortfarande är i ett relativt tidigt stadium. Detta var dock förväntat i och med att vi från grunden skulle utveckla både ett (för oss) nytt koncept samt ett nytt program, nästan från grunden. Det huvudsakliga konceptet med mer digitalisering av lärande och inläring är något som kommer att ske naturligt i takt med att tekniken utvecklas, och det är inget man kan komma ifrån. Frågan är bara hur snabbt den här utvecklingen kommer ske. Vi är idag i situationen att nästan varje elev i de industrialiserade länderna har en laptop eller surfplatta som han eller hon använder i sin undervisning, och det är enkelt att se en framtid där föreläsare och elever har en mycket högre grad av kontakt och samspel via sin elektronik under exempelvis föreläsningar.

7.1 Kritisk diskussion

Ifall vi hade fått börja om projektet helt från början men ha kvar de kunskaper som samlats under projektets gång hade vi gjort detta annorlunda:

- Vi hade satt krav på att föra loggbok och anteckna antal timmar jobbade. Då ex-jobbet pågår under så lång tid är det svårt att minnas tidigare anledningar till vissa beslut. I dessa fall underlättar att ha en loggbok att referera till.
- Vi hade tidigt under mjukvaruutvecklingen, när man börjar se vilka olika delar som kommer behövas för programmet, börjat planera strukturen för koden på

ett mer noggrant sätt. Vi hade även omarbetat den gamla koden till en högre grad. Under detta projekt fortsatte vi mest bygga ut den kod som vi fick från början, och omarbetade bara existerande äldre kod från det tidigare projektet när den kom i direkt konflikt med vår egen (eller då den var uppenbart otillräcklig för våra syften). Det har inte existerat någon djupgående planering kring strukturen alls, vilket har lett till en mängd olika klasser med varierande längd vars syfte kan ifrågasättas. Vissa hade kanske kunnat slås ihop, och andra är alldeles för långa för vad de tillför programmet (de innehåller delar som i slutändan inte används av slutprogrammet, vilket gör det svårare att få en översikt).

8

Vidareutveckling av projektet

Projektet resulterade i en prototyp av ett program som skulle kunna användas för att förenkla för föreläsare i deras arbete, och det har stora möjligheter för förbättring och vidareutveckling. Den nuvarande versionen har ett flertal mindre och större buggar som behöver redas ut, och med användartesterna som bas så kan man se en mängd utvecklingsmöjligheter som skulle kunna resultera i en lättanvänd men effektiv produkt.

Den mest självklara utvecklingsmöjligheten är att förbättra online-funktionaliteten så att programmet inte enbart fungerar när både servern och klienterna är i samma lokala nätverk, och sedan utöka den delens funktionalitet så att den även innehåller möjligheten för föreläsaren att snabbt kunna ställa ja-och-nej-frågor, vilka ska kunna besvaras av eleverna i deras mobilapp, där svaren sedan ska tas emot av serverprogrammet och, efter bekräftelse från föreläsaren, visas upp som ett diagram. Den här funktionen skulle även kunna utökas ytterligare med möjligheten för läraren att formulera frågeformulär som eleverna kan svara på online via en mobilapp.

Man hade även kunnat titta på att göra vissa funktioner mer finslipade och användarvänliga, zoom-funktionen är en klar kandidat då den är relativt klumpig. Man skulle även kunna lägga tid på att finslipa funktionen med att importera och visa upp text så att allting är formaterat 1:1 med originaldokumenten (med bilder) från de vanligaste textformaten. Det finns en rad möjliga förbättringsområden och om mer tid läggs på detta program så finns det möjlighet till en riktigt bra slutprodukt.

Litteratur

- [1] *JavaFX FAQ*. 11 juli 2019. URL: <https://www.oracle.com/technetwork/java/javafx/overview/faq-1446554.html#6>.
- [2] James F. Kurose och Keith W. Ross. *Computer networking: A top-down approach*. 7. utg. Pearson education, 2017, s. 114–115.
- [3] James Glynn Jr. *THE EFFECTS OF A FLIPPED CLASSROOM ON ACHIEVEMENT AND STUDENT ATTITUDES IN SECONDARY CHEMISTRY*. URL: <https://scholarworks.montana.edu/xmlui/bitstream/handle/1/2882/GlynnJ0813.pdf?sequence=1&isAllowed=y>.
- [4] Nibal Abdelkarim Mousa Malkawi. *The Effect of Using Smart Board on the Achievement of Tenth Grade Students in English language and on Verbal Interaction during Teaching in Public Schools*. URL: https://www.researchgate.net/publication/320491050_The_Effect_of_Using_Smart_Board_on_the_Achievement_of_Tenth_Grade_Students_in_English_Language_and_on_Verbal_Interaction_during_Teaching_in_Public_Schools.
- [5] Thomas Fridén Emma Fridén. ”*MEN SMARTBOARD HAR VI RÅD MED*” – *EN UNDERSÖKNING KRING PEDAGOGERS SYN PÅ SMARTBOARDS MÖJLIGHET SOM BIDRAG FÖR LÄRANDE I SKOLAN*. 2011. URL: <http://www.diva-portal.org/smash/get/diva2:1308212/FULLTEXT01.pdf>.
- [6] Olof Bälter. *Tips för presentationer*. 17 maj 2019. URL: <https://www.kth.se/social/group/presentationsteknik/page/tips-for-presentationer/>.
- [7] Dennis Sale. *Creative Teaching: An Evidence Based Approach*. Springer Science+Business Media Singapore, 2015. ISBN: 978-981-287-534-1.
- [8] The Room 241 Team. *The History of the Classroom Blackboard*. URL: <https://education.cu-portland.edu/blog/classroom-resources/the-history-of-the-classroom-blackboard/>.
- [9] *Tavlan som rymmer världen – tips för dig som ska köpa interaktiv skrivtavla till skolan*. URL: <https://web.archive.org/web/20100812234000/http://www.interaktivskrivtavla.se/guide-interaktivskrivtavla.pdf>.
- [10] *How Wacom Pens work – EMR (Electro-magnetic Resonance) Explained*. 24 sept. 2019. URL: <https://essentialpicks.com/how-wacom-pens-work/>.
- [11] *Elproduktion*. 24 sept. 2019. URL: <https://www.ekonomifakta.se/fakta/energi/energibalans-i-sverige/elproduktion/>.

A

Checklista till testning

Ritfunktioner:
Byt storlek på pennan till en valfri storlek
Byt färg på pennan till en valfri färg
Rita någonting på skärmen
Sudda ut någonting som redan är ritat
Använd någon funktion för att ta bort allting på den nuvarande bilden (ej sudd-funktionen)
Undo/redo, slide-hantering:
Rita någonting igen, och backa ett steg genom undo-funktionen
Få tillbaka det du ritade genom redo-funktionen
Lägg till en ny slide (ett nytt "papper")
Gå tillbaka till föregående slide
Ta upp en översiktsvy över alla slides
Select-funktionen:
Flytta någonting som redan är ritat till en ny plats
Kopiera någonting som är ritat till en ny plats
Ändra storleken på någonting som redan är ritat
Använd funktionen för att zooma in och ut
Återgå till den vanliga ritfunktionen igen
Chattfunktioner:
Vänta en liten stund medans vi skickar en fråga, och försök sedan läsa frågan
Banna en användare från att använda chattfunktionen
Importerering av diverse filer:
Importera en PDF-fil, Powerpoint-presentation, Keynote-presentation, eller valfri text-fil till programmet (vilken som helst av de jag räknade upp)
Importera en bild till programmet
Avsluta programmet
Inspelning och transkribering
Börja spela in en video, sluta spela in videon, hitta och spela upp videofilen
Starta speech-to-text funktionen, hitta filen, läs den
Zoomning och dragning
Testa zoom-funktionen
Dra skärmvyn med drag-funktionen
Återställ zoomnivån till hur det var från början

B

Intervjufrågor

- Vad var ditt helhetsintryck av programmet?
- Vilka funktioner tyckte du fungerade bra?
- Vilka funktioner tyckte du fungerade dåligt?
- Har du några förslag på förbättringar du vill se?
- Skulle du använda Systemet i undervisningen?
- Vad anser du om grundkonceptet?

C

Funktionstabell

Funktion	Beskrivning	Pedagogiska syfte
Ban user	Banna en användare är en funktion som kan användas för att banna en användare från att kunna koppla upp sig till chattservern igen. Detta leder till att en användare som är bannad inte längre kan skicka in anonyma frågor till frågechatten.	Funktionen är till främst för att enskilda elever inte ska störa läraren med olämpliga frågor eller till och med spam i chatten. Det pedagogiska värdet ligger i att ha möjlighet att ta bort eventuella störningsmoment som kan få läraren att tappa fokus på undervisningen.
Change pen color	Det finns en funktion där läraren kan ändra färg på det penndragen.	Vid användning av olika färger vid förklaring av något kan det bli lättare att förstå.
Change pen size	Med denna funktion kan man ändra storlek på penndragen.	Denna funktion kan vara användbar ifall läraren vill skapa illustrationer som ett pedagogiskt hjälpmedel.
Change window size	Med denna funktion kan man ändra storleken och flytta på programfönstret när det befinner sig i sitt inte maximerade läge.	Med denna funktion är det möjligt för en lärare att hantra fönstret vilket underlättar ifall läraren vill visa något annat fönster.
Chat help	Hjälp-knappen kan användas ifall läraren inte riktigt vet vad frågechatten eller dess funktioner gör för något. Den skriver ut ett meddelande där det förklaras kort om frågechatten och dess tillhörande knappar.	Denna är användbar ifall en lärare är osäker på all funktionalitet hos frågechatten. Då kan läraren få hjälp och lära sig använda alla funktioner fullt ut.

Clear all	Denna funktion kastar den aktuella bilden som är öppen på canvasen. Om det finns en äldre bild kommer denna öppnas annars skapas en ny bild med tom canvas.	Denna är till för att spara tid då läraren annars kanske behövt sudda hela canvasen. Den tid som skulle blivit en paus i undervisningen kan istället skippas.
Disconnect	Denna funktion är till för att kunna stänga av frågechatt-funktionen.	Funktionen kan vara användbar ifall läraren blir mycket störd av frågechatten så att det påverkar undervisningen negativt.
Erase	Med sudd-funktionen kan läraren sudda de delar av canvasen som blivit fel.	Det sparar mycket tid att ha en sudd-funktion då alternativen är att ta bort allt från skärmen och skriva om allt igen.
Exit programme	Denna knapp stänger ner programmet.	Denna har ingen direkt pedagogisk inverkan.
Hide/show chat window	Denna funktion används för att dölja eller visa frågechatten.	Denna funktion är viktig då i nuläget frågechatten tar upp delar av canvasen. För att läraren ska kunna använda sig av hela canvasen ska man då kunna dölja chatten. Det pedagogiska värdet i denna funktion ligger i att chatten kanske kan vara störande för läraren vilket kan leda till sämre undervisning eller så kan helt enkelt läraren bli irriterad på att rutan ligger i vägen och slutar använda produkten.
Import file	Med importera fil-funktionen kan man som det låter importera en fil. I nuläget stödjer programmet filerna PPT, PDF och txt.	Med denna funktion kan man lättare förklara dessa typer av filer då man fortfarande kan använda alla de andra funktionerna som finns för programmet. Man kan bland annat lätt ringa in eller stryka under något i en power point, något som inte går i vanliga fall.

Import picture	Med denna funktion importeras en bild av typen PNG, GIF. Bilden kan sedan flyttas och förstoras eller förminskas för att passa in i övriga objekt på canvasen.	Denna funktion hjälper läraren på så sätt att bilder kan hjälpa till att förstå saker lättare.
Maximize window	Denna funktionen förstorar programfönstret till att ta upp hela skärmen. När skärmen är i fullskärmsläge går det inte att flytta eller ändra storlek på den.	Denna funktion gör det lättare för läraren att använda hela canvasen. Läraren borde få en bättre överblick över den plats denna har att arbeta på och kan då planera sin text bättre. Detta borde leda till ökad inläring hos eleverna.
Minimize window	Denna funktion ändrar programfönstret från fullskärmsläge till inte fullskärmsläge. När denna funktion är aktiv kan läraren också använda sig av ändra storlek på fönstret-funktionen.	Om läraren vill visa öppna upp något annat på sin skärm och visa detta för sina elever är det bra ifall läraren kan minska fönstret. Alternativet är att stänga ner hela programmet och starta det igen, något som skulle ta mycket mer tid. Denna funktion är alltså till för att läraren ska slippa göra tidskrävande aktiviteter där man riskerar att eleverna tappar fokus.
Move focus	Med funktionen flytta fokus kan man navigera sig på canvasen efter att läraren har zoomat in på canvasen.	Denna funktion är främst en hjälpfunktion till zoom och kan komma att spara tid för läraren.
New slide	Denna funktion används för att skapa en ny tom bild.	Denna funktion gör det möjligt att spara tidigare bilder. Tack vare denna funktion är det lätt att implementera spara-funktionen.
Next slide/Previous slide	Nästa bild/Förra bild är två funktioner som används för att navigera mellan bilder som har skapats medan programmet körs. Nästa bild hämtar hämtar och öppnar nästa bild som ligger i en lista av bilder. Förra bilden gör detsamma fast åt andra hållet i listan.	Denna funktion gör det möjligt att kunna enkelt gå tillbaka till bilder och få återkoppling. Denna funktion ger värde då upprepning är ett bra verktyg för att lära in något.

Reconnect	Denna funktion används för att kunna återansluta chatten. När chatten är återansluten kan elever skicka anonyma frågor till chatten igen.	Denna kan användas för att ta emot anonymafrågor direkt från elever.
Redo stroke	Gör om ett penndrag som har ångrats via Undo.	Denna hjälper till att enkelt kunna manipulera innehållet på canvasen för en mindre hackig undervisning.
Save	Spara används för att antingen spara undan den aktuella bilden som är öppen eller alla bilder som skapats under tiden programmet körts. Läraren väljer själv vart på datorn dessa ska sparas.	Denna funktion är till för att underlätta för både lärare och föreläsare. Med möjligheten att spara undan alla bilder från en lektion är det lätt för läraren att lägga upp dessa på internet. Om anteckningarna redan kommer finnas tillgängliga efter undervisningen behöver inte elever längre anteckna själva utan kan lägga allt fokus på undervisningens innehåll.
Screen recording	Denna funktion används för att spela in skärmen och ljudet från undervisningen. Det finns också möjlighet att göra om ljudet till text och på så sätt få ett manus.	Denna funktion är främst riktad till universitetet där skolplikt inte längre är en grej. Med denna funktion kan lärare enkelt lägga upp hela filmen från undervisningen på nätet. Elever som missat lektionen kan enkelt hänga med igen genom att titta på videon. Andra elever kan också lätt kolla på videon för att få upprepning av innehållet.
Select	Markera-funktionen används för att flytta eller manipulera ett område i canvasen. När väl området är valt kan man flytta runt, ändra storlek, ta bort eller kopiera området.	Denna funktion är till för att underlätta för läraren i sitt användande av canvasen. Funktionen kan också användas kreativt för att försöka förklara något tydligare.

Slide overview	Denna funktion används för att välja gå tillbaka till en bild ifall den kanske är väldigt långt bort i listan. Denna funktion kommer ge en tydlig överblick över alla bilder i ett rutnät. Läraren kan klicka på en bild i rutnätet för att hoppa till denna bild direkt.	Funktionen ska användas för att minska tiden det tar att hitta en bild. Denna funktion är till för att inte undervisningen ska behöva få onödigt långa pauser.
Undo stroke	Raderar det senaste penndraget.	Denna hjälper till att enkelt kunna manipulera innehållet på canvasen för en mindre hackig undervisning.
Zoom	Zoom-funktionen används för att zooma in i canvasen.	Kan vara användbar ifall läraren vill förtydliga något specifikt.

D

Intervjuer

D.1 Testare 1

Vad var ditt helhetsintryck av programmet?

Programmet är överlag buggigt, men det har jag överseende med i och med att det är en prototyp.

Vilka funktioner tyckte du fungerade bra?

Sättet man gör slides på och hur man använder dem är bra.

Vilka funktioner tyckte du fungerade dåligt?

Zoomen är inte så viktig för mig och skulle kunna tas bort, dock så kanske den är viktigare om man har liten text i bilden.

Har du några förslag på förbättringar du vill se?

Jag skulle vilja att programmet frågar innan allting suddas ut med ta bort allt-funktionen. Pennan ser ut för mycket som en knapp trots att den inte är det. Zoomningen borde zooma in där man har muspekaren istället för så som det är nu. Jag skulle vilja att frågor till läraren sparades i en textfil, och att det som man ritar på en slide kunde sparas i steg (utan att man spelar in en video).

Skulle du använda systemet i undervisning?

Ja, jag skulle kunna använda ett sådant här system i min undervisning.

Vad anser du om grundkonceptet?

Jag tror absolut att detta är framtiden. Det krävs något mer än standardmodellen där en föreläsare står med ett gäng slides och bläddrar igenom. Möjligheten att få igång en diskussion med eleverna är extra viktig, en dialog mellan han som talar och de som lyssnar.

D.2 Testare 2

Vad var ditt helhetsintryck av programmet?

Programmet är lättanvänt, trevligt och känns 'ofarligt' (ej ett överflöd av funktioner). Gränssnittet är lättläst och bra.

Vilka funktioner tyckte du fungerade bra?

Bild-importeringen ser jag som speciellt värdefull. Spela-in funktionen ser jag som mycket användbar om man castar en lektion hemifrån, så att den senare kan laddas upp på nätet. Jag ser dock inte att den kommer vara så värdefull under föreläsning på plats i skolan/universitetet.

Vilka funktioner tyckte du fungerade dåligt?

Det fanns en mängd mindre användarvänlighetsproblem och mindre missar, till exempel att man automatiskt ritar en prick om man klickar utanför en markerad bild (när man egentligen bara vill att bilden ska bli avmarkerad) och att det är svårt att pricka kanten rätt med muspekaren när man vill skala om en bild. Jag är inte säker på att importera-text funktionen är så värdefull.

Har du några förslag på förbättringar du vill se? Nej, men tanken från den förra intervjupersonen som ni delgav er av angående om att ge eleverna chansen att rösta för om de förstår konceptet som lärs ut för tillfället via nätet anser jag vara en mycket bra idé.

Skulle du använda systemet i undervisning?

Ja, visserligen är jag inte föreläsare men om jag sätter in mig i den situationen tror jag det skulle vara mycket möjligt.

Vad anser du om grundkonceptet?

Jag gillar grundkonceptet och stödjer helhjärtat digitalt lärande. Förr i tiden (under senare 90-talet) så fanns det en del motstånd hos den äldre lärarkåren mot digitalt lärande, man ville göra saker som det alltid gjorts. Men efter generationsskiftet av lärare så har detta blivit allt mer accepterat, och jag tror att det bara kommer bli mer i framtiden, stödet kommer öka i takt med att folk blir vana vid teknologin.

D.3 Testare 3

Vad var ditt helhetsintryck av programmet?

Testaren tyckte det var bra att ha en niche där man kan anteckna på en powerpoint slide. Testaren gillade även de extra funktionerna.

Vilka funktioner tyckte du fungerade bra?

Testaren gillade funktionen där man kan ställa frågor till föreläsaren.

Vilka funktioner tyckte du fungerade dåligt?

Ingen specifik.

Har du några förslag på förbättringar du vill se?

Testaren ville se en förändring av programmet där man skulle kunna se en panorama-vy av bilderna, där man kan se tidigare och senare bilder också och få en bättre överblick i vart man befinner sig i slides. Testaren föreslog också att lägga till en timer för att hålla koll på hur länge man pratat.

Skulle du använda Systemet i undervisningen?

Absolut. Det är klart. Om det fungerar det vill säga.

Vad anser du om grundkonceptet?

Jag är för. Det finns elever som tycker det går för fort med powerpoints då skulle det vara ett bra alternativ att kunna anteckna i dem. Men frågan är ämnesberörande. Systemet passar bättre för vissa ämnen.

D.4 Testare 4

Vad var ditt helhetsintryck av programmet?

Testaren gav programmet 4/5 poäng. Tyckte det var bra med möjligheten att rita och utveckla slides för mer dynamisk föreläsning. Testaren skulle tänka sig använda programmet som powerpoint fast man kan rita också. Gillade tanken att kunna förbereda halvgjorda slides innan föreläsning och sedan lägga till fler bitar under föreläsningen.

Vilka funktioner tyckte du fungerade bra?

Testaren hade inga speciella tankar om vilka funktioner som var extra bra. Men pennfunktionen är helt klart den viktigaste där de andra mest spelar in som komplement.

Vilka funktioner tyckte du fungerade dåligt?

Testaren var skeptisk till chatt-funktionen då det kan vara svårt för en föreläsare att förhålla sig till alla frågor som kommer in. Framförallt var testaren negativt inställd till den anonyma delen av funktionen då man inte vet vem som behöver hjälp. Testaren vill främja kontakt mellan elev och lärare och är för direkt kontakt.

Har du några förslag på förbättringar du vill se?

Testaren föreslog en quiz-funktion där man kan skriva en fråga och skicka ut till eleverna som de kan svara på. Istället för att eleverna skickar frågor kan istället läraren lätt få fram statistik över vad eleverna tänker och tycker. Testaren skulle gilla ifall resultatet kunde synas i ett diagram.

Skulle du använda Systemet i undervisningen?

Mycket möjligt. Testaren tror att denna skulle kunna integrera detta system med sin undervisning för att kunna skapa bättre förutsättningar för att lära sig.

Vad anser du om grundkonceptet?

Testaren tror på digitalt lärande, att det fungerar. Testaren är som tidigare nämnt väldigt inriktad på kontakt och tror det ökar effektiviteten och ett liknande system skulle kunna öka kontakten.

D.5 Testare 5

Vad var ditt helhetsintryck av programmet?

Testaren stötte på smart board-liknande produkt under året och tyckte den verkade bra. Det är bra att det är enkelt att använda produkten. Den viktigaste egenskapen är att man kan skriva saker i PPT. Ritplattan blir där en nyckelgrej för att kunna skriva saker snyggt.

Vilka funktioner tyckte du fungerade bra?

Testaren gillade allt och tyckte att alla funktioner behövdes. Måla är ju såklart den viktigaste så den är extra bra. Men gillar utbytet generellt.

Vilka funktioner tyckte du fungerade dåligt?

Översiktsfunktionen var lite buggig.

Har du några förslag på förbättringar du vill se?

Testaren tycker det är viktigt att fundera kring hur mycket man vill utöka chattfunktionen. Idag använder testaren externa program för att skicka enkäter under lektionstid som kan svaras på med hjälp av mobiltelefon. Det skulle vara jätteskönt att få allt samlat i ett verktyg men man får tänka på att inte lägga till för många funktioner så programmet blir svårt att lära sig och klottrigt.

Skulle du använda Systemet i undervisningen?

Testaren vill använda programmet men mestadels i kombination med PPT importerat i programmet. Testaren tyckte det var viktigt att kunna ha möjlighet att kunna måla figurer eller skriva upp anteckningar i slidesen. Det var ganska snyggt att man kan lägga in egna slides mellan powerpoint bilderna.

Vad anser du om grundkonceptet?

Jag tycker väl det är bra och dåligt. Som föreläsare känner man sig ibland lite handikappad vid användning av bara PPT. Det blir statiskt och det är lätt att det går för fort. Vid användning av skrivtavla tvingas tempot ner då allt skrivs efterhand. Systemet kan öka interaktionen igen som vid användandet av en skrivtavla. Testaren gillar vissa saker med PPT och då är det bra att man kan ta tillbaka de positiva egenskaperna av en skrivtavla med ökad interaktion och minskat tempo samtidigt som man kan ha den digitala lösningen.

D.6 Testare 6

Vad var ditt helhetsintryck av programmet?

Testaren gillade att gränssnittet är enkelt. Ju mindre du har desto bättre. Menade att det var bra att ha standardverktygen och hoppa de andra.

Vilka funktioner tyckte du fungerade bra?

Gillade overview-funktionen för bilder. Gillade att det var valfritt att se chatt-

funktionen eller inte. Ifall den är uppe hela tiden finns risk att det förstöt rytmen då användaren tappar fokus vid ny fråga. Testaren gillade chatt-funktionen generellt. Det kan vara jobbigt med tystlåten publik där man inte får någon feedback. Detta gör det svårt att veta hur föreläsningen går. Testaren ser också potential i ritprogrammet i kombination med PPT.

Vilka funktioner tyckte du fungerade dåligt?

Inga direkt. Det är lätt och bra att kunna byta sida. De knapparna ligger nära varandra på ett logiskt sätt. Det kanske kan bli problematiskt på en mindre platta dock. Man kanske får studera vilket sätt folk tycker det är lättast och effektivast för att hitta en bra lösning på hur menyn som läggs.

Har du några förslag på förbättringar du vill se?

Testaren ser hellre att programmet ska köras på en mobilplatta istället för en ritplatta, för ökad mobilitet. Då kan man använda gester för att zooma och flytta runt skärm o.s.v. Menar att detta är lättare och mer intuitivt. Gav förslaget att kunna ha en fuskklapp för olika slides. Då skulle man kunna skriva stödtext. Konceptet är bra att det går att zooma men man måste fundera mer över vilken hårdvara som används. Det skulle vara bra ifall man kunde ha en kontroll som kan användas för att bläddra mellan bilder.

Skulle du använda Systemet i undervisningen?

Ja visst varför inte.

Vad anser du om grundkonceptet?

PPT har en viss bra del men man förlorar dynamiken. Skrivtavlan har också dåliga egenskaper då man får slut på plats ibland. Det är en begränsning och det tar ganska mycket tid. Jag tycker att less is more angående systemet. Det är bra att ha möjlighet att i realtid lägga till saker och svara på frågor.

Det är nog bra tror jag. Funderar på ifall det är just bra att rita. Bra som komplement. Man måste kunna anpassa sig efter elever och då är det bra att ha möjlighet att välja snabbt och systematiskt eller långsamt och intuitivt. Det är också bra att kunna använda systemet till många olika hårdvaror med olika storlekar för att det ska passa olika typer av lärare.