



Bachelor-Arbeit

ENTWICKLUNG EINES GUI-PROTOTYPEN FÜR ZEIT-METAPHERN IN GRAPHICUSS AUF MOBILEN GERÄTEN

Jasmin Delling

Geboren am: 26. März 1996 in Ebersdorf, Thüringen

Matrikelnummer: 4048254

Immatrikulationsjahr: 2014

zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

Betreuer

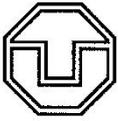
Dr.-Ing. Iris Braun

Dr.-Ing. Tenshi Hara

Betreuender Hochschullehrer

Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Eingereicht am: 7. April 2018



AUFGABENSTELLUNG FÜR DIE BACHELORARBEIT

**Thema: Entwicklung eines GUI-Prototypen für Zeit-Metaphern in
Graphicuss auf mobilen Geräten**

Name, Vorname	Delling, Jasmin	Studiengang	Bachelor Informatik 2009
Matrikel-Nr.	4048254	Projekt/Schwerpunkt	AMCS / Tech-enhanced Learning
Betreuer	Dr. Iris Braun	Weiterer Betreuer	Dr. Tenshi Hara
Startdatum	22.1.2018	Enddatum	7.4.2018

ZIEL

An der Professur für Rechnernetze wurde mit Graphicuss ein Prototyp zur Kombination von klassischen Diskussionsforen mit virtuellen interaktiven Whiteboards entwickelt. Der innovative Ansatz besteht darin, dass alle Beiträge als Canvas-basierte Bildinformation mit vollständiger Erstellungshistorie gespeichert werden. Dies ermöglicht neue Formen von Diskussionen, bspw. können UML-Diagramme bis zu einem zeitlichen Entstehungspunkt zitiert werden, um dann auf einen zu diesem Zeitpunkt gemachten Fehler hinzuweisen. Das gleiche (Teil-)Diagramm kann dann korrigiert fortgesetzt werden. Ein anderer Anwendungsfall ist das Nachvollziehen des Entstehungsprozesses eines Textes. Eventuell sind Absätze in einer anderen Reihenfolge erstellt worden, als sie sich im endgültigen Ergebnis darstellen. Auf diese Art können die Gedankengänge, die gegebenenfalls zu einem falschen Ergebnis geführt haben, nachvollzogen werden, wodurch eine gezieltere Korrektur und Unterstützung im Lernprozess möglich wird. Graphicuss existiert derzeit als web-basierte, vollständig anonym nutzbare Prototyp-Applikation für Web-Browser. Es existieren studentische Vorarbeiten zu Darstellungskonzepten (Metaphern) für den zeitlichen Verlauf sowie zu einem Deanonymisierungskonzept.

Ziel dieser Bachelor-Arbeit ist das Zusammenführen der bisherigen Erkenntnisse in eine auf Smartphones und Tablets nutzbare Client-Applikation. Insbesondere soll dabei auf ein existierendes (bzw. in einem parallel stattfindenden Komplexpraktikum zu implementierendes) Backend zugegriffen werden.

Um das Ziel zu erreichen, soll der Stand der Technik in Bezug auf graphische Eingabekonzepte auf Mobilgeräten untersucht werden. In Kombination mit den vorhandenen Ergebnissen zu Zeitmetaphern soll darauf aufbauend ein Nutzungskonzept erstellt werden. Das Konzept soll im Anschluss prototypisch implementiert werden (bspw. für Android), wobei notwendige Schnittstellen zwischen dem Prototypen und dem Backend definiert werden müssen. Der erstellte Prototyp soll abschließend einer Evaluation unterzogen werden, wobei besonders auf die Usability und Akzeptanz des Nutzungskonzeptes eingegangen werden soll.

SCHWERPUNKTE

- Untersuchung des Standes der Technik in Bezug auf graphische Eingabekonzepte auf Mobilgeräten
- Erstellung eines Nutzungs- und Bedienkonzeptes für Graphicuss-App
- Proof-of-Concept-Implementierung
- Evaluation und Auswertung der Ergebnisse

Iris Braun

ERKLÄRUNG DER URHEBERSCHAFT

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Dresden, 07.04.2018
Ort, Datum

Jasmin Delling

INHALTSVERZEICHNIS

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung	1
1.3. Aufbau der Arbeit	2
1.4. Hinweis	2
2. Analyse des aktuellen Standes	3
2.1. Auditorium Mobile Classroom Service	3
2.1.1. Weiterentwicklung von <i>AMCS</i>	4
2.2. Graphicuss	4
2.2.1. Graphicuss als Webanwendung	4
2.2.2. Visualisierung zeitlicher Erstellungsprozesse	6
2.2.3. Visualisierung der Erstellungshistorie in Graphicuss	7
3. Konzeption	9
3.1. Analysebefragung	9
3.1.1. Vorbereitung	9
3.1.2. Durchführung	11
3.1.3. Resultierende Entwurfsentscheidungen	12
3.2. Anforderungsanalyse	14
3.2.1. Muss-Anforderungen	14
3.2.2. Kann-Anforderungen	14
3.3. Grundelemente	16
3.4. Slideshow-UI	17
3.4.1. Smartphone: Slideshow-Menü	19
3.4.2. Tablet: Slideshow-Timeline	19
3.5. Videoplayer-UI	21
4. Implementierung	23
4.1. Zeichentool	23
4.2. Slideshow-UI	24
4.2.1. Smartphone: Slideshow-Menü	24

4.2.2. Tablet: Slideshow-Timeline	26
4.3. Videoplayer-UI	27
5. Evaluation	29
5.1. Evaluationsziele	29
5.2. Evaluationsbefragung	30
5.2.1. Versuchsaufbau	31
5.2.2. Erhebungsmethoden	32
5.2.3. Durchführung	33
5.3. Auswertung der Evaluationsbefragung	34
5.3.1. Slideshow-UI für Smartphone	34
5.3.2. Slideshow-UI für Tablet	37
5.3.3. Videoplayer-UI	40
5.3.4. Vergleiche der UIs	44
5.4. Android Guidelines	44
5.4.1. Slideshow-UI	44
5.4.2. Videoplayer-UI	46
5.5. Fazit	47
6. Optimierung nach Evaluation	48
6.1. Prototypoptimierung	48
6.1.1. Slideshow-UI	48
6.1.2. Smartphone: Slideshow-Menü	49
6.1.3. Tablet: Slideshow-Timeline	50
6.2. Verifikation der Optimierung	52
6.2.1. Verifikationsziele	52
6.2.2. Verifikationsbefragung	53
6.2.3. Auswertung der Verifikationsbefragung	54
6.2.4. Android Guidelines	58
6.3. Fazit	58
7. Zusammenfassung und Ausblick	60
Literaturverzeichnis	62
A. Anhang	63
A.1. Verwandte Arbeiten	63
A.1.1. Chen, Kaijun: <i>Graphical Discussion System</i>	63
A.2. Analysebefragung	65
A.2.1. Beispiel Analysebefragung	65
A.3. Evaluation	67
A.3.1. Einleitung und Legende	67
A.3.2. Fragebogen Slideshow-UI für Smartphone I	69
A.3.3. Fragebogen Slideshow-UI für Tablet I	71
A.3.4. Fragebogen Videoplayer-UI für Smartphone und Tablet I	73
A.3.5. Fragebogen Vergleich und Persönliches I	75
A.4. Verifikationsbefragung	76
A.4.1. Fragebogen Slideshow-UI für Smartphone II	76
A.4.2. Fragebogen Slideshow-UI für Tablet II	77

ABBILDUNGSVERZEICHNIS

2.1. Zeichentool Webanwendung [Che16]	5
2.2. Screenshot Recording- Panel [She17]	6
2.3. Screenshot Video- Playing-Panel [She17]	6
2.4. Papierprototyp Slideshow-UI mit zusätzlichen Buttons [Ilj17]	7
2.5. Papierprototyp Slider-UI [Ilj17]	8
2.6. Papierprototyp Videoplayer-UI [Ilj17]	8
3.1. Screenshot PC <i>Adobe Acrobat Reader DC</i>	11
3.2. Screenshot PC <i>Microsoft PowerPoint</i>	11
3.3. Screenshot Smartphone <i>YouTube</i>	11
3.4. Screenshot Tablet <i>Netflix</i> (aus rechtlichen Gründen wurde der Hintergrund geschwärzt)	11
3.5. Skizze Menü Slideshow-UI Smartphone	19
3.6. Skizze Menü Slideshow-UI Tablet	19
3.7. Skizze Video- player-UI	21
4.1. Zeichentool	23
4.2. Homescreen Smartphone	25
4.3. Slideshow-UI Smartphone	25
4.4. Slideshow-Menü	25
4.5. Slideshow-UI Tablet mit Timeline	26
4.6. Video- player-UI Smartphone	27
4.7. Video- player-UI mit Mediaplayer	27
5.1. Antworten Slideshow-UI Smartphone Allgemein MF	34
5.2. Antworten Slideshow-UI Smartphone Allgemein EF	35
5.3. Antworten Slideshow-UI Tablet MF	37
5.4. Antworten Slideshow-UI Tablet EF	38
5.5. Antworten Videoplayer-UI MF	40
5.6. Antworten Videoplayer-UI EF	41
5.7. Antworten Vergleiche der UIs	44
6.1. Slideshow-UI Smartphone*	48
6.2. Slideshow-Menü* Smartphone	49
6.3. Slideshow-Menü* erweitert	50
6.4. Slideshow-UI* Tablet	51
6.5. Slideshow-Timeline* Tablet minimal	52

6.6. Slideshow-Timeline* Tablet erweitert	52
A.1. Kursübersicht, <i>Graphicuss</i> als Webanwendung [Che16]	63
A.2. Fragenübersicht, <i>Graphicuss</i> als Webanwendung [Che16]	64

TABELLENVERZEICHNIS

3.1. Einordnung der Befragten Analysebefragung	9
3.2. Konzeption der Bedienelemente: Buttons	17
5.1. Einordnung der Befragten Evaluationsbefragung	31
5.2. Bewertung Slideshow-Buttons für Smartphone	35
5.3. Bewertung Slideshow-Menü für Smartphone	36
5.4. Bewertung Slideshow-UI für Smartphone Allgemein	37
5.5. Bewertung Slideshow-Buttons für Tablet	38
5.6. Bewertung Slideshow-Timeline für Tablet	39
5.7. Bewertung Slideshow-UI für Tablet Allgemein	39
5.8. Bewertung Flag-Buttons Videoplayer-UI	41
5.9. Bewertung Mediaplayer Videoplayer-UI	42
5.10. Bewertung Videoplayer-UI Allgemein	42
6.1. Einordnung der Befragten Verifikationsbefragung	54
6.2. Bewertung Slideshow-Buttons für Smartphone II	54
6.3. Bewertung Slideshow-Menü für Smartphone II	55
6.4. Bewertung Slideshow-UI für Smartphone Allgemein II	55
6.5. Bewertung Slideshow-Buttons für Tablet II	56
6.6. Bewertung Slideshow-Timeline für Tablet II	56
6.7. Bewertung Slideshow-UI für Tablet Allgemein II	57

ABKÜRZUNGSVERZEICHNIS

AMCS Auditorium Mobile Classroom Service

ARS Audience Response System

AI (Entwurfsentscheidungen von) Anastasia Ijassova

E Entwurfsentscheidungen

MA Muss-Anforderungen

KA Kann-Anforderungen

SA Slideshow-UI Allgemein

SD Slideshow-UI Detail

VA Videoplayer-UI Allgemein

VD Videoplayer-UI Detail

EZ Evaluationsziel

FAB Floating Action Button

SPL SlidingPaneLayout

1. EINLEITUNG

1.1. MOTIVATION

In der heutigen Zeit sind technische Geräte allgegenwärtige Begleiter geworden. Rund 80 % aller Deutschen nutzen ein Smartphone, etwa 40 % ein Tablet [eMa16]. Die wohl größten Vorteile dieser Mobilgeräte bestehen darin, zu jedem Zeitpunkt über ihre Internetverbindung schnell und individuell Informationen abrufen oder mit anderen Personen auf die verschiedensten Weisen kommunizieren zu können.

Neben der privaten Nutzung dieser Geräte gewinnen sie vor allem im Bildungsbereich immer mehr an Bedeutung. Sowohl zur Nach-, als auch zur Vorbereitung von Unterrichtsstoff haben sich Onlineplattformen etabliert, in welchen sich Schüler oder Studenten untereinander oder aber mit Lehrenden austauschen und auf spezifische Fragen eingehen können. Einige dieser Plattformen wurden bereits speziell auf Mobilgeräte angepasst, um immer und überall genutzt werden zu können.

Der *Auditorium Mobile Classroom Service*, kurz AMCS, ist ein am *Lehrstuhl für Rechnernetze der Fakultät Informatik der TU Dresden* entwickeltes Online-System. AMCS ist eine anonymisierte Lernplattform, die dazu dient, Lehrveranstaltungen durch Fragestellungen an alle Lernenden während des Unterrichtsgeschehens effektiver zu gestalten. Die gestellten Fragen und die gegebenen Antworten können zudem später zur Nachbereitung oder Prüfungsvorbereitung erneut eingesehen werden. Damit soll der Frontalunterricht an der Universität unterbrochen und so Studierende motiviert werden, sich aktiver zu beteiligen und ihr eigenes Verständnis zu hinterfragen.

Eine vorgesehene Erweiterung des AMCS stellt *Graphicuss* dar. Die Namensgebung setzt sich aus den Wortteilen *Graphic* (Graphik) und *discuss* (diskutieren) zusammen. *Graphicuss* realisiert ein einzigartiges Forensystem, in dem zur Veranschaulichung der gestellten Fragen Graphiken mittels eines Zeichentools erstellt werden können. In diesem Forensystem ist es also möglich, über ein Thema zu diskutieren und Erläuterungen durch Graphiken zu erweitern.

1.2. ZIELSETZUNG

Ziel dieser Arbeit ist es, einen vertikalen Prototypen für die Anwendung *Graphicuss* auf mobilen Geräten mit dem Betriebssystem Android zu erstellen. Der Prototyp soll die graphische Benutzeroberfläche zur Darstellung der Erstellungshistorie für den betrachtenden Nutzer in *Graphicuss* umsetzen.

Dazu werden zunächst Darstellungskonzepte erstellt, die auf bisher erlangte Ergebnisse von [Ilj17] basieren und neue Erkenntnissen einer Anforderungsanalyse einbeziehen.

Anschließend sollen diese Konzepte prototypenhaft implementiert werden.

Das so entstandene Working Partial System wird dann einer repräsentativen Nutzergruppe vorgestellt und von dieser verglichen, sowie bezüglich der Usability bewertet. Die Ergebnisse der Evaluation werden anschließend ausgewertet und eventuell erkannte Mängel durch Änderungen in der Implementierung der Nutzeroberfläche behoben. Das Resultat stellt einen funktionsfähigen vertikalen Prototypen dar, der erfolgreich eine Evaluation, sowie die daraus resultierende Optimierung durchlaufen hat.

1.3. AUFBAU DER ARBEIT

Kapitel 2: Analyse des aktuellen Standes Im Rahmen dieser Arbeit werden zunächst die für das Verständnis wichtige Grundlagen erläutert, indem verwandte Arbeiten vorgestellt werden. Auf den Erkenntnissen dieser Bachelor- und Masterarbeiten wird das folgende Kapitel aufgebaut.

Kapitel 3: Konzeption Anschließend wird eine Analysebefragung durchgeführt, um gegebene Erkenntnisse hinterfragen zu können und die anschließend aufgestellten Anforderungen genauer zu definieren. Schließlich werden unter Berücksichtigung dieser Anforderungen zwei eigene Konzepte für ein User Interface entworfen, die die Erstellungshistorie visualisieren sollen.

Kapitel 4: Implementierung Die erarbeiteten Konzepte werden prototypenhaft in Android implementiert. Der Aufbau der Anwendung und die verwendeten Komponenten der Layouts werden beschrieben.

Kapitel 5: Evaluation Anschließend wird der vertikale Prototyp durch die *Android User Interface Guidelines* und Testpersonen umfangreich ausgewertet.

Kapitel 6: Optimierung Die Evaluation soll zur Auswahl eines der konzipierten Interfaces führen, welches folglich einen zweiten Iterationsschritt durchlief. Dabei werden die im Prototyp ermittelten Schwachstellen durch Änderungen in der Implementierung optimiert. Anschließend erfolgte eine Validierung der Änderungen.

Kapitel 7: Zusammenfassung und Ausblick Abschließend werden die in der Arbeit erzielten Ergebnisse resümiert und weiterführende Ansätze diskutiert.

1.4. HINWEIS

Da die Arbeit thematisch direkt auf der Bachelorarbeit von Iljassova, Anastasia ([Ilj17]) aufbaut, werden die dort erklärten Grundlagen nicht erneut erörtert. Diese Grundlagen umfassen Forensysteme, virtuelle interaktive Whiteboards und geläufige Metaphern auf mobilen Geräten.

2. ANALYSE DES AKTUELLEN STANDES

In diesem Kapitel werden Grundlagen des Themas erläutert. Zudem erfolgt eine Aufzählung und Erläuterung von verwandten Arbeiten.

2.1. AUDITORIUM MOBILE CLASSROOM SERVICE

Der Auditorium Mobile Classroom Service (AMCS) stellt ein *Audience Response System* (ARS) dar. Ein ARS beschreibt Geräte oder Software, das die Interaktivität in Lehrveranstaltungen steigern soll. Die Zuhörer sollen aktiv ins Unterrichtsgeschehen eingebunden werden, was bspw. durch direkte Fragestellungen umgesetzt wird.

AMCS wird am *Lehrstuhl für Rechnernetze* der *TU Dresden* entwickelt und setzt sich aus mehreren plattformspezifischen Anwendungen zusammen.

Lehrende können dabei Kurse und zugehörige Lehrveranstaltungen erstellen. Einer Lehrveranstaltung können Single-, Multiple-Choice- oder offene Fragen zugeordnet werden, die während einer Vorlesungen gestellt und ausgewertet werden können. Außerdem werden Instant-Feedback und Nachrichten mit konditionalen Freigaben ermöglicht.

Innerhalb des Systems werden während des Anmeldevorgangs der Lernenden Pseudonyme zur Anonymisierung eingesetzt. Lernende können die von ihnen besuchte Kurse auswählen und haben anschließend Zugriff auf die sichtbaren Fragen eines Kurses. Gegebene Antworten werden evaluiert und können zu einem späteren Zeitpunkt eingesehen werden.

AMCS nutzt zur Umsetzung der vermehrten und individuelleren Kommunikation zwischen Lernenden und Lehrenden die Methode der *Peer Instruction*. Dieser Begriff wurde maßgeblich von Eric Mazur [Maz96], einem Physikdozent an der *Harvard University*, geprägt. Bei der *Peer Instruction* werden sogenannte *ConcepTest*-Fragen in das Vorlesungskonzept eingebaut.

ConcepTest-Fragen beinhalten zunächst das eigenständige Erarbeiten einer Lösung durch jeden Studierenden und eine anschließende Gruppendiskussion, die *Peer Instruction*.

Bei Einbezug dieser Lehrmethode wird nachweislich das Verständnis der gegebenen Konzepte, die Antwortsicherheit und die Fähigkeit zur fachbezogenen Problemlösung der Studenten gefördert.

Gründe dafür sind vor allem, dass Kommilitonen einen ähnlichen Wissensstand teilen. Außerdem müssen die Studierenden sich vor ihren Kommilitonen in der Diskussion behaupten und hinterfragen so aktiv ihren Lösungsweg. Des Weiteren können Studenten die Schwierigkeiten, die sie beim Erfassen der neuen Thematik hatten, in der Diskussion durch hilfreiche Erläuterungen mit ihren Kommilitonen teilen [MKH17].

Ausführung zum Funktionsumfang von *AMCS* befinden sich in [Kub16] und [Ilj17].

2.1.1. WEITERENTWICKLUNG VON AMCS

Die Masterarbeit von Kubica, Tommy [Kub16] trägt den Titel „Entwicklung eines Prototyps zur Auswahl und zum Einsatz technischer Werkzeuge/Werkzeugkombinationen in unterschiedlichen Lehrformen“. Dabei wurden die konkreten Beispiele des *AMCS* (Kap. 2.1) und des *Exercise Tutorial Tool Kit* (ETTK) für den Einsatz von technischen Werkzeugen herangezogen. Sie wurden zunächst auf vorhandene Funktionen zur Kommunikation zwischen Lehrenden und Lernenden, sowie die Funktionsauswahlmechanismen untersucht und sollten anschließend zu einem Prototypen zusammengefasst werden.

Bei dem ETTK handelt es sich ebenfalls um ein Audience Response System, das allerdings speziell auf die Anforderungen in kleineren Lehrveranstaltungen, wie Übungen oder Seminaren, zugeschnitten worden ist. Während es sich bei *AMCS* um ein gut im Unterrichtsgeschehen erprobtes System handelt, wurde ETTK durch Fast Prototyping entwickelt und bis zu diesem Zeitpunkt weniger gut evaluiert. Auf Grund dessen wurden die Änderungen an Grundzüge von *AMCS* hinzugefügt.

Neu implementierte Funktionen sind bspw. die Möglichkeit für Zuhörer, graphische Antworten abzugeben und während der Lehrveranstaltung anonym Fragen zu stellen. Zudem können Diskussionsfragen und deren Antworten bezüglich ihrer Qualität bewertet werden.

Eine Erweiterung von *AMCS* soll durch *Graphicuss* umgesetzt werden. Die beiden Plattformen sollen künftig ein Login-Portal teilen.

2.2. GRAPHICUSS

Graphicuss ist ein webbasiertes Diskussionsforum, das zur Unterstützung von textuellen Foreneinträgen graphische Beiträge ermöglicht. Innerhalb des Forums können Lernende Fragen stellen, die von Lehrenden oder anderen Lernenden beantwortet werden können. Somit wird ebenfalls die Methodik der Peer Instruction umgesetzt.

Besonders hervorzuheben ist dabei die Möglichkeit, als betrachtender Nutzer die Erstellungshistorie der geposteten Graphiken einzusehen. Durch diese Neuerung kann einerseits der Prozess des Zeichnens genauer wahrgenommen und andererseits spezifisch ab einem bestimmten Punkt im Verlauf zitiert werden. Fehlerquellen können so auch in einer Web-Anwendung konkretisiert werden.

Am *Lehrstuhl für Rechnernetze* wurden bereits wissenschaftliche Arbeiten verfasst, die Teile der Funktionsweise und der Visualisierungsmethoden von *Graphicuss* definieren. Folgend werden die Arbeiten aufgezählt und deren Inhalt erläutert:

2.2.1. GRAPHICUSS ALS WEBANWENDUNG

Die Masterarbeit von Chen, Kaijun [Che16] mit dem Titel „Graphical Discussion System“ definierte erste Grundzüge von *Graphicuss* als Webanwendung. Dazu wurden vor allem die Basisfunktionalitäten des Forums spezifiziert.

Es erfolgte eine Konzeption und eine Implementierung der Webanwendung unter Verwendung eines RESTful APIs und eines *WebSocket*-Protokolls. Ein Fokus lag dabei auf der Umsetzung von Echtzeitkommunikation. Die Architektur wurde sehr detailliert beschrieben. Die umgesetzte Kurs- und Fragenübersicht der Webanwendung sind in Anhang A.1.1 zu sehen.

In Abb. 2.1 ist die Implementierung des Zeichentools sichtbar. Oberhalb der Graphik sind in einer Menüleiste verschiedene Zeichenmodi wählbar. An der rechten Seite ist der Verlauf der Erstellung durch Zeitstempel veranschaulicht worden.

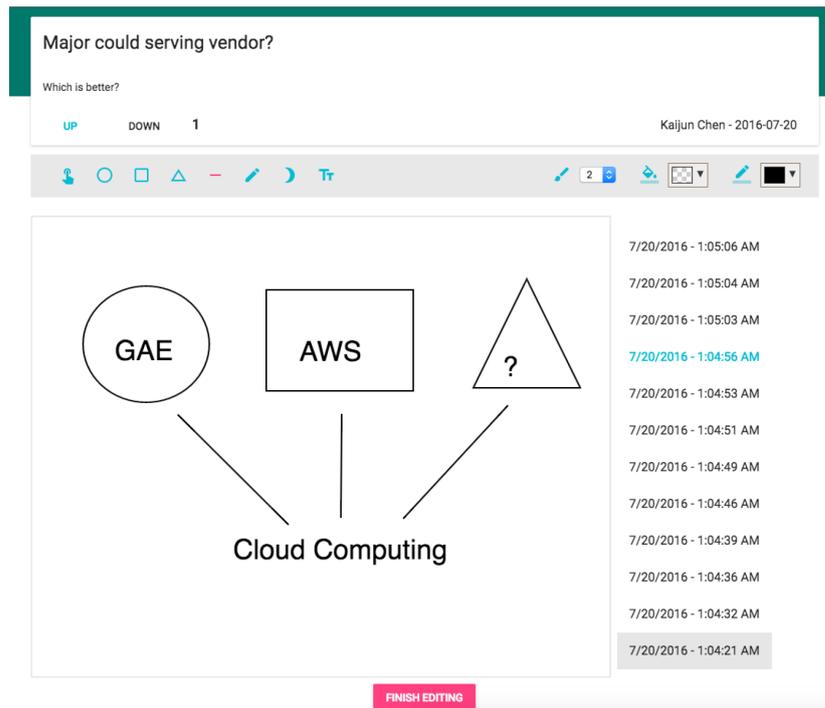


Abbildung 2.1.: Zeichentool Webanwendung [Che16]

Ein Schwerpunkt der Arbeit stellte die Definition eines geeigneten Datenmodells zur Speicherung der gezeichneten Graphiken dar. Dazu wurden Canvas und Scalable Vector Graphics (SVG, englisch für skalierbare Vektorgraphik) verglichen.

Ein Canvas (englisch für Leinwand) ist ein in der Sprache HTML5 standardisiertes Element, das einen Bereich beschreibt, auf dem Objekte gezeichnet werden können. Dazu existieren in HTML5 einige Zeichenfunktionen. Allerdings enthält ein Canvas-Element nach erfolgreicher Zeichnung keine Information über die in ihm gezeichneten Objekte, wie Form oder Größe der Zeichenelemente.

Eine SVG ist eine XML-basiertes Dateiformat zur Darstellung von zweidimensionalen Vektorgrafiken. Wie auch HTML unterstützt SVG eine Vielzahl von vordefinierten Zeichenfunktionen, wie Kreise und Rechtecke. Im Unterschied zu Canvas-Elementen können die gezeichneten Elemente allerdings zu einem späteren Zeitpunkt leicht manipuliert werden.

Zur Festlegung eines Datenformates wurden die Zeiten zum Rendern von verschiedenen vielen Zeichenelementen und verschiedenen großen Zeichenbereichen verglichen, wobei die Canvas-Elemente eine etwas bessere Performance aufwiesen.

Um eine Möglichkeit umsetzen zu können, auch nach der Zeichnung auf dem Canvas Zeichenelemente verändern zu können, wurde als Lösungsansatz ein objektifizierter Canvas vorgestellt. Dabei werden die auf dem Canvas gezeichneten Elemente mit den relevanten Eigenschaften in einem Datenmodell gespeichert.

2.2.2. VISUALISIERUNG ZEITLICHER ERSTELLUNGSPROZESSE

Die Masterarbeit von Shen, Tong [She17] mit dem Titel „Visualizing the Temporal Creation process of Graphical Discussion Contributions“ ergründet die Visualisierung des zeitlichen Erstellungsprozesses von graphischen Diskussionsbeiträgen. Dazu wird zunächst das Konzept von *Graphicuss* als ein online Diskussionsforum vorgestellt. Die Nutzung von Forenbeiträgen, die über einen graphischen Anteil verfügen, zeichnet sich in *Graphicuss* vor allem durch den gespeicherten Erstellungsprozess aus. So können Nutzer im Nachhinein schrittweise die Gedankengänge von geteilten Zeichnungen nachvollziehen. Durch die aktive Beteiligung des betrachtenden Nutzer ist eine zielgerichtete Navigation möglich.

Ein weiteres Konzept wird durch die Nutzung von Webvideos vorgeschlagen. Videos stellen ein grundlegendes Medium des Internets dar. Das Betrachten eines Videos erfolgt weit passiver, als die Verwendung von *Graphicuss*. Zur gedanklichen Rekonstruktion der Erstellung bedarf es keiner Interaktion zwischen Anwendung und Nutzer. Vor allem Lernende können von der Reichhaltigkeit an audiovisuellen, sowie zeitlichen Informationen in Videos profitieren. Deshalb soll die Webanwendung von *Graphicuss* um eine solche Videokomponente erweitert werden.

Nach einer Anforderungsanalyse der Darstellungsprinzipien und einer Evaluation durch Testpersonen wurde deutlich, dass weitere, durch den erstellenden Nutzer definierte Unterteilungen der Graphik notwendig sind, sogenannte Antwortkomponenten. Fortführend werden diese Unterteilungen auch als *Gruppen* bezeichnet. Diese Gruppen sollen eine gezielteren Navigation bei umfangreichen Graphiken erleichtern.

Nach erfolgreicher Implementierung der Erweiterung eines Video-Panels und deren Evaluation wurde deutlich, dass die Umsetzung alle erforderlichen Basisfunktionalitäten beinhaltet. In Abb. 2.2 ist die Ansicht für einen erstellenden Nutzer zu erkennen. Am oberen Rand des Recording-Panels wird eine Toolbar eingeblendet, die zum Aufzeichnen eines Videos dient. Dabei wurde ein Button mit dem Icon einer Flagge verwendet, der dazu dient, wichtige Punkte im Erstellungsprozess zu markieren. Diese Punkte unterteilen die gezeichneten Primitive der Graphik in einzelne Gruppen.

In Abb. 2.3 ist die Ansicht eines betrachtenden Nutzers zu sehen. Am unteren Bildschirmrand befindet sich ein Mediaplayer. In dessen (blauer) Progressbar verdeutlichen rote Markierungen die mit einer Flagge gekennzeichneten Punkte im Erstellungsverlauf.

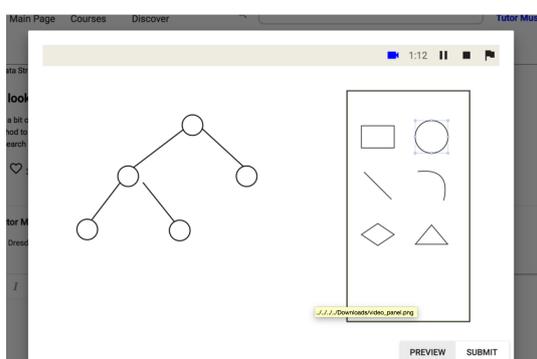


Abbildung 2.2.: Screenshot Recording-Panel [She17]

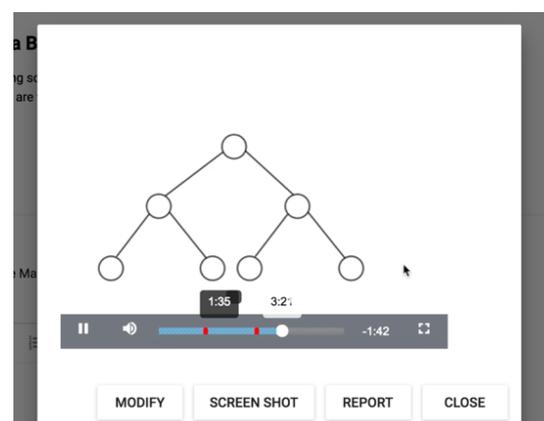


Abbildung 2.3.: Screenshot Video-Playing-Panel [She17]

2.2.3. VISUALISIERUNG DER ERSTELLUNGSHISTORIE IN GRAPHICUSS

Die Bachelorarbeit von Iljassova, Anastasia [Ilj17] trägt den Titel „Nutzerschnittstellenentwurf für unterschiedliche Nutzungskontexte zur Visualisierung von Zeitinformationen innerhalb Canvas-basierter Diskussionsbeiträge in *Graphicuss*“. Die Zielsetzung war es also, eine passende Nutzerschnittstelle für die Visualisierung von Zeitinformationen in *Graphicuss* zu entwerfen. Dazu wurden zunächst mögliche Nutzungskontexte untersucht und anschließend darauf zugeschnittene Zeitmetaphern zur Darstellung der Erstellungshistorie vorgestellt. Nach zwei Iterationsschritten der Evaluation durch Testpersonen mittels Papierprototypen konnte die Auswahl auf drei geeignete Vorschläge für User Interfaces eingegrenzt werden:

SLIDESHOW-UI

Das Konzept des *Slideshow-UI* ist in Abb. 2.4 zu erkennen. Dort befinden sich unter dem dargestellten Canvas vier Bedienelemente, die dazu dienen, eine feste Anzahl von Zeichenschritten in der Erstellungshistorie voranzuschieben bzw. zurückzugelangen. Der Ursprung des Verlaufs befindet sich links. Für die Buttons mit dem Icon eines Doppelpfeils wurde eine Schrittgröße von 5 Zeichenelementen festgesetzt. Die Buttons mit den einfachen Pfeilen dienen dazu, einen Zeichenschritt in der Erstellungshistorie zurückzulegen.

Die zusätzlichen Buttons wurden während des ersten Durchlaufs der Evaluation ergänzt, um die Metapher benutzerfreundlicher zu gestalten.

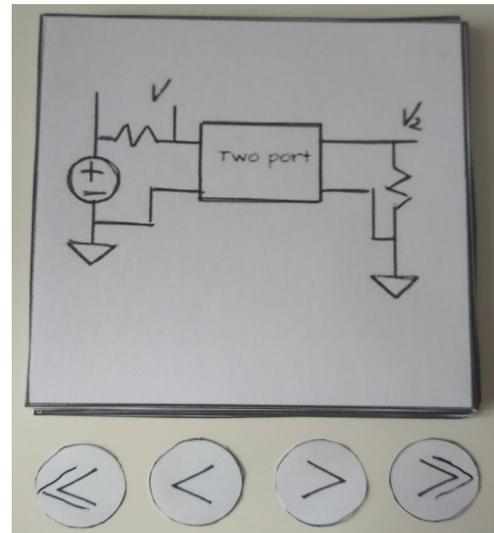


Abbildung 2.4.: Papierprototyp Slideshow-UI mit zusätzlichen Buttons [Ilj17]

SLIDER-UI

Das Konzept des *Slider-UI* ist in Abb. 2.5 zu erkennen. Über dem dargestellten Canvas befinden sich mehrere Vorschaubilder von Zeichenschritten der Erstellungshistorie im sogenannten Slider. Die in der Mitte befindliche Graphik innerhalb des Sliders wird zudem im Canvas im unteren Teil des Bildschirms dargestellt. Der Ursprung der Historie befindet sich links. Zur Verdeutlichung des Fortschrittes im Zeichenverlauf befindet sich direkt über dem Canvas eine Progressbar. Die sichtbare Auswahl im Slider kann mittels Wischgesten auf dem Slider oder auf der Progressbar verschoben werden.

Zusätzlich wurden zwei Buttons eingesetzt, die mit dem Icon eines Doppelpfeils ausgestattet wurden. Durch Klicken auf die Buttons wird der Slider um fünf Positionen verschoben und damit fünf Zeichenschritte weniger bzw. fünf weitere Zeichenschritte auf dem Canvas dargestellt. Die Buttons dienen somit zur schnelleren Navigation.

In der Evaluation wurde deutlich, dass eine Markierung der aktuell dargestellten Graphik im Slider zum besseren Verständnis der Benutzeroberfläche notwendig ist.

Eine Begründung für die Positionierung des Sliders wurde nicht gegeben.

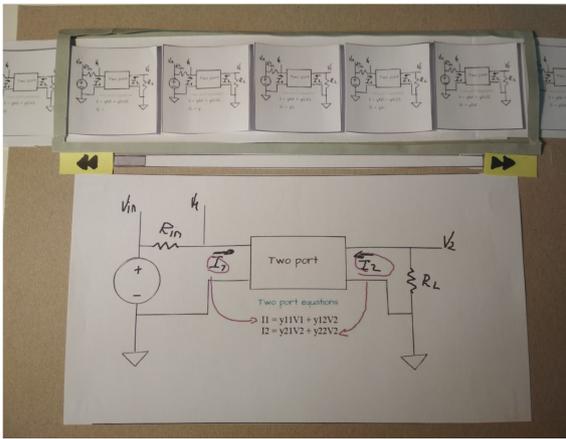


Abbildung 2.5.: Papierprototyp Slider-UI [Ilj17]

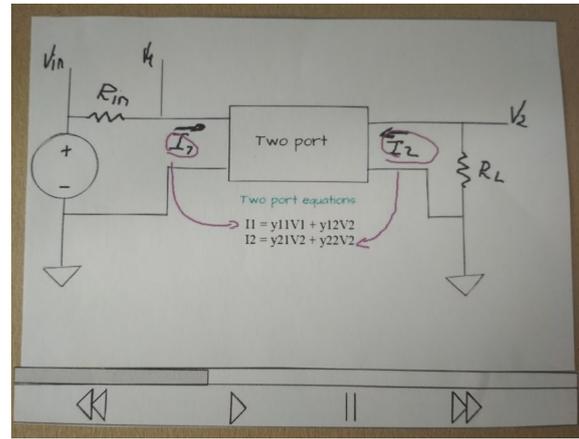


Abbildung 2.6.: Papierprototyp Videoplayer-UI [Ilj17]

VIDEOPLAYER-UI

Das Videoplayer-UI ist dem Slider-UI sehr ähnlich empfunden worden. Angelehnt ist die Gestaltung an das bekannte Interface einer Film-Metapher zum Umgang mit Videodateien, der am unteren Rand des Prototypen in Abb. 2.6 zu sehen ist. Darin enthalten sind ein Play- und ein Pause-Button, durch die das Video gestartet bzw. gestoppt werden kann. Zudem existieren Buttons mit Doppelpfeilen, die ebenfalls eine Navigation in einer Schrittgröße von fünf Zeichenelementen umsetzen. Der Ursprung der Erstellungshistorie befindet sich links. Des Weiteren existierte der Vorschlag, Vorschaubilder beim Hovern über der Videoleiste einzublenden, um die Navigation zu vereinfachen.

EMPFEHLUNGEN

Während der Evaluation wurden entsprechende Ziele festgestellt, die bei der Weiterentwicklung der Interfaces übernommen werden sollten:

A11: Hervorhebung von Änderungen Es wurde festgestellt, dass die Nutzer es als schwierig empfanden, die Veränderung in der Graphik bei einer Navigation in einer Schrittgröße von einem Zeichenelement festzustellen.

A12: Richtung des Zeichenverlaufs Der Nutzer sollte die Möglichkeit haben, den Start- und Endpunkt des Zeichenverlaufs zu invertieren.

A13: Tooltips Es wurde als hilfreich erachtet, Buttons durch Tooltips in ihrer Vorhersehbarkeit leichter zu gestalten. Dabei würden kleine Informationsfenster beim Hovern mit der Maus über dem jeweiligen Button angezeigt werden.

A14: Schrittgröße schnelle Navigations Eine Schrittgröße von fünf Zeichenelementen wurde als idealen Wert für die schnellere Navigation präferiert. Eine schnellere Navigation erfolgte jeweils über die Buttons mit dem Icon eines Doppelpfeils.

A15: Nummerierung der Zeichenschritte Während der Evaluation wurde von den Probanden eine Nummerierung der Zeichenschritte vorgeschlagen. Mithilfe der zusätzlichen Informationen sollen die Buttons mit Doppelpfeilen leichter verständlich werden.

3. KONZEPTION

In diesem Kapitel werden die bisher definierten Entwurfsentscheidungen zum User Interface mit neuen Erkenntnissen kombiniert. So werden Konzepte zur Visualisierung der Erstellungshistorie unter Berücksichtigung der Anforderungen graphische Schnittstellen auf mobilen Geräten entworfen.

Die Entwürfe wurden zunächst auf die Verwendung der Mobilgeräte im Hochformat beschränkt, um die Eignung der verwendeten Metaphern zu testen. Vor allem Smartphones werden hauptsächlich im Hochformat verwendet und zudem häufiger, als Tablets [eMa16].

3.1. ANALYSEBEFRAGUNG

Um einen besseren Einblick in die Anforderungen der Benutzer zu erhalten, wird im Vorfeld der Implementierung eine Umfrage durchgeführt. Vorrangiges Ziel ist, festzustellen, welches mentale Modell von Androidnutzern mit dem Grundgedanken der Applikation am ehesten übereinstimmt und ob eine oder mehrere der von [Ilj17] (Kap. 2.2.3) definierten UIs ihre Erwartungen erfüllt.

3.1.1. VORBEREITUNG

Die Umfrage wird auf fünf Personen beschränkt, da lediglich ein grober Einblick in die Assoziationen der Anwender erlangt werden soll. Die Probanden sind in verschiedenen beruflichen Feldern tätig und zwischen 20 und 42 Jahren alt. Eine Einordnung der Testpersonen erfolgt in Tab. 3.1.

Tabelle 3.1.: Einordnung der Befragten Analysebefragung

Gruppe	Geschlecht	Altersgruppe	Berufsfeld	Anzahl Befragte
IT1	männlich	20 - 39	Informatik	1
IT2	weiblich	20 - 39	Informatik	1
IT4	weiblich	40 - 60	Informatik	1
DI1	männlich	20 - 39	Anderes	2

Um Parallelen sowie Unterschiede zur Implementierung eines Prototypen in iOS zu finden, wird ein iOS-Nutzer zur Befragung herangezogen, wohingegen alle anderen vier Teilnehmer

in ihrem Alltag ausschließlich Android-Geräte verwenden.

Zunächst wird die Zielsetzung der Anwendung erläutert und ein Beispiel anhand von Screenshots (Anh. A.2.1) aus dem Zeichentool des Prototypen vorgestellt. Es wird erwähnt, dass zur besseren Übersicht eine Einteilung der Graphiken in Gruppen vorgesehen ist (Kap. 2.2.2). Aufgrund der häufig offen formulierten Fragestellungen, ergeben sich individuelle Gespräche zu den verschiedenen Entwürfen.

Im Zuge der Umfrage werden folgende Fragen gestellt:

Frage 1: Welche Software oder Anwendung würden Sie mit *Graphicuss* assoziieren?

Diese Frage soll zunächst definieren, welche bereits bekannte Problemstellung und das damit zusammenhängende Bedienkonzept als gleich oder ähnlich empfunden wird.

Frage 2: Erfüllt der gezeigte Papierprototyp Ihre Anforderungen?

Abhängig von den Antworten auf **Frage 1** wird anschließend der Papierprototyp aus [Ilj17] (Kap. 2.2.3, Abb. 2.4, 2.5 und 2.6) vorgestellt, der den Beschreibungen am Ähnlichsten ist. Die Frage hat die Intention, eventuell fehlende Elemente oder Unstimmigkeiten innerhalb der Konzepte zu extrahieren.

Frage 3: Stellen Sie sich vor, Ihnen wird eine Graphik des Verlaufs angezeigt. Wie würden Sie vorgehen, um zur nächsten Graphik zu navigieren?

Auf mobilen Geräten wird eine Bedienung über Wischgesten häufig eingesetzt. Vor allem in der Slideshow-Metapher (Abb. 2.4) wird durch den Nutzer eine Implementierung von Wischgesten vermutet. Durch diese Frage sollen somit notwendige Bedienmöglichkeiten festgestellt werden.

Frage 4: Ordnen Sie die erstellten Graphiken eher als Bilder oder als Dokumente ein?

Aufbauend auf Frage 3 soll in dieser Frage die Wischrichtung innerhalb der Slideshow-Metapher definiert werden. Bei Software, die Dokumente behandelt, wie PDF-Readern bzw. bei Power-Point-Präsentationen oder bei Word wird senkrecht gescrollt während man plattformübergreifend in den meisten Foto-Apps durch horizontales Wischen navigieren kann, um zum nächsten oder vorherigen Bild zu gelangen.

3.1.2. DURCHFÜHRUNG

Frage 1: Welche Software/Anwendung würden Sie mit *Graphicuss* assoziieren?

Drei von fünf Befragten nannten in diesem Zusammenhang Software wie den *Adobe Acrobat Reader DC* (Abb. 3.1) und/oder *Microsoft PowerPoint* (Abb. 3.2).

Signifikant für diese Beispiele ist eine am linken Bildschirmrand angeordnete Übersichtsleiste, mit der innerhalb eines Dokuments durch Klicken zu einem beliebigen anderen Punkt navigiert werden kann.

Im Gegensatz zu *Graphicuss* behandelt diese Art von Software allerdings ausschließlich statische Versionen eines Dokuments und keinen Zeitverlauf.

Innerhalb der Einzelansichten in der Mitte des Bildschirms kann durch vertikales Scrollen an PCs bzw. Wischen auf mobilen Geräten navigiert werden.

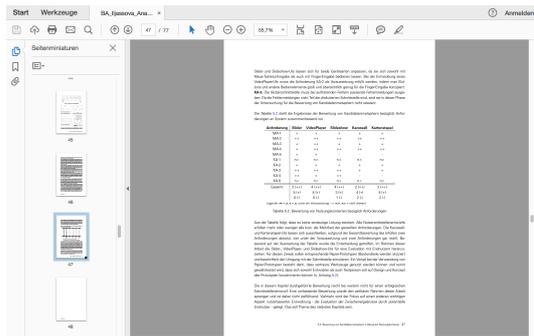


Abbildung 3.1.: Screenshot PC *Adobe Acrobat Reader DC*

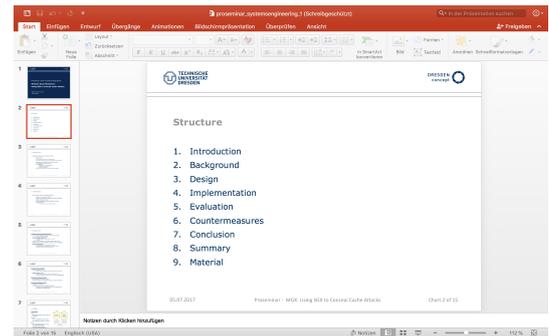


Abbildung 3.2.: Screenshot PC *Microsoft PowerPoint*

Zwei der Befragten assoziierten die Anwendung mit Apps zur Wiedergabe von Videos wie *YouTube* (Abb. 3.3) und *Netflix* (Abb. 3.4). Charakteristisch für diese Anwendungen ist, dass beim Hovern oder beim Wischen über die Zeitleiste am unteren Bildschirmrand ein Vorschauenfenster eingeblendet wird, die das Navigieren erleichtern soll.



Abbildung 3.3.: Screenshot Smartphone *YouTube*

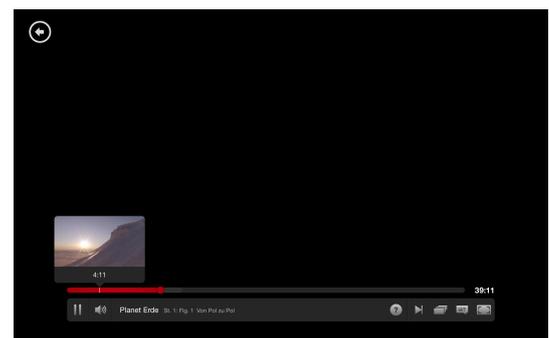


Abbildung 3.4.: Screenshot Tablet *Netflix* (aus rechtlichen Gründen wurde der Hintergrund geschwärzt)

Frage 2: Erfüllt der gezeigte Papierprototyp Ihre Anforderungen?

Drei der Befragten beschrieben mit der in Verbindung gebrachten Software eine Kombination aus dem Slideshow-UI und dem Slider-UI. Diesen wurden anschließend beide Bedienkonzepte zur Evaluation vorgelegt.

Zwei der drei Probanden gaben an, dass ihnen die Slideshow-UI mit zusätzlichen Buttons nicht ausreichen würde, da keine beliebigen zeitlichen Sprünge möglich sind. Ein Proband empfand das Slideshow-UI als ausreichend.

Jeder der drei Befragten gab an, dass das Slider-UI ihre Anforderungen erfüllen würde. Von diesen drei Personen wurde die Platzierung des Sliders als störend empfunden. Grund dafür war vor allem, dass der Slider am oberen Bildschirmrand vergleichsweise schwer zu erreichen ist, sowohl bei ein- als auch beidhändiger Bedienung eines mobilen Gerätes. Außerdem äußerten alle drei Befragten, dass eher eine Platzierung am linken Bildschirmrand erwartungskonform sei, wie bspw. aus Figur 3.1 und 3.2 bekannt.

Zwei der fünf Befragten beschrieben eine starke Ähnlichkeit zum Videoplayer-UI.

Beide gaben an, dass der Papierprototyp ihren Anforderungen entsprechen würden. Ergänzungen wurden dahingehend gemacht, dass Markierungen den Anfang einer neuen Gruppe kennzeichnen sollten. Des Weiteren sollte ein Vorschauenfenster beim Berühren der Zeitleiste eingeblendet werden, um das Navigieren zu erleichtern.

Die genannten Markierungen und das Vorschauenfenster wurden in der Arbeit von [She17] (Kap. 2.2.2) bereits umgesetzt. Die signifikanten Zeitpunkte in der Historie wurden durch Flaggen markiert.

Frage 3: Stellen Sie sich vor, Ihnen wird eine Graphik des Verlaufs angezeigt. Wie würden Sie vorgehen, um zur nächsten Graphik zu navigieren?

Zwei der Befragten gaben an, dass sie zuerst versuchen würden, durch horizontales Wischen/Swipen zu navigieren.

Zwei der Befragten gaben an, dass sie wischen würden, falls keine Buttons vorhanden sein sollten. Anderenfalls würden sie diese bedienen.

Keiner der vier Befragten war der Meinung, dass sie die parallele Nutzung von Buttons und horizontalem Wischen als problematisch empfinden würden.

Der iOS-Nutzer gab an, dass er auf Buttons angewiesen wäre. Er würde ausdrücklich nicht versuchen, über Wischen zu navigieren, da dies nicht seinen Erwartungen entsprechen würde. Insbesondere das Wischen nach rechts würde der iOS-Nutzer mit einer Up-Navigation in Verbindung bringen, also ein Zurückgehen zum vorher besuchten Fenster.

Frage 4: Ordnen Sie die erstellten Graphiken eher als Bilder oder als Dokumente ein?

Jeder der fünf Befragten assoziierten die Graphiken mit Bildern.

3.1.3. RESULTIERENDE ENTWURFSENTSCHEIDUNGEN

Die Analysebefragung wird durchgeführt, um die gegebenen Papierprototypen (Kap. 2.2.3) hinsichtlich der Nutzerfreundlichkeit und der Intuitivität weiterzuentwickeln. Daher werden anschließend konkrete Änderungen vorgenommen.

Im Folgenden sind die resultierenden Entwurfsentscheidungen aufgelistet:

E1: Vereinigung von Slideshow-UI und Slider-UI

Resultierend aus den Antworten auf Frage 1 und Frage 2 werden die Bedienkonzepte von Slideshow-UI und Slider-UI kombiniert. Da insbesondere Rücksicht auf das verwendete Mobilgerät genommen werden soll, werden getrennte Layouts für Smartphones und Tablets entworfen, die allerdings in ihrer grundlegenden Erscheinungs- und Bedienungsweise uni-

form auftreten sollen, um einen evidenten Umgang garantieren zu können.

Im folgenden Text wird als Überbegriff für die Fusion der beiden UIs der Begriff Slideshow-UI verwendet.

Da auf Smartphones vergleichsweise wenig Darstellungsraum zur Verfügung steht, ist eine Verwendung des Slider-UIs dort vermutlich zu unübersichtlich. Deshalb wird in dem Layout für Smartphones das Slideshow-UI um eine zusätzliche Funktionalität ergänzt. Über ein vollständig schließbares Menü soll an jeden beliebigen Punkt innerhalb des Zeichenverlaufs gesprungen werden können.

Bei der Verwendung von Tablets soll der zusätzlich zur Verfügung stehende Raum effektiv genutzt werden. Der Vergleich mit der Software aus Abb. 3.1 und Abb. 3.2 legt eine ähnliche Verwendung einer permanenten Timeline am linken Bildschirmrand nahe.

Um uniform zur Verwendung des Slideshow-UIs auf dem Smartphone agieren zu können, soll im Zentrum des Bildschirms trotzdem genau wie im Layout für Smartphones navigiert werden können.

E2: Wischgesten Slideshow-UI

Die Bedienelemente im Slideshow-UI werden auf Buttons und horizontale Wischgesten festgelegt.

Aus Frage 4 ergibt sich, dass horizontales Wischen verwendet werden soll, da dieses in Foto-Slideshows auf Smartphones gängiger Standard ist.

Resultierend aus Frage 3 wird definiert, dass trotzdem Buttons als Hilfselemente verwendet werden sollen, um dem Nutzer mehrere Möglichkeiten zu bieten, zu benachbarten Graphiken zu gelangen. Insbesondere Buttons sind sehr einfache Bedienelemente, die von einem Großteil der Nutzer sofort als solche erkannt und auch benutzt werden, wohingegen Wischgesten größere Vorkenntnisse und ein höheres Abstraktionslevel voraussetzen.

E3: Markierungen im Videoplayer-UI

Aus Frage 2 ergibt sich für das Videoplayer-UI, dass dort zusätzliche Markierungen zur Kennzeichnung für den Übergang von Gruppen der Graphiken angebracht werden sollen.

Angelehnt an [She17] sollen diese durch die Verwendung eines Flaggensymbols markiert werden.

E4: Vorschau im Videoplayer-UI

Um das Navigieren zu erleichtern, wird außerdem in Frage 2 deutlich, dass eine Vorschau wie in Abb. 3.3 und Abb. 3.4 zu sehen ist, implementiert werden sollte, während die Progressbar des Mediaplayers bedient wird.

3.2. ANFORDERUNGSANALYSE

Um vergleichbare UIs zu schaffen, ist es notwendig, Anforderungen an die beide Prototypen zu definieren. Diese haben sich aus [Ilj17], sowie der Analysebefragung 3.1 ergeben.

3.2.1. MUSS-ANFORDERUNGEN

Um eine Funktionsfähigkeit der Interfaces garantieren zu können, werden Anforderungen definiert, die beide UIs erfüllen müssen.

Die Muss-Anforderungen werden festgelegt auf:

MA1: Darstellung des aktuellen Zeichenschrittes

Die Hauptanforderung der Anwendung ist die Darstellung des Verlauf der Zeichenschritte. Der Großteil des Bildschirms sollte deshalb dazu verwendet werden, einen Canvas (Kap. 2.2.1) darzustellen, auf dem der gewählte Fortschritt des Zeichenprozesses einsehbar ist.

MA2: Navigation zu benachbarten Elementen

Es muss eine Möglichkeit geben, die Anzahl der auf dem Canvas dargestellten Zeichenschritte um eins zu erhöhen bzw. zu verringern. Dieses stellt das wichtigste Bedienelement dar.

MA3: Navigation in festgelegten größeren Abständen

Um bei größeren Zeichenverläufen eine schnellere Navigation zu ermöglichen, muss es ein Bedienelement geben, mit dem mehr als ein Zeichenelement in der Erstellungshistorie vor- bzw. zurückgesprungen werden kann.

MA4: Erleichterung der Navigation zu einem beliebigen anderen Zeichenschritt

Aus der Analysebefragung (Kap. 3.1) hat sich ergeben, dass zudem eine Möglichkeit existieren muss, zu einem beliebigen anderen Fortschritt des Canvas zu navigieren. Dieses Bedienelement muss weniger leicht zugänglich sein, da die Navigation auch durch **MA2** oder **MA3** erzielt werden kann. Ziel dieser Anforderung ist es, durch ein weiteres Bedienelement die Nutzerfreundlichkeit zu maximieren.

Dieser Aspekt gilt als Muss-Anforderung, da bei der Umsetzung die Usability im Vordergrund stehen soll.

3.2.2. KANN-ANFORDERUNGEN

Beide UIs können zusätzlich weitere Anforderungen erfüllen. Diese müssen nicht zwingend umgesetzt werden, erhöhen jedoch die Benutzerfreundlichkeit.

Die Kann-Anforderungen werden festgelegt auf:

KA1: Nummerierung der Zeichenschritte

Um den Nutzer beim Nachvollziehen der Zeichnung zu unterstützen, kann zu jedem Zeitpunkt der Zeichenfortschritt, sowie die aktuelle Gruppe des aktuell dargestellten Canvas angezeigt werden. Dabei werden sowohl Primitive, als auch Text nummeriert. Zudem kann eingesehen werden, wie viele Zeichenschritte und Gruppen insgesamt für die ausgewählte Graphik existieren. Diese Erweiterung wurde durch [Ilj17] initiiert (AI5).

KA2: Bereitstellung von Zeitinformationen

Weitere Informationen können durch den betrachtenden Nutzer erlangt werden, indem er einsehen kann, wie viel Zeit zwischen den gezeichneten Primitiven bzw. der Erstellung des Textes vergangen ist.

KA3: Bereitstellung der Zeichenverläufe

Um die Intention des zeichnenden Nutzers genau nachvollziehen zu können, ist es möglich, die Zeichenprozesse der einzelnen Primitive zu speichern. Dabei könnte in Echtzeit oder in einer festgelegten Zeit der Fingerinput dargestellt werden.

Diese können dem betrachtenden Nutzer helfen, die Fragestellung besser zu verstehen. Dazu kann auch gehören, die Texteingabe buchstabenweise wiederzugeben.

Zudem kann es die Animation der Zeichnungen dem betrachtenden Nutzer erleichtern, Änderungen der Graphik zu erkennen (A11).

KA4: Unterscheidung der UIs von Smartphone und Tablet

Um eine Adaption an die gegebenen Möglichkeiten des mobilen Gerätes zu gewähren, können unterschiedliche Nutzeroberflächen für Smartphones und Tablets existieren.

3.3. GRUNDELEMENTE

Die entworfenen Interfaces beinhalten folgend erläuterte Objekte bzw. Eigenschaften, die zur Darstellung der Erstellungshistorie signifikant sind:

CANVAS

Der Großteil des Bildschirms sollte dazu verwendet werden, einen bestimmten Zeitpunkt der Erstellungshistorie der gewählten Graphik darzustellen. Der Canvas stellt daher das zentrale Element der Anwendung dar und wird mittig platziert.

Die Verwendung eines objektifizierten Canvas zur Darstellung und Speicherung der Graphiken wurde durch [Che16] in Kapitel 2.2.1 definiert.

ZEITSTRAHL

Grundsätzlich werden alle Graphiken der Erstellungshistorie in einem Zeitstrahl angeordnet. Der Verlauf hat dabei in einer horizontalen Zeitleiste seinen Ursprung am linken Ende. Diese Festlegung ergab sich aus der Analysebefragung (Kap. 3.1, E2). Die Orientierung ergibt sich aus dem gängigen Standard bei Videoplayern, sowie der Schreibrichtung der lateinischen Schrift.

Am Anfang des Zeitstrahls wird ein Canvas mit einem gezeichneten Primitiv angezeigt und am rechten Ende des Zeitstrahls die gesamte Graphik.

Die verschiedenen UIs sollen auf dieser Basis Navigationsmöglichkeiten definieren, die es ermöglichen, den Stand des Canvas zu verändern bzw. innerhalb des Zeitstrahls Sprünge zurückzulegen.

Durch 2.2.3 wird festgelegt, dass die Richtung der Historie durch den Nutzer bestimmt werden sollte (A12). Diese Möglichkeit wird jedoch ausgeschlossen, da sie die Komplexität der Interfaces durch größere Freiheitsgrade der Nutzer erhöhen würde.

GRUPPEN

Um bei einer Vielzahl von Zeichenschritten einer Graphik mehr Übersichtlichkeit zu generieren, wird die Unterteilung des gesamten Verlaufs in kleinere Gruppen eingeführt. Diese werden vom zeichnenden Nutzer selbst nach der Erstellung im Zeichentool festgelegt und können einen Titel erhalten.

Unter der Prämisse, dass eine sinnvolle Einteilung der Graphik in Gruppen erfolgt, muss die gesamte Graphik aus mehr als einer Gruppe bestehen. Zudem dürfen keine einelementigen Gruppen existieren.

Diese Annahme wird auf der Basis getroffen, dass die Nutzer gewissensvoll und bedacht mit der Anwendung umgehen, da sie sich Hilfe von anderen Nutzern erhoffen.

BUTTONS

Für eine intuitive Bedienweise müssen eindeutige Bedienelemente mit entsprechender Beschriftung und/oder Icons definiert werden. Angelehnt werden die Bedienelemente an [Ilj17]. Die Bedienelemente sind notwendig, um die Muss-Anforderungen umzusetzen. Deswegen sollten sie in auffälligen Farben gestaltet werden, groß genug und direkt ersichtlich sein.

Tabelle 3.2.: Konzeption der Bedienelemente: Buttons

Icon	Bezeichnung	Beschreibung
▶	Play-Button (Videoplayer-UI)	Dieser Button dient dazu, das Video im Videoplayer-UI abzuspielen. Er wird von einem Pfeil repräsentiert, der in Richtung Ende des Zeitstrahls verweist.
▶	Next-Button (Slideshow-UI)	Dieser Button dient dazu, einen Zeichenschritt mehr auf dem Canvas anzeigen zu lassen. Er wird von einem Pfeil repräsentiert, der in Richtung Ende des Zeitstrahls verweist.
◀	Last-Button	Dieser Button dient dazu, einen Zeichenschritt weniger auf dem Canvas anzeigen zu lassen. Er wird von einem Pfeil repräsentiert, der in Richtung Anfang des Zeitstrahls verweist.
▶▶	Fastforward-Button	Dieser Button dient dazu, eine festgelegte Anzahl an Zeichenschritten mehr auf dem Canvas anzuzeigen oder einen bestimmten Zeitintervall im Verlauf der Erstellung nach vorne zu springen. Er wird von einem Doppelpfeil repräsentiert, der in Richtung Ende des Zeitstrahls verweist. Ein Doppelpfeil kombiniert in diesem Fall die Funktionalität von mehreren Next-Buttons.
◀◀	Rewind-Button	Dieser Button dient dazu, eine festgelegte Anzahl an Zeichenschritten weniger auf dem Canvas anzuzeigen oder einen bestimmten Zeitintervall im Verlauf der Erstellung zurück zu springen. Er wird von einem Doppelpfeil repräsentiert, der in Richtung Anfang des Zeitstrahls verweist. Ein Doppelpfeil kombiniert in diesem Fall die Funktionalität von mehreren Last-Buttons.
▣	Flag-Button	Dieser Button dient dazu, zu einem bestimmten Punkt innerhalb der Erstellungshistorie zu springen. Pro Graphik kann es mehrere Flag-Buttons geben. Sie repräsentieren wichtige Punkte innerhalb des Zeichenverlaufs. Er wird von einer Flagge repräsentiert, da diese wichtige Punkte markieren. Angelehnt wurde die Gestaltung an die Idee von [She17].

3.4. SLIDESHOW-UI

In Kapitel 3.1.3 wurde durch die Befragung erster potentieller Nutzer deutlich, dass Slideshow-UI und Slider-UI zusammengefasst werden sollen (E1).

Im Hauptscreen des UIs wird ein Canvas angezeigt, der den aktuellen Fortschritt der Zeichnung anzeigt (MA1). Beim Öffnen einer Graphik in diesem Interface wird dem Nutzer zunächst das Ende der Erstellungshistorie angezeigt, da dieses alle Zeichenschritte enthält, die Problemstellung zusammenfasst und deshalb einen guten Ausgangspunkt darstellt.

Das Interface erfüllt, wie im Folgenden ersichtlich, alle Muss-Anforderungen und die Kann-Anforderungen 1 und 4.

Buttons Am unteren Bildschirmrand werden zur Bedienung *Next-Button*, *Last-Button*, *Fastforward-Button* und *Rewind-Button* wie im Layout von [Ilj17] (Abb. 2.4) zur Navigation zwischen den Zeichenschritten implementiert.

Next-Button und *Last-Button* dienen daher zur Navigation zwischen benachbarten Zeichenschritten im Verlauf (MA2).

Die Belegung der Fastforward- und Rewind-Button gilt es im Vorfeld festzulegen. Von [Ilj17] (Kap. 2.2.3, A14) wurde als Sprunggröße eine feste Anzahl von fünf Zeichenschritten vorgeschlagen. Ein weiterer Ansatz wäre, zwischen verschiedenen Gruppen navigieren zu können, da diese logisch zusammenhängende Bestandteile der Zeichnung vereinen.

Es wird davon ausgegangen, dass die Gruppen vom zeichnenden Nutzer in sinnvollen Größen erstellt werden.

Da die Festlegung einer Sprunggröße unabhängig des Zeichnungsumfangs zu statisch wäre, wird beschlossen, mittels des Fastforward- und des Rewind-Button zwischen Gruppen zu navigieren (MA3).

Die aussagekräftigen Elemente stellen hierbei die Endpunkte der Gruppen dar, da in dem jeweiligen Canvas alle Zeichenelemente der Gruppe enthalten sind. Deshalb dienen die beiden genannten Buttons dazu, zum Ende der nächsten bzw. vorherigen Gruppe zu navigieren, in sofern der Nutzer sich momentan in einer Ansicht für einen Zeitpunkt der Erstellungshistorie befindet, der einen Endpunkt darstellt. Befindet er sich innerhalb einer Gruppe, also explizit nicht auf einem Endpunkt, gelangt er mithilfe des Fastforward-Buttons zum Ende der Gruppe des aktuell dargestellten Zeichenschrittes. Der Rewind-Button navigiert also immer zum Endpunkt der vorher gezeichneten Gruppe.

Eine beispielhafte Anordnung ist in Abbildung 3.6 im grau hinterlegten Bereich zu sehen.

Wischgesten Aus der Analysebefragung (Kap. 3.1) hat sich ergeben, dass zusätzlich zu den Buttons das Verwenden von horizontalen Wischgesten als intuitiv wahrgenommen wird (E2).

Menü Um alle Anforderungen erfüllen zu können, ist es also zusätzlich zur Slideshow-UI notwendig, ein Menü zu konzipieren, das beliebig große Sprünge von Zeichenschritten ermöglicht (MA4). Innerhalb dieses Menüs wird das Slider-UI (Kap. 2.2.3, Abb. 2.5) umgesetzt. Durch die Analysebefragung (Kap. 3.1) wurde ebenfalls deutlich, dass Menüs am linken Rand des Bildschirms erwartet werden, insbesondere, wenn sie zur Navigation dienen.

Außerdem sollen dem Nutzer die Nummer des aktuellen Zeichenschrittes und der aktuellen Gruppe angezeigt werden (A15). Die Darstellung der Informationen erfolgt innerhalb der in Android häufig verwendeten ActionBar am oberen Bildschirmrand, um den Platz effektiv nutzen zu können (KA1).

Unterscheidung UI Smartphone und Tablet Um die Nutzungskontexte und Möglichkeiten von verschiedenen mobilen Geräten adaptieren zu können, ist es bei diesem Layout sinnvoll, zwischen Smartphones und Tablets zu unterscheiden (KA4). Das primäre Unterscheidungsmerkmal stellt die Größe da, allerdings können auch Tablets mit einer vergleichsweise großen Displaydiagonale eine schlechtere Auflösung aufweisen. Deshalb wurde das Kriterium zur Unterscheidung zwischen Smartphone und Tablet auf die Breite des Displays in Pixel festgelegt. Durch den Vergleich von aktuellen Smartphones [Typ17] ist festzustellen, dass wenige Smartphones eine Displaybreite von mehr als 1500 Pixeln besitzen. Die Grenze wurde weit oben in dem fraglichen Bereich angesetzt, da es zwar möglich ist, auch das platzsparende UI auf Tablets zu verwenden, aber tendenziell nicht funktionstüchtig, wenn die Tabletvariante auf einem zu kleinen Display dargestellt werden soll. Zudem soll die Möglichkeit bestehen, den Darstellungsmodus zu wechseln.

3.4.1. SMARTPHONE: SLIDESHOW-MENÜ

Das Menü auf dem Smartphone soll den beschränkten Platz so effektiv wie möglich nutzen. Deshalb wurde ein Layout (Abb. 3.5) entworfen, bei dem sich ein Menü öffnen lässt, das den Großteil des dargestellten Canvas überdeckt. Der grau hinterlegte Teil stellt den nicht fokussierten Hintergrund dar. Diese Überlappung stellt keine Einschränkung für den Nutzer dar, da vom menschlichen Auge bei einem Leseabstand von 50cm maximal ein Durchmesser von 17 mm fokussiert werden kann [SB16]. Das bedeutet, dass ein Nutzer den Fokus nicht gleichzeitig auf den Canvas und das Menü legen kann, was eine getrennte Darstellung von beiden Items ermöglicht.

Innerhalb des Slideshow-Menüs befinden sich Miniaturansichten von jeder Graphik der Erstellungshistorie. Durch Klicken auf einen Zeichenschritt schließt sich das Menü und der Canvas im Hauptteil des Bildschirms navigiert zum gewählten Punkt im Zeichenprozess.

Um die Nutzung des Platzes zu optimieren, werden innerhalb des Menüs sowohl horizontales, als auch vertikales Navigieren mittels des Konzeptes der **Scrollbars** kombiniert.

Das horizontale Bedienelement repräsentiert den Fortschritt innerhalb einer Gruppe. Durch eine Platzaufteilung, die eine unvollständige Graphik anzeigt, wird dem Nutzer verdeutlicht, dass weitere Zeichenschritte existieren.

Das vertikale Bedienelement repräsentiert den Fortschritt der gesamten Graphik, welcher in Gruppen unterteilt wurde. Auch im vertikalen soll bei der Implementierung weitere Gruppen unvollständig angezeigt werden.

Den jeweiligen Fortschritt innerhalb der Scrollbar kann über eine Progressbar an der beweglichen Achse angezeigt werden.

Das Menü im Layout für Smartphones wird im Folgenden als *Slideshow-Menü* betitelt.

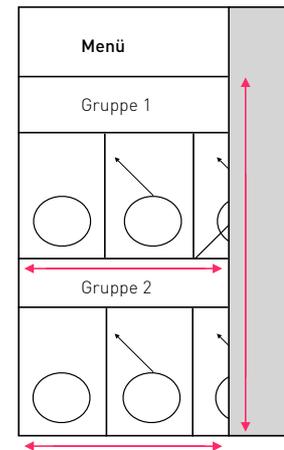


Abbildung 3.5.: Skizze Menü Slideshow-UI Smartphone

3.4.2. TABLET: SLIDESHOW-TIMELINE

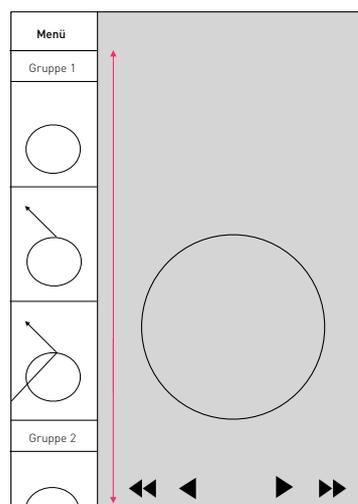


Abbildung 3.6.: Skizze Menü Slideshow-UI Tablet

Da auf einem hier definierten Tablet eine größere Auflösung und damit mehr Raum für die Darstellung eines Menüs befindet, ist es naheliegend, ein ähnliches Interface umzusetzen, wie es in der Analysebefragung 3.1 und bspw. in Abbildung 3.1 bei einem Interface zur Darstellung von pdf-Dateien angewandt wird.

Üblich für diese Art von Layout ist ein Menü, das sich im linken Teil des Bildschirms befindet, das in der Vertikale scrollbar ist. Einen Entwurf dafür ist in Abbildung 3.6 zu sehen. Im rechten, grau unterlegten Teil ist der Canvas zu sehen, der jeweils einen der Zeichenschritte des Verlaufes anzeigt. Im unteren Bereich ist ein Teil der in Kap. 3.3 beschriebenen Buttons zu sehen.

Alle Elemente der Erstellungshistorie befinden sich innerhalb des Menüs vertikal untereinander.

Das Menü ist während der gesamten Nutzung geöffnet, um dem Nutzer die Navigation zu jedem Zeitpunkt zu ermöglichen und einen Interaktionsschritt zu sparen, was eine höhere Effizienz ermöglicht.

Das Menü im Layout für Tablets wird im Folgenden als *Slideshow-Timeline* betitelt, um es von dem Layout für Smartphones begrifflich abgrenzen zu können.

3.5. VIDEOPLAYER-UI

Der größte Unterschied zum Slideshow-UI stellt der geringere Interaktionszwang zwischen Nutzer und Videoplayer-UI dar. Beim Öffnen des Interfaces wird automatisch das Video der Erstellung der gewählten Graphik abgespielt. Es ist somit keine Interaktion durch den Nutzer notwendig, um alle Zeichenschritte des Canvas nachvollziehen zu können.

Außerdem stellt das Videoplayer-UI zusätzliche Informationen dar, die bei dem Slideshow-UI nicht berücksichtigt werden. Durch die Darstellung als Video können sowohl die Dauer des gesamten Zeichenvorganges, die Zeiten zwischen den Zeichenschritten und der Verlauf der Zeichnung von einzelnen Primitiven veranschaulicht werden. Zudem ist es möglich, den Verlauf der Texteingabe des Nutzers, sowie die jeweils ausgewählten Primitive aufzuzeichnen. Allerdings gilt es zu beachten, dass Videodateien bezüglich einer späteren Bearbeitung weniger flexibel sind.

Das Interface erfüllt, wie bisher und im Folgenden ersichtlich, alle Muss-Anforderungen und die Kann-Anforderungen 1, 2 und 3.

Canvas Bei der Konzeption wird die Absicht verfolgt, zwischen den UIs eine größtmögliche Diskrepanz zu generieren, um den später befragten Personen ein breiteres Spektrum von unterbreiteten Informationen und damit an Darstellungsmöglichkeiten darbieten zu können. Deshalb werden im Videoplayer-UI alle Informationen, die der Nutzer bei der Eingabe einer Graphik erstellt, gespeichert und dem betrachtenden Nutzer vorgestellt (KA2, KA3).

Im Detail heißt das, dass der später betrachtende Nutzer sich genau den Bildschirminhalt in Echtzeit ansehen kann, den der zeichnende Nutzer bei der Erstellung wahrgenommen hat. Die Anwendung würde dann sogenannte Screencasts, also Aufnahmen von Bildschirmhalten, generieren bzw. speichern und diese ähnlich wie ein Videoportal bereitstellen.

Der Großteil des Bildschirms wird also im Interface des betrachtenden Nutzers dazu verwendet, den Canvas darzustellen, der sich im Zeichenprozess befindet. In der Abbildung 3.7 ist dieser Bereich grau unterlegt (MA1). Zur Vereinfachung wird der Touch-Input des Nutzers mit einem farbigen Kreis auf dem Bildschirm symbolisiert.

Ein weiterer deutlicher Unterschied zum *Slideshow-UI* stellt der Startpunkt für die Interaktion mit der Graphik dar. Der Ausgangspunkt ist im *Videoplayer-UI* der Anfang der Graphik, da der gesamte Verlauf im Anschluss betrachtet werden kann.

Außerdem sollen dem Nutzer die Nummer des aktuellen Zeichenschrittes und der aktuellen Gruppe angezeigt werden (IA5). Die Darstellung erfolgt innerhalb der in Android häufig verwendeten Actionbar am oberen Bildschirmrand, um den Platz effektiv nutzen zu können (KA1).

Navigationsmöglichkeiten Die Bedienelemente zur Navigation innerhalb des Videoplayer-UIs sollten, um für die Nutzer verständlich zu sein, ähnlich zu bekannten Interfaces auf Videoplattformen gestaltet werden. Ein Beispiel für einen Videoplayer ist in Abb. 3.3 und Abb. 3.4 zu sehen. Auf deren Basis wurde ein eigenes Konzept entworfen, das in 3.7 beispielhaft auf der Größe eines Smartphones zu sehen ist. Es enthält am unteren Bildschirmrand ein Menü, das immer eingeblendet werden soll. Einerseits wird dadurch dem Nutzer verdeutlicht, dass ein Menü existiert und andererseits vereinfacht es die Bedienung, da keine Interaktion zum Öffnen des Menüs vorausgesetzt wird. Durch die genannten Beispiele wurde das Menü erwartungskonform platziert.

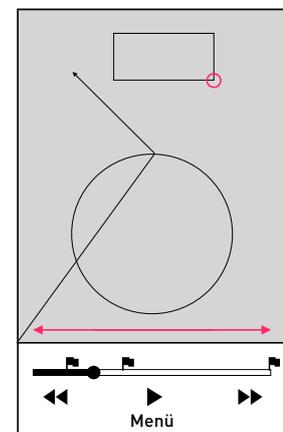


Abbildung 3.7.: Skizze Videoplayer-UI

Im Folgenden wird das Menü im Videoplayer-UI als Media-player bezeichnet.

Buttons In diesem Interface werden Play-Button, Fastforward-Button und Rewind-Button umgesetzt (MA2, MA3).

Fastforward-Button und Rewind-Button dienen in diesem Fall nicht dazu, zu einer konkreten weiteren Graphik zu navigieren, sondern ein festes Zeitintervall zurückzulegen. Dabei wurde die Größe des Sprungs in die natürliche Richtung des Zeichenverlaufs mit dem Fastforward-Button bewusst größer gewählt, da der Zeitraum auch mit dem Play-Button absolviert werden könnte.

Die Sprunggröße für den Rewind-Button wird testweise auf 10 Sekunden und für den Fastforward-Button auf 20 Sekunden festgelegt. Die Festlegung basiert auf der Überlegung, dass für die Zeichnung eines Primitivs etwa 10 Sekunden erfordert. Die Sprunggröße für den Fastforward-Button wurde größer gewählt, da ein Voranschreiten in der Richtung des Zeitverlaufs auch durch den Play-Button ermöglicht wird.

Außerdem befinden sich Flag-Buttons auf dem Progress-Slider des Mediaplayers. Diese symbolisieren, wie im Slideshow-UI, markante Zeitpunkte im Verlauf der Erstellungshistorie (E3). Diese markanten Punkte wurden bei beiden UIs auf das Ende einer Gruppe festgelegt.

Wischgesten Des Weiteren befindet sich im Mediaplayer eine Zeitleiste, der sogenannte *Progress-Slider*, am oberen Rand des Menüs. Auf dieser ist zu jedem Zeitpunkt der Fortschritt der Erstellung sichtbar. Durch Drag and Drop kann der schwarze Punkt auf der Leiste bewegt und so ein beliebiger Zeitpunkt der Erstellung auf dem Canvas abgebildet werden (MA4).

Um die Navigation bei der Drag and Drop-Methode zu vereinfachen, könnte wie in Abb. 3.4 ein Vorschauenfenster während des Drag-Prozesses eingeblendet werden, während der Canvas seine Ansicht beibehält (E4). Allerdings ist das Vorschauenfenster relativ klein und deshalb bei größeren Graphiken bzw. kleinen Bildschirmen wenig aussagekräftig.

Eine andere Variante wäre, dass der gesamte Canvas die Ansicht des Zeitpunktes darstellt, die während des Drag-Prozesses vom Nutzer ausgewählt wird. Somit fungiert der gesamte Canvas als Vorschauenfenster.

Zusätzlich zu den dargestellten Inhalten in der Skizze soll die Länge des ganzen Videos und der aktuelle Zeitpunkt in Minuten und Sekunden angezeigt werden. Diese Informationen sollten, wie in anderen Mediaplayern üblich, unterhalb der Progress-Slider platziert werden, um einen späteren Bezug auf den genauen Zeitpunkt zu ermöglichen.

Unterscheidung UI Smartphone und Tablet In diesem Interface erfolgt keine Unterscheidung zwischen Smartphone- und Tabletversion, da das Menü einen relativ kleinen Anteil des Bildschirms bedeckt.

4. IMPLEMENTIERUNG

Es folgt eine Ausführung der implementierten Komponenten anhand vorgegebener Maßstäbe und während der Analyse getroffene Entwurfsentscheidungen.

Der Prototyp wird komplett in *Android Studio* in Java und XML implementiert. Die Software setzt mindestens ein API-Level von 21 voraus, was Android 5.0 (Lollipop) entspricht.

Das Farbschema entspricht dem von Android vorgeschlagenen Standard.

4.1. ZEICHENTOOL

Die Grundzüge, die das Zeichentool zur Erstellung der Graphiken beinhalten sollte, wurden bereits von [Che16] (Kap. 2.2.1) definiert. Zusätzlich dazu wurden im parallel stattfindenden Komplexpraktikum am *Lehrstuhl für Rechnernetze* folgende Primitive festgelegt:

- Linien
- Pfeile
- Rechtecke
- Ovale

Zudem ist es den Nutzern möglich, Text auf den gegebenen Canvas zu zeichnen. Dieser soll der Beschriftung der Primitive dienen. Die Beschreibung des Problems für den vollständigen Foreintrag erfolgt zusätzlich zu der generierten Graphik.

Explizit ausgeschlossen wurde die Möglichkeit der Freihandzeichnungen. Die Entscheidung wurde im Vorfeld am *Lehrstuhl Rechnernetze der TU Dresden* in Kooperation mit Psychologen getroffen. Zurückzuführen ist diese Entscheidung auf die Tatsache, dass willkürliche Zeichnungen von den festgelegten Primitiven abstrahiert werden können und sich die Speicherung der Objekte vereinfacht.

In Abbildung 4.1 ist die Verwendung aller Zeichenmöglichkeiten zu sehen. Die Auswahl erfolgt über eine *Toolbar* am unteren Bildschirmrand. Alle Primitive werden zunächst durch ein Icon repräsentiert und nach ihrer Auswahl mit einer Beschriftung dargestellt.

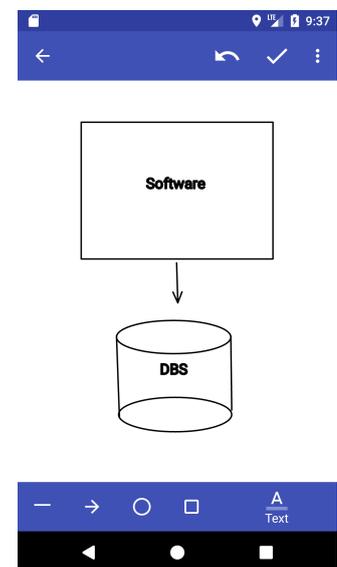


Abbildung 4.1.: Zeichentool

In der *Actionbar* am oberen Bildschirmrand sind außerdem Icons für das Rückgängigmachen (Undo) von Zeichenschritten und das Bestätigen einer abgeschlossenen Graphik (Confirm). Im Menüpunkt am rechten oberen Bildschirmrand kann die Zeichenfarbe gewählt werden.

4.2. SLIDESHOW-UI

Homescreen In Abb. 4.2 wird der Homescreen dargestellt, in der man zukünftig alle bereits erstellten Graphiken finden kann. Sie sind bisher in einer *GridView*, also einer Gittersicht, angeordnet. In der späteren Implementierung sollten hier zudem der Titel und eine kurze Beschreibung die vorgestellte Problematik genauer erläutern, wozu eine Listenansicht (*List-View*), wie aus Foreneinträgen bekannt und in Abb. A.2 zu sehen ist, die Anforderungen besser erfüllen könnte. Durch den *Floating Action Button* in der rechten unteren Ecke des Bildschirms wird das Zeichentool geöffnet.

Durch Klicken auf eine Graphik, die man sich im Detail ansehen möchte, gelangt man zu einer Ansicht wie in Abb. 4.3 auf dem Smartphone-UI bzw. Abb. 4.5 auf Tablet-UI.

Detailscreen In dieser Ansicht wird im Zentrum ein bestimmter Fortschritt des Canvas dargestellt. Zunächst wird die gesamte Zeichnung inklusive aller Zeichenschritte angezeigt. Jedes gezeichnete Objekt der Graphik wird von einem Canvas vertreten, das die Historie der Zeichnung bis zu dem bestimmten Zeichenschritt zeigt. Diese Canvas werden in einem *Viewpager* horizontal miteinander verbunden, ähnlich wie herkömmlichen Foto-Apps für Smartphones. Der Ursprung der Erstellungshistorie ist dabei am linken Ende.

Durch horizontales Wischen kann auf dieser Timeline zwischen benachbarten Graphiken navigiert werden, d.h. es befindet sich nach der Interaktion ein Primitiv mehr bzw. weniger auf dem Canvas. Die Wischgestik wurde durch den verwendeten *Viewpager* unterstützt.

Zudem wird in der *Actionbar* am oberen Bildschirmrand permanent die aktuell dargestellte Graphik und die dazu gehörige Gruppe angezeigt. Dadurch soll die Übersichtlichkeit garantiert und die Graphiken für die Nutzer verständlicher machen werden. Die Positionierung dieser Information wurde ebenfalls aus Platzgründen gewählt.

Buttons Zur Darstellung der Buttons wurden *Floating Action Buttons* verwendet. Diese werden einzeln auf dem Canvas platziert, um möglichst wenig Fläche in Anspruch zu nehmen. Die *Floating Action Buttons* wurden gewählt, da sie standardmäßig in auffälligen Farben, dem *colorAccent* einer Android-App, auftreten und wichtige Funktionen repräsentieren. Sie werden zudem ausgeblendet, wenn aktuell der Anfang oder das Ende der Timeline im Canvas dargestellt werden.

4.2.1. SMARTPHONE: SLIDESHOW-MENÜ

Um zudem eine Navigation zu beliebigen Graphiken zu vereinfachen, wurde ein *Slideshow-Menü* wie in Abbildung 4.4 angelegt. Dieses wird durch einen *Navigation Drawer* verkörpert, der durch Bedienen des *Nav-Icon* im linken Bereich der *Actionbar* geöffnet wird. Außerdem kann er, wie für diese Bedienelemente standardmäßig implementiert, durch horizontales Wischen vom linken Bildschirmrand aus aufgezogen werden. Die Breite dieses Bereiches wird durch den sogenannten *EdgeSlop* innerhalb der *Android ViewConfiguration* eines Gerätes beschrieben. Auf den beiden Testgeräten, dem Smartphone *LGE Nexus 5* und dem Tablet *Samsung SM-T700*, betrug er jeweils 36 Pixel.

Die Breite des *Navigation Drawer* wird von Android als die Breite des Bildschirms abzüglich der Höhe der *Actionbar* am oberen Bildschirmrand definiert.

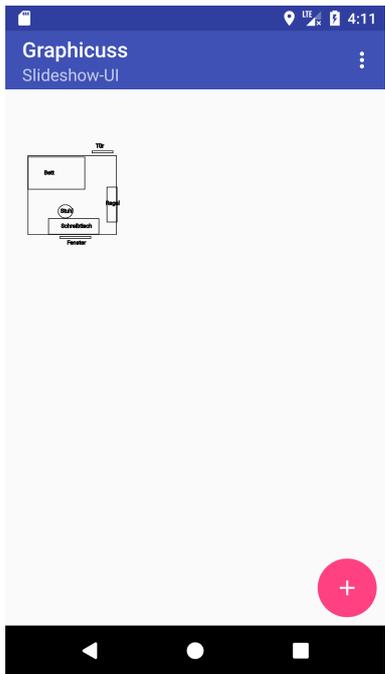


Abbildung 4.2.: Homescreen Smartphone

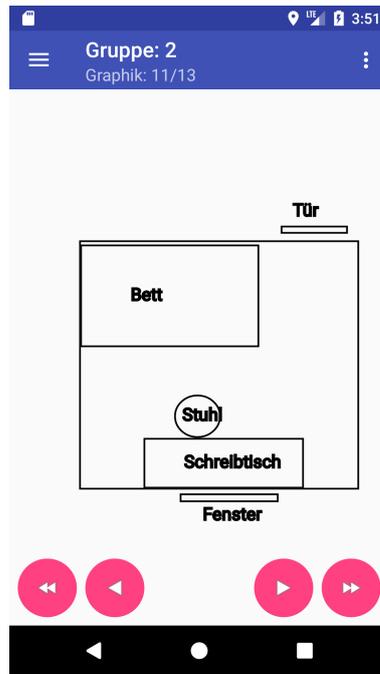


Abbildung 4.3.: Slideshow-UI Smartphone

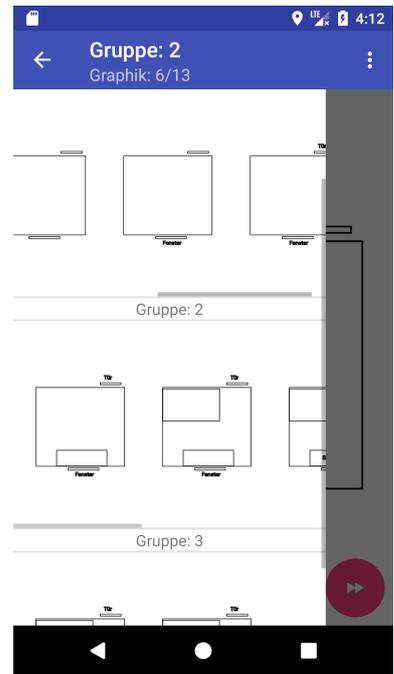


Abbildung 4.4.: Slideshow-Menü

Die Aufteilung des *Slideshow-Menüs* (fortfolgend auch Menü) ergab sich durch die Einschränkung der Größe. Ein Navigieren innerhalb des Menüs erfolgt mittels der korrespondierenden Wischrichtung. Der aktuelle Fortschritt innerhalb der Layouts wird jeweils durch grau unterlegte Scrollbars angezeigt. Alle Zeichenschritte einer Gruppe werden von jeweils einer *ImageView* dargestellt und sind in einem *LinearLayout* angeordnet, das wiederum in eine *HorizontalScrollView* eingefügt wurde. Eine senkrecht scrollbare *ListView* beinhaltet schließlich eine *TextView* mit dem Namen der Gruppe und die *HorizontalScrollViews* mit allen Zeichenschritten einer Gruppe. Diese *ListView* enthält also doppelt so viele Einträge, wie die Graphik Gruppen enthält.

Um einen Indiz dafür zu liefern, dass Scrollen möglich ist, wurden bewusst halbe Elemente von Zeichenschritten und Gruppen angezeigt. Die Anzahl für vertikal angezeigte Gruppen und auch die Anzahl von horizontal nebeneinander angezeigten Graphiken wurde auf etwa 2,5 festgelegt. Bei einer größeren Anzahl von Objekten würde der Inhalt der Zeichenschritte auf kleineren Bildschirmen zu undeutlich werden. Bei weniger Objekten wäre kein Mehrwert an Übersichtlichkeit gegeben.

Das Schließen des Slideshow-Menüs kann über den Home-Button am linken oberen Bildschirmrand, ein Tippen in den dunkel unterlegten Teil des Bildschirms am rechten Rand oder den Back-Button des Smartphones (hier links in der schwarzen Leiste am unteren Bildschirmrand) erwirkt werden.

Das Menü wurde im offenen Zustand für horizontale Wischbewegungen, die im Normalfall zum Zuschieben des Menüs dienen würden, gesperrt, um die horizontale Navigation innerhalb der Gruppen zu ermöglichen.

Durch Anklicken eines Items, also eines Zeichenschrittes, innerhalb des Menüs führt ebenfalls dazu, dass sich das Menü schließt und der *Viewpager* sich zum zugehörigen Zeichenschritt bewegt. Dies soll eine gezielte Navigation in großen Schritten ermöglichen.

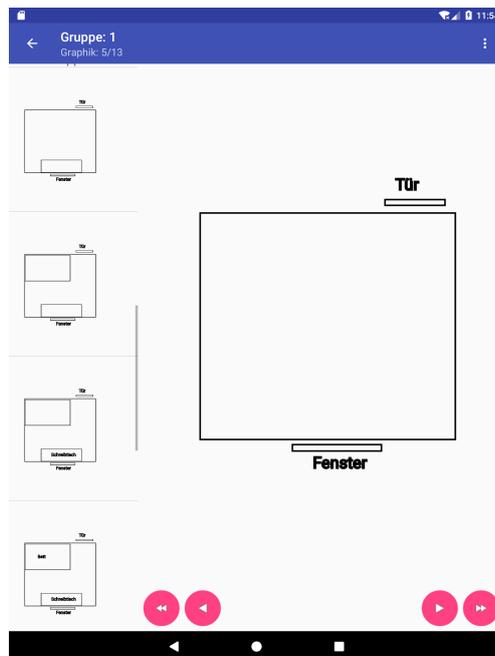


Abbildung 4.5.: Slideshow-UI Tablet mit Timeline

4.2.2. TABLET: SLIDESHOW-TIMELINE

Das Layout für Tablets sah eine Slideshow-Timeline vor (fortfolgend auch Timeline). Diese wurde am linken Bildschirmrand platziert. Der Viewpager und die Timeline befinden sich in einem horizontal ausgerichteten *LinearLayout*. Die Timeline kann außerdem ebenfalls als permanent geöffneter *Navigation Drawer* angesehen werden.

Die Breite der Timeline sollte ca. 25 % des Layouts einnehmen. Der Anteil wurde nach Einsetzen von verschiedenen Werten gewählt, um einen Kompromiss zwischen dem dominierenden *Viewpager* und einer gut erkennbaren Timeline zu finden.

Untereinander werden in einer *ListView* innerhalb der Timeline etwas mehr als 4 Graphiken dargestellt, die jeweils von einer *ImageView* repräsentiert werden. Die Anzahl ergab sich aus der festgelegten Breite der Timeline. Zusätzlich befinden sich innerhalb der *ListView* die Gruppennamen, die durch *TextViews* dargestellt werden.

Die *Timeline* kann über vertikale Wischgesten bewegt werden, d.h. sie ist scrollbar. Ein Indiz über den Fortschritt in der Historie wird über eine graue *Progressbar* am rechten Rand der *Timeline* bereitgestellt.

Durch Anklicken eines Elementes in der Timeline navigiert der *Viewpager* auf der rechten Seite zur zugehörigen Graphik.

4.3. VIDEOPLAYER-UI

Das Videoplayer-UI zeichnet sich durch seine Vielzahl an Bewegungen und seinem geringen Abstraktionslevel aus. Da der betrachtende Nutzer laut der Konzeption (Kap. 3.5) genau die Inhalte sehen kann, die der erstellende Nutzer gesehen hat, wurde für das Interface beispielhaft ein Screencast bei der Erstellung einer Graphik aufgenommen. Diese Vorgehensweise vereinfachte die prototypenartige Implementierung und ist für den Test der Workflows ausreichend. Bei einer Umsetzung dieses UIs für die Anwendung sollte dabei die Datenspeicherung optimiert werden.

Zur Ansicht der Videodatei wurde eine *VideoView* in das Layout eingebunden. Diese stellt den bearbeiteten Canvas dar. Die *VideoView* ist im Mittelpunkt von Abb. 4.6 zu sehen.

Zum Umgang mit der Videodatei wurde der *VideoView* ein *MediaController* zugewiesen. Der *MediaController* ist eine Ansicht, die die in der Konzeption 3.5 beschriebenen Bedienelemente des konzipierten *Mediaplayer* umsetzt. Er ist in Abb. 4.7 zu sehen.

Mediaplayer Unterschiede bestehen zwischen *MediaController* und *Mediaplayer* insofern, dass der *MediaController* nicht immer eingeblendet wird. Nach Tippen auf das Video wird dieser nach 3 Sekunden ausgeblendet, insofern keine Interaktion mit dem *MediaController* stattfindet. Das Timeout wurde von Android so festgelegt, es kann allerdings manuell verändert werden. Außerdem wurden die Buttons mit den Doppelpfeilen anders belegt. Der *Rewind-Button* erzeugt durch Antippen einen Sprung auf der Zeitachse von 5 Sekunden in Richtung Anfang des Videos. Der *Fastforward-Button* erzeugt durch Antippen einen Sprung von 15 Sekunden in Richtung Ende des Videos. Sie sind standardmäßig nicht für anderen Touchinput, wie längeres drücken, belegt worden.

Trotz der Abweichungen zum konzipierten *Mediaplayer* wurde der *MediaController* verwendet. Diese Entscheidung verfolgt die Absicht, das Design für mit Android vertrauten Nutzern so intuitiv wie möglich zu gestalten. Eine Abweichung von eventuell bereits bekannten Konzepten könnte die Nutzer irritieren. Außerdem kann, wenn die Bedienelemente nicht zu jedem Zeitpunkt zu sehen sind, mehr Fläche des Bildschirms zur Darstellung des Videos beansprucht werden.

Flag-Buttons Abweichend zum Konzept wurden in diesem Schritt auch die Flag-Buttons anders umgesetzt. Die Größe des *MediaControllers* bietet keine Fläche für zusätzliche Bedienelemente. Außerdem sollten dem Nutzer, als Abweichung zur Ansicht des Zeichentools

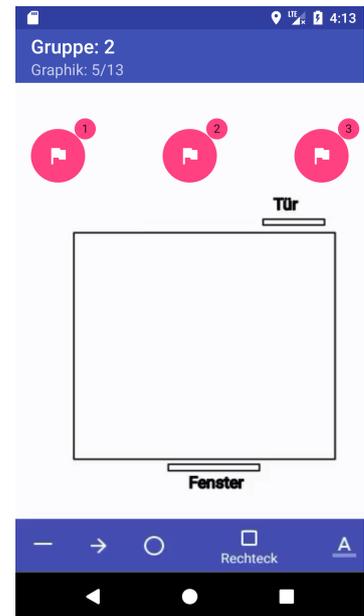


Abbildung 4.6.: Videoplayer-UI Smartphone

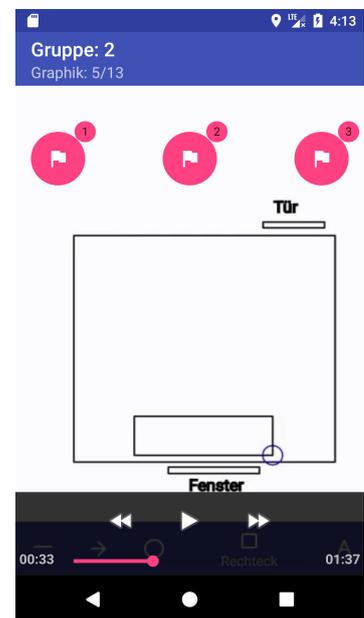


Abbildung 4.7.: Videoplayer-UI mit Mediaplayer

permanente Bedienelemente eingeblendet werden, um dem Nutzer signalisieren zu können, dass er sich im betrachtenden und nicht im zeichnenden Modus befindet. Des Weiteren kann so trotzdem eine schnellere Interaktion stattfinden.

Die Flag-Buttons wurden oberhalb des Videos horizontal nebeneinander platziert. Es existiert somit eine räumliche Trennung von Flag-Buttons und *MediaController* und folglich auch von neuen und bereits bekannten Bedienelementen. Außerdem befinden sich die Flag-Buttons so räumlich näher an den Informationen über das aktuelle Zeichenelement und die aktuelle Gruppe in der Actionbar am oberen Bildschirmrand. Die Anzeige ändert sich nach jedem abgeschlossenen Zeichenschritt.

Das Flaggen-Icon wurde beibehalten. Allerdings wurde eine Nummerierung hinzugefügt, welche die jeweiligen Gruppen symbolisieren. Durch Klicken springt das Video zum Ende der jeweiligen Gruppe.

Umgesetzt werden die Flag-Buttons von *Floating Action Buttons*. Diese sind im *colorAccent* der Androidanwendung gestaltet und setzen sich so deutlich vom Video ab.

5. EVALUATION

Die Evaluation stellt einen wesentlichen Teil des Usability-Engineerings dar. Grob wird in der Usability-Evaluation zwischen empirische und analytische Methoden unterschieden. Während im Rahmen von empirischer Evaluation die Meinung von Usability-Experten zur Bewertung herangezogen wird, basieren analytische Methoden auf der Befragung und Beobachtung von potentiellen Nutzern [SB16]. Da die Entwicklung des Konzepts bereits vorangeschritten ist und auf einem früheren Papierprototypen (Kap. 2.2.3) aufbaut, wurden im Rahmen dieser Arbeit größtenteils analytische Methoden zur formativen Evaluation umgesetzt.

5.1. EVALUATIONSZIELE

Im Rahmen der Evaluation sollen alle bisher getroffenen Entscheidungen bezüglich des Designs der User Interfaces hinterfragt und ggf. verbessert werden. Die Evaluation dient dazu, einen Überblick über die objektive und subjektive Einschätzung der Interfaces zu erlangen. Im Vordergrund der Evaluation stehen dabei die in der Konzeption festgelegten Layouts und die in der Implementierung verwendeten Objekttypen, sowie die verwendeten Bedienelemente.

Anschließend ist eine Liste der existierenden Layouts mit zugehörigen zu untersuchenden Bedienelementen zu sehen:

- Slideshow-UI für Smartphone
 - Slideshow-Buttons
 - Slideshow-Menü
- Slideshow-UI für Tablet
 - Slideshow-Buttons
 - Slideshow-Timeline
- Videoplayer-UI
 - Flag-Buttons
 - MediaPlayer bzw. MediaController

Aus den in der Entwurfsphase getroffenen Designentscheidungen ergaben sich folgende Evaluationsziele:

EZ1: Komplexität Der Prototyp soll einer Usability-Evaluation unterzogen werden. Deshalb muss festgestellt werden, wie einfach die Bedienung von potentiellen Nutzern empfunden wird. Dazu gehört das aktive Finden und Benutzen von implementierten Interaktionsmöglichkeiten.

EZ2: Wischgesten Alle Interfaces beziehen Wischgesten in die möglichen Interaktionen ein. Dabei soll untersucht werden, ob die Reaktion des Systems erwartungskonform abläuft. Zusätzlich soll eine konkretere Fragestellung für das Slideshow-Menü im Slideshow-UI für Smartphones untersucht werden. Durch horizontales Wischen von der linken Kante aus kann das Slideshow-Menü geöffnet werden. Dieser Bereich dient also nicht dazu, den *ViewPager* zu bewegen. Daher muss untersucht werden, ob es erwartungskonform wäre, an der rechten Kante ebenfalls einen Bereich zu implementieren, der den *ViewPager* nicht bewegt, um die Symmetrie zu wahren.

Außerdem soll evaluiert werden, ob die potentiellen Nutzer es als intuitiv empfinden, durch horizontales Wischen ein Menü zu öffnen.

EZ3: Notwendigkeit Für die Bedienelemente muss festgestellt werden, ob sie für die fertige Anwendung hilfreich und damit notwendig sind. Sollten mehrere Optionen zum Erreichen einer bestimmten Interaktion existieren, muss festgestellt werden, ob alle Optionen notwendig sind. Außerdem muss definiert werden, ob das Interaktionsspektrum vollständig ist.

EZ4: Optik Für die Bedienelemente muss festgestellt werden, ob sie bezüglich Größe, Icon und Farbe optimal gestaltet worden sind. Zudem muss für den Canvas und dessen Miniaturansichten festgestellt werden, ob Auflösung und Größe für den Verwendungszweck ausreichend gewählt worden sind.

EZ5: Positionierung Für die Bedienelemente muss festgestellt werden, ob sie richtig platziert worden sind.

EZ6: Übersichtlichkeit Für die Layouts muss festgestellt werden, ob sie übersichtlich aufgeteilt worden sind.

EZ7: Adaption Für die Layouts muss festgestellt werden, ob eine Unterscheidung des Designs von Smartphone und Tablet notwendig ist. Falls eine Unterscheidung erfolgte, sollten die Komponenten des Layouts, die beide Varianten enthalten, einheitlich gestaltet werden bzw. die gleichen Bedienmöglichkeiten aufweisen.

EZ8 Vergleich der Interfaces Ziel der beiden parallel erstellten Prototypen ist es, festzustellen, ob eines der Interfaces *Slideshow-UI* und *Videoplayer-UI* besser für den Nutzungskontext geeignet ist. Dazu sollten sie bezüglich Informationsgehalt, Komplexität und Gestaltung untersucht werden.

5.2. EVALUATIONSBEFRAGUNG

In diesem Kapitel sollen die Prototypen einer formativen Evaluation unterzogen werden. Das Ergebnis ist eine subjektive Bewertung durch die Befragung von Testpersonen.

5.2.1. VERSUCHSAUFBAU

Auswahl der Testpersonen Für ein realistisches/aussagekräftiges Ergebnis der Befragung, ist es maßgeblich, dass die Gruppe der Testpersonen alle potentiellen Endnutzer repräsentiert.

Insgesamt wurden 10 Nutzer befragt. Diese Zahl wurde gewählt, da bereits mit 5-6 Testpersonen im Durchschnitt 80 Prozent der Benutzbarkeitsprobleme auffindbar sind, aber diese Zahl eventuell stark abweichen kann. Um auch Feinheiten in den Systemfunktionalitäten evaluieren zu können, wurde eine bewusst größere Zahl an Testpersonen gewählt. Da bereits zu dem vorherigen Iterationsschritt von QUOTE eine Nutzerbefragung durchgeführt worden ist, wurde die Zahl der Testpersonen nicht noch höher angesetzt.

Da im Vorfeld nicht klar zu unterscheiden war, welche Nutzer *Graphicuss* explizit *nicht* nutzen, wurde eine heterogene Zusammensetzung der Tester hinsichtlich Geschlecht, Alter und Ausbildung erstellt. Bevorzugt sollten die Berufsgruppen Pädagogen und Informatiker vertreten sein.

Pädagogen sollten den Prototypen als intuitiv empfinden, um die Anwendung auch später in ihrem Berufsfeld nutzen zu wollen. Drei der Probanden sind im Bereich der Pädagogik tätig. Informatiker haben zusätzliches Hintergrundwissen und Erfahrungswerte, die gezielte Anregungen und differenzierte Testergebnisse liefern können. Eben so ausschlaggebend war die bewusste Auswahl an Probanden mit wenig Erfahrung bezüglich dem Umgang mit EDV. Die Hälfte der Testpersonen ist im Feld der Informatik berufstätig oder studiert innerhalb dieser Richtung. Die Einordnung der Befragten wird in Tabelle 5.1 ersichtlich.

Die Testpersonen hatten keine Vorkenntnisse zu dem zu testenden System. Sie waren zudem mit dem Konzept von *Graphicuss* nicht vertraut.

Alle Befragten nutzen seit mindestens drei Jahren ausschließlich Mobilgeräte mit dem Betriebssystem Android. Die Analysebefragung (Kap. 3.1) hat gezeigt, dass ein Einbeziehen von iOS-Nutzern das Ergebnis verfälschen könnte. Vor allem die Verwendung von Wischgesten werden auf Apple-Geräten anders umgesetzt. Zudem sind iOS-Nutzer nicht mit der Bedienung der für Android-Geräte üblichen Buttons. Diese umfassen Home-, Back- und Menü-Button und wurden in die Implementierung einbezogen.

Die Zusammenstellung der Probanden wird in Tabelle 5.1 ersichtlich:

Tabelle 5.1.: Einordnung der Befragten Evaluationsbefragung

Gruppe	Geschlecht	Altersgruppe	Berufsfeld	Anzahl Befragte
IT1	männlich	20 - 39	Informatik	2
IT2	weiblich	20 - 39	Informatik	1
IT3	männlich	40 - 60	Informatik	1
IT4	weiblich	40 - 60	Informatik	1
DI1	männlich	20 - 39	Anderes	1
DI2	weiblich	20 - 39	Anderes	2
DI3	männlich	40 - 60	Anderes	1
DI4	weiblich	40 - 60	Anderes	1

Untersuchungsumgebung Das Umfeld für den Test durch die Probanden sollte so realistisch wie möglich gewählt werden. Aufgründdessen erfolgten die Tests in einer Bibliothek, in einem Café, in einem Firmengebäude, sowie in den privaten Räumlichkeiten der Befragten. Diese Standorte wurden gewählt, da sie häufig von Studierenden zum Lernen besucht werden und verschiedene Arten und Lautstärken von Umgebungsgeräuschen vorhanden sind.

Es wurde im Vorfeld davon ausgegangen, dass die Anwendungsszenarien von den Nutzern im Sitzen durchgeführt werden. Die Annahme basiert auf der nötigen Konzentration, die der potentielle Nutzer für den Umgang mit dem System aufbringen muss, auch obwohl die Anwendung auf einem mobilen Gerät ausgeführt wird.

Auswahl des Beispiels Das gezeigte Beispiel sollte für alle Probanden gleich komplex gewählt werden. Deshalb wurde eine Graphik einer Raumplanung, die die Platzierung von Möbeln innerhalb eines Raumes veranschaulicht gewählt. Das Beispiel besaß drei Gruppen, die den Grundriss, die fest platzierten Möbel und die diskutabel platzierten Möbel jeweils mit deren Beschriftung beinhalteten.

In der zukünftigen Anwendung sollen Undo-Schritte möglich sein und auch in der Erstellungshistorie ersichtlich werden. Ein Einbeziehen dieser Problemstellung wurde in das Beispiel beabsichtigt nicht eingebaut. Mittels dieser Evaluation soll vor allem der Workflow der Anwendung und das Verständnis der Bedienoberfläche durch potentielle Nutzer untersucht werden. Eine zu komplexe Gestaltung des Beispiels könnte das Ergebnis verfälschen.

5.2.2. ERHEBUNGSMETHODEN

Screencast Um alle Aktionen der Testpersonen im Anschluss rekapitulieren zu können, wird während des Tests bei allen Probanden ein Screencast erstellt, d.h. der Bildschirminhalt des Testgerätes wird aufgezeichnet.

Außerdem sollte der genauen Touch-Input nachvollzogen werden, weshalb das visuelle Feedback und die Zeigerposition in den Android-Entwickleroptionen aktiviert wurden.

Audiomitschnitt Während der gesamten Befragung erfolgt zudem ein Audiomitschnitt, um eventuelle auftretende Fragen oder Kommentare der Probanden zu dokumentieren. Primär sollten dabei Begründungen für die Bewertungen innerhalb des Fragebogens durch die Probanden dokumentiert werden.

Methode des *lauten Denkens* Explizit sollte bei der Bearbeitung die Methode des *lauten Denkens* angewendet werden, d.h. die Tester sollten aktiv alle Handlungen, Erwartungen und Empfindungen laut aussprechen. Diese Durchführung kann allerdings zu einer längeren Bearbeitungszeit und weniger auftretenden Fehlern führen, da der Proband aktiv alle seine Handlungen hinterfragt.

AB-Test Alle Probanden testeten **beide** User Interfaces. Dabei wurde der sogenannte *AB-Test* angewendet, d.h. bei der Hälfte der Probanden wurde die Befragung mit der Vorstellung des *Slideshow-UI* begonnen, bei der anderen Hälfte mit dem *Videoplayer-UI*. Durch die Methode sollen Beeinflussungen durch die Reihenfolge der Vorstellung der UIs unterbunden werden.

Anregungen Falls die Testpersonen eher passiv auftraten und keine weitere Interaktion durchgeführt wurden, obwohl fundamentale Bestandteile des jeweiligen UIs noch nicht verwendet worden waren, wurden Anregungen in Form von kleinen Fragen durch den Testleiter gegeben. Diese bestanden bspw. darin, nach weiteren ungenutzten Navigationsmöglichkeiten zu suchen.

Fundamental heißt in diesem Zusammenhang, dass sie einen wichtigen Teil der Befragung darstellen und eine Interaktion für die Beantwortung der Fragebögen notwendig sind.

5.2.3. DURCHFÜHRUNG

Alle Probanden erhalten vor der Evaluation eine schriftliche Einführung in das Themengebiet (Anh. A.3.1). Damit solle ein einheitlicher Wissensstand/Vorkenntnisse bei allen Befragten generiert werden.

Abhängig von der Unterteilung für den AB-Test, wird mit einem vordefinierten UI begonnen. Beide UIs werden auf einem Smartphone und auf einem Tablet vorgestellt. Die Reihenfolge der Testgeräte wird ebenfalls durch einen AB-Test variiert.

Im Anschluss an die Benutzung eines UIs erfolgt die Ausgabe der zugehörigen Fragebögen. Die Nutzer können während sie diese bearbeiten weiterhin auf die Mobilgeräte zurückgreifen.

Es werden insgesamt vier Fragebögen ausgehändigt (Anhang A.3). Insgesamt beinhalten diese 72 Items.

Der Aufbau der Fragebögen ergibt sich unmittelbar aus den Evaluationszielen 5.1. Dabei werden die jeweiligen Fragebögen in die zu untersuchenden Bedienelemente unterteilt. Die Fragebögen beinhalten negierte Fragen, um die Aufmerksamkeit des Lesenden zu überprüfen. Sie beinhalten ausschließlich Multiple-Choice-Fragen. Bei einer dieser Fragen wurde zudem nach einem konkreten Wert gefragt. Über die Hälfte der Fragen besteht darin, Kriterien der UIs mittels einer Likert-Skala von eins bis sieben zu bewerten. Diese Skala wurde gewählt, da sie bspw. beim UEQ verwendet wird [SB16]. Außerdem besteht bei einer ungeraden Skala die Möglichkeit einer neutralen Antwort. Eine kleinere Skala wird nicht gewählt, um den Nutzern eine größere, aber nicht zu große Freiheit bezüglich der Genauigkeit der Bewertung zu geben.

Da für Slideshow-UI unterschiedliche Layouts für Smartphone und Tablet vorliegen, existieren getrennte Fragebögen für beide Gerätearten.

Es werden relativ viele Fragen gestellt, um explizit Schwächen der UIs ergründen zu können. Den Nutzern werden folgende Fragebögen vorgelegt:

- Slideshow-UI für Smartphone (Anh. A.3.2)
- Slideshow-UI für Tablet (Anh. A.3.3)
- Videoplayer-UI (Anh. A.3.4)
- Vergleich der UIs (Anh. A.3.5)

Die Items der Fragebögen werden entsprechend nummeriert. Der erste Buchstabe repräsentiert das untersuchte Interface (S für Slideshow-UI oder V für Videoplayer-UI). Der zweite Buchstabe definiert die Art der Frage, wobei A für allgemeine Frage steht. Ist der zweite Buchstabe ein D, wurde eine Detailfrage gestellt. Diese werden auf einer Skala von 1 bis 7 bewertet.

Nach dem aktiven Test werden die Fragebögen vom Testleiter evaluiert. Dabei wird gezielt nach Mängeln an der Anwendung gesucht.

Abschließend folgt ein auswertendes Evaluationsgespräch mit dem Probanden. Bei negativen Bewertungen wird nach Gründen und Verbesserungsvorschlägen gefragt, um konkrete Problemstellungen herausfinden zu können. Das Evaluationsgespräch wird ebenfalls auditiv aufgezeichnet.

Die Befragungen dauern etwa 60 Minuten pro Testperson.

5.3. AUSWERTUNG DER EVALUATIONSBEFragung

Bei der Auswertung ist zu Beachten, dass manche Items bewusst negiert formuliert worden sind, um die Aufmerksamkeit und die Genauigkeit der Nutzer zu kontrollieren.

Die Negation wurde bei der Auswertung wieder entfernt, d.h. die Items sind alle positiv formuliert. Die Bewertungen wurde entsprechend invertiert. Das trifft auf die Fragen SD12, SD15, SD23, SD24, VD10 und VD12 zu.

Die Werte wurden zudem auf eine Nachkommastelle gerundet.

Sieben entspricht *trifft voll zu/ Sehr gut.*

Eins entspricht *trifft gar nicht zu/ sehr schlecht.*

Bei den Bewertungen von 1 bis 7 werden jeweils die Mittelwerte gebildet.

Außerdem wird ein Grenzwert anhand der vorliegenden Ergebnisse definiert. Sollte der erzielte Wert unter 6,0 bzw. etwa 85% der maximal möglichen Bewertung (7,0) erreicht worden sein, wurde die zugehörige Zelle grau markiert. Dieser Grenzwert wurde definiert, da dieser eine minimale Schrittgröße vom Maximalwert abweicht. Somit kann eine Wert von 6,0 oder besser nur bei kleineren Kritikpunkten oder einzelnen, stark abweichenden Wertungen erreicht werden und symbolisiert somit größtenteils Zustimmung. Wurde der Grenzwert nicht erreicht, ist eine Entwurfsentscheidung zur Verbesserung dieses Items zu treffen.

5.3.1. SLIDESHOW-UI FÜR SMARTPHONE

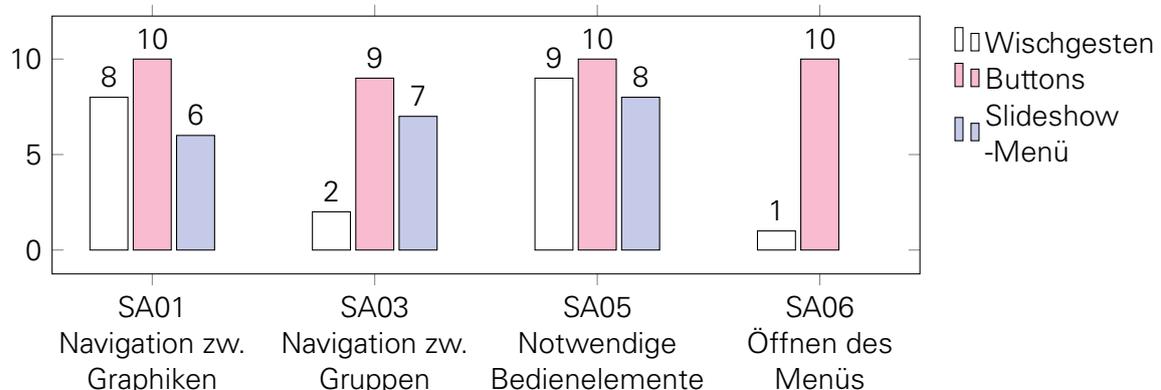


Abbildung 5.1.: Antworten Slideshow-UI Smartphone Allgemein MF

Aus Abb. 5.1 und den Screencasts ergab sich, dass das meistbenutzte Bedienelement für die Grundelemente der Navigation mithilfe der Slideshow-Buttons umgesetzt wurde.

Zwei der Probanden versuchten als erste Interaktion im Slideshow-UI für Smartphone oder Tablet, mit Wischgesten zu navigieren. Zwei weitere Testpersonen erkannten diese Möglichkeit der Interaktion nach Betätigung der Slideshow-Buttons, da der *Viewpager* dabei horizontal gescrollt wurde. Die weiteren vier Testpersonen, die Wischgesten verwendeten, erkannten diese Möglichkeit der Interaktion bei dem Versuch, die Größe der Ansicht des Canvas zu verändern, also zu zoomen.

Alle Bedienelemente wurden von einem Großteil der Probanden genutzt und als notwendig wahrgenommen (EZ3).

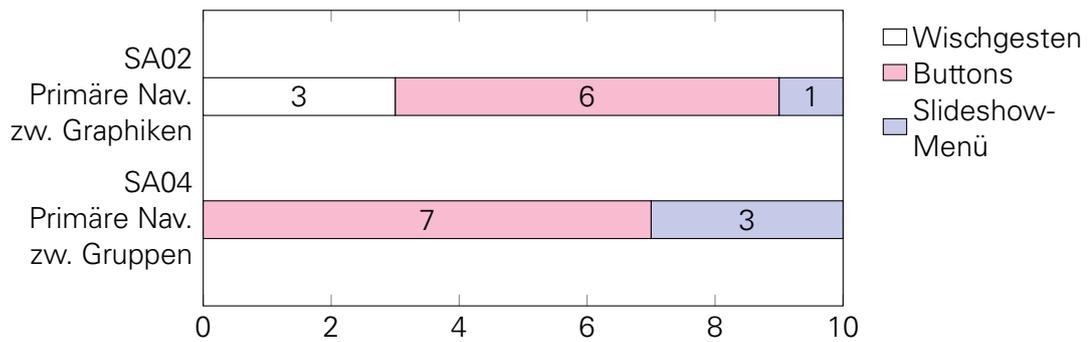


Abbildung 5.2.: Antworten Slideshow-UI Smartphone Allgemein EF

Aus Abb. 5.2 wird ebenfalls ersichtlich, dass die Slideshow-Buttons die am häufigsten genutzten Bedienelemente zur Navigation sowohl zwischen benachbarten Graphiken, als auch zwischen benachbarten Gruppen repräsentiert. Zurückzuführen ist diese Feststellung auf die direkte Präsenz der Slideshow-Buttons beim Öffnen des Interfaces. Wischgesten werden eher aus Erfahrungswerten angewendet. Das Slideshow-Menü muss zunächst durch eine Interaktion geöffnet werden, bevor es zur Navigation genutzt werden kann.

Tabelle 5.2.: Bewertung Slideshow-Buttons für Smartphone

Item	Beschreibung	Wert
SD01	Positionierung	6,4
SD02	Notwendigkeit	7,0
SD03	Eindeutigkeit	5,5
SD04	Optik (Farbe, Größe, Icon)	5,9

Die Bewertung der **Slideshow-Buttons** in Tab.5.2 zeigt auf, dass die Buttons notwendig (EZ3) und erwartungskonform platziert (EZ5) wurden.

Allerdings wurde deutlich, dass die Testpersonen nicht im Vorfeld Rückschlüsse auf die Funktion aller Buttons ziehen konnten. Dieses Problem beschränkte sich auf den Fastforward- und den Rewind-Button. Vier der Probanden erkannten deren wirkliche Funktionalität nach mehrmaligem Bedienen. Die restlichen 6 Versuchspersonen erkannten keinen Zusammenhang zwischen diesen Buttons und der aktuell dargestellten Gruppe. 8 der Probanden gaben an, dass sie vermutet hätten, mittels dieser Buttons zum ersten und letzten Zeichenschritt der gesamten Graphik navigieren zu können.

Zurückzuführen ist dieses Problem auf die Platzierung der Informationen über den aktuell angezeigten Zeichenschritt und die zugehörige Gruppe in der *Actionbar*. Die Slideshow-Buttons und die *Actionbar* sind räumlich zu weit auseinander platziert, um einen direkt Zusammenhang wahrnehmen zu können.

9 der Testpersonen vertraten außerdem die Meinung, dass eine *Actionbar* erfahrungsgemäß ausschließlich statische Informationen beinhaltet. Aufgründessen wird der dort dargestellte Text seltener betrachtet.

Toolbar Um die Eindeutigkeit (SD03) der Slideshow-Buttons zu verbessern, wurde beschlossen, am unteren Bildschirmrand eine *Toolbar* einzufügen, die die bereits implementierten Buttons, sowie Informationen über die momentan angezeigte Graphik enthält. Um den Zusammenhang zu verdeutlichen und eine direkte Beeinflussung zu suggerieren, wird die *Toolbar* eine einheitliche Farbgebung erhalten.

Die Bewertung der Optik (SD04) ist darauf zurückzuführen, dass manche Testpersonen durch die Signalfarbe der Buttons vom eigentlich zentralen Canvas abgelenkt wurden oder dass die Farbe ihnen persönlich nicht gefiel. (EZ4)

Um die Optik der Slideshow-Buttons zu verbessern, sollten sie nicht im *colorAccent* der Anwendung gestaltet werden. Trotzdem sollte sie sich farblich vom Canvas abheben.

Tabelle 5.3.: Bewertung Slideshow-Menü für Smartphone

Item	Beschreibung	Wert
SD05	Positionierung	6,5
SD06	Notwendigkeit	6,4
SD07	Erkennbarkeit der Möglichkeit zur Navigation	5,0
SD08	Übersichtlichkeit	5,3
SD09	Erkennbarkeit der Graphiken	5,0
SD10	Offensichtlichkeit Existenz	4,6
SD11	Leicht zu Öffnen	7,0
SD12	Leicht zu Schließen	6,4
SD13	Erwartungskonform, Menü mit horizontalem Wischen zu öffnen	3,6

Das **Slideshow-Menü** wurde in Tabelle 5.3 bewertet. Fast alle Probanden hielten das Menü zwar für richtig platziert (EZ5) und notwendig (EZ3), allerdings stellte das größte Problem die *Offensichtlichkeit der Existenz* (SD10) dar.

In Diagramm 5.1 ist zu sehen, dass nur einer der Befragten das Menü über Wischgesten öffnete. Nur drei der Befragten öffneten zudem das Slideshow-Menü ohne Anregung des Testleiters (EZ2). Die positive Bewertung des Items SD11 ist darauf zurückzuführen, dass der *Navigation-Button* als leicht zu bedienen empfunden wurde.

Zurückzuführen ist die Bewertung auf die Tatsache, dass Wischgesten nicht zum Öffnen verwendet wurden, da die Probanden es nicht für erwartungskonform hielten, ein Menü auf diese Weise zu öffnen (Tab. 5.3, SD13).

Positiv zu erkennen war die Tatsache, dass die Probanden das Schließen des Slideshow-Menüs als leicht empfanden. Um im Menü horizontales Scrollen zu ermöglichen, wurde die Wischgesten zum Schließen des Slideshow-Menüs gesperrt. Diese Tatsache bemerkte nur einer der Probanden, da Touchinput rechts außerhalb des Menüs ebenfalls zum Schließen des Menüs führt.

Auf Nachfrage durch den Testleiter wurden alle drei Möglichkeiten zum Schließen des Menüs als erwartungskonform eingestuft. Das Menü kann über ein Berühren des *Viewpagers*, Klicken des *Navigation-Buttons* oder Klicken des *Back-Buttons* erreicht werden.

Drag-Element Die Offensichtlichkeit der Existenz soll durch ein Drag-Element an der linken Seite des Bildschirms behoben werden, das das Slideshow-Menü durch Aufziehen oder Anklicken öffnet.

Zudem müssen Erkennbarkeit der Möglichkeit zur Navigation (SD07), Übersichtlichkeit (SD08) und Erkennbarkeit der Graphiken (SD09) verbessert werden (EZ4, EZ6).

Divider Um anzudeuten, dass die im *Slideshow-Menü* existierenden Graphiken angeklickt werden können, sollen Trennlinien, sogenannte *Divider*, zwischen den einzelnen Bildern dargestellt werden. Diese trennen die einzelnen Zeichenschritte räumlich deutlich voneinander

ab und gestalten somit das Layout übersichtlicher.

Übersichtlichkeit des Menüs Zusätzlich können SD08 und SD09 verbessert werden, indem die Elemente des Slideshow-Menüs ausgeklappt werden können. So wären beim ersten Öffnen ausschließlich die Gruppennamen bzw. -nummern zu sehen und durch Anklicken dieser würden sich zusätzlich die horizontal angeordneten Elemente der Gruppe im Menü öffnen.

Tabelle 5.4.: Bewertung Slideshow-UI für Smartphone Allgemein

Item	Beschreibung	Wert
SD14	Erwartungskonforme Reaktion auf Wischgesten	5,8
SD15	Einfachheit, bestimmten Zeichenschritte zu finden	6,5
SD16	Berechenbarkeit	5,4
SD17	Eignung GUI für Smartphones	6,5

Die **Allgemeine Bewertung** (Tab. 5.4) zeigte auf, dass die Probanden das Interface für Smartphones geeignet und als einfach zu bedienen empfanden (EZ1).

Die Bewertung von Fragestellung SD14 und SD16 ist auf die Bewertung von SD13 zurückzuführen. Für ein Großteil der Probanden war es nicht intuitiv, ein Menü durch Wischgesten zu öffnen, was die Berechenbarkeit negativ beeinflusst.

Durch das zusätzliche Bedienelement des Slideshow-Menüs sollen ebenfalls die beiden Items unterhalb des Grenzwertes aus Tabelle 5.4 positiv beeinflusst werden.

Eine zusätzliche Fragestellung, die sich während der ersten Befragungen ergeben hatte, bestand darin, wie die Testpersonen das Slideshow-Menü tendenziell am häufigsten schließen würden. Dabei gaben vier Testpersonen an, dass sie zunächst den Navigation-Button verwenden würden. Fünf weitere würden einen Touchinput im Bereich rechts außerhalb des Menüs verwenden, also klicken oder horizontal von rechts aus wischen. Einer der Befragten würde den Back-Button des Mobilgerätes verwenden. Diese Fragestellung gab Aufschluss darüber, dass die Sperrung von horizontalen Wischgesten innerhalb des Menüs zum Schließen desselben auf den erwartungskonformen Umgang mit dem UI keinen negativen Einfluss hat.

5.3.2. SLIDESHOW-UI FÜR TABLET

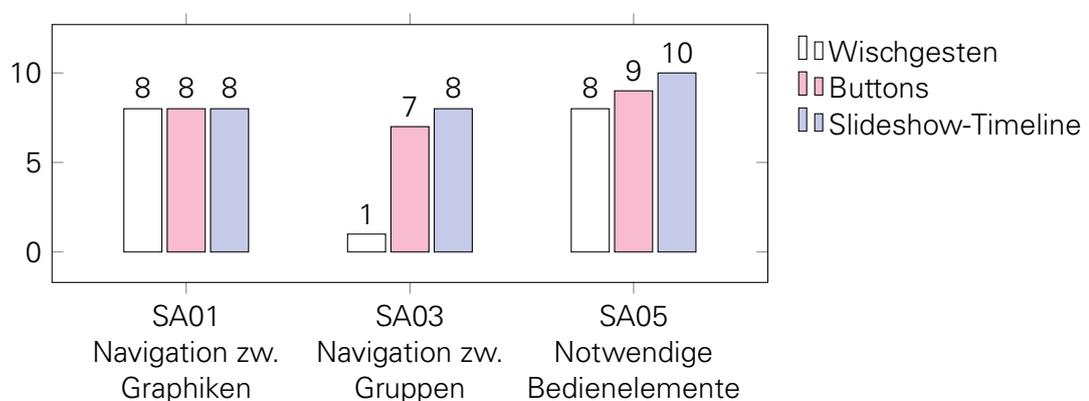


Abbildung 5.3.: Antworten Slideshow-UI Tablet MF

Diagramm 5.3 verdeutlicht, dass die Slideshow-Timeline ein wesentliches Element zur Navigation im Slideshow-UI für Tablets darstellt, im Gegensatz zum Slideshow-Menü im Interface für Smartphones. Diese Tatsache ist auf die ständige Präsenz der Timeline zurückzuführen.

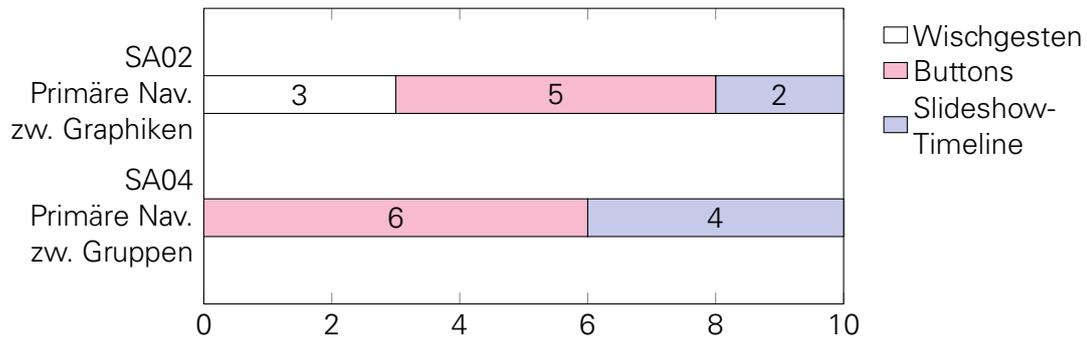


Abbildung 5.4.: Antworten Slideshow-UI Tablet EF

Aus den Diagrammen 5.3 und 5.4 ist außerdem erkennbar, dass alle Bedienmöglichkeiten notwendig sind, da sie für notwendig empfunden wurden (SA05) und genutzt worden sind (EZ3). Dabei ist erkennbar, dass Wischgesten eine weniger wichtige Interaktionsmöglichkeit darstellten. Durch die ständige Präsenz von Slideshow-Buttons und Slideshow-Timeline wurden den Testpersonen vermutlich bereits zwei ausreichende Bedienmöglichkeiten präsentiert. Das Testverhalten für die Verwendung von Wischgesten wurde in 5.3.1 unabhängig vom zuerst getesteten Mobilgerät zusammengefasst.

Tabelle 5.5.: Bewertung Slideshow-Buttons für Tablet

Item	Beschreibung	Wert
SD01	Positionierung	4,1
SD02	Notwendigkeit	7,0
SD03	Eindeutigkeit	6,3
SD04	Optik (Farbe, Größe, Icon)	4,5

Die Bewertung der Slideshow-Buttons (Tab.5.5) zeigt auf, dass die Positionierung (SD01) (EZ5) und die Optik (SD04) (EZ4) den Grenzwert nicht erreichten.

An der Positionierung (SD01) wurde bemängelt, dass die Slideshow-Buttons zu weit am unteren Bildschirmrand und deshalb zu weit entfernt vom Canvas platziert worden sind. Die Optik wurde vergleichsweise schlecht bewertet, da die Buttons von den Probanden als zu klein für Tablets empfunden worden sind.

Toolbar Sowohl die Positionierung, als auch die Optik waren konform zum Interface für Smartphones gestaltet. Die Standardfarbe und die Standardgröße der *Floating Action Buttons* wurde bis dahin nicht verändert.

Um ein konformes Auftreten weiterhin gewährleisten zu können, soll auch die in 5.3.1 bereits beschlossene Entwurfsentscheidung einer *Toolbar* für Tablets entsprechend umgesetzt werden. Die Größe der Buttons muss im UI für Tablets angepasst werden.

Die Bewertung der Slideshow-Timeline (Tab.5.6) verdeutlichte eine Zustimmung zu einem Großteil der Entwurfsentscheidungen.

Der größte Kritikpunkt bezüglich der Timeline stellte die mangelnde Übersichtlichkeit dar. Die Probanden erwähnten, dass vor allem bei einer Vielzahl von Zeichenschritten ein vertikales

Tabelle 5.6.: Bewertung Slideshow-Timeline für Tablet

Item	Beschreibung	Wert
SD19	Notwendigkeit	7,0
SD20	Offensichtlichkeit der Navigationsmöglichkeit	6,5
SD21	Übersichtlichkeit	4,8
SD22	Erkennbarkeit der Graphiken	6,5
SD23	Einfachheit Bedienung	6,9

Scrollen umständlich sein kann. Vor allem bei schnellem Scrollen können eventuell die Gruppennamen bzw. -nummern überlesen werden.

Übersichtlichkeit der Timeline Die Übersichtlichkeit soll, wie im UI für Smartphones, durch einklappbare Gruppen verbessert werden (EZ6).

Außerdem sollen Gruppenbezeichnungen hervorgehoben werden, indem die Schrift farbig und größer gestaltet wird (EZ6).

Präsenz der Timeline Alle Probanden, die zunächst das Slideshow-UI auf dem Smartphone untersucht haben, versuchten bei dem Interface für Tablets, die Slideshow-Timeline auszublenken. Die Testpersonen hielten diese Unterscheidung für inkonsistent, da das Slideshow-Menü geschlossen werden kann (EZ1, EZ7). Da das UI für Smartphones kein ständig präsent Slideshow-Menü erlaubt, wurde beschlossen, die Slideshow-Timeline um die Möglichkeit zu erweitern, ausgeblendet zu werden. Geöffnet und geschlossen werden soll die Timeline wie im Interface für Smartphones. Im Detail heißt das, dass Öffnen der Slideshow-Timeline zum einen über horizontales Wischen vom linken Bildschirmrand aus innerhalb des *EdgeSlop* ermöglicht werden soll. Zum anderen soll die Timeline über einen Navigation-Button am linken Rand der *Actionbar* und über ein Drag-Element bzw. einen Button an der linken Seite des Bildschirms geöffnet werden. Geschlossen werden kann die Timeline über horizontales Wischen, den Navigation-Button oder den Back-Button des Mobilgerätes. Die Breite der Timeline soll dabei nicht verändert werden, da sie den Anforderungen der Nutzer gerecht wurde. Während die Timeline geöffnet ist, soll weiterhin eine Bedienung des *Viewpagers* möglich sein.

Breite der Timeline Während der Befragung ist vier der Probanden aufgefallen, dass die dargestellten Zeichenschritte innerhalb der Slideshow-Timeline schmaler sind als die Timeline. Beim Anklicken der Elemente in der Timeline werden diese hellgrau hinterlegt.

Da die Erkennbarkeit der Graphiken (SD22) positiv bewertet worden ist und außerdem 8 der Testpersonen nicht befürworten würden, die Graphiken kleiner darzustellen, sollte die Breite der Timeline auf die Breite der dargestellten Graphiken angepasst werden.

Tabelle 5.7.: Bewertung Slideshow-UI für Tablet Allgemein

Item	Beschreibung	Wert
SD14	Erwartungskonforme Wischgesten	6,5
SD15	Einfachheit, bestimmten Zeichenschritt zu finden	5,3
SD16	Berechenbarkeit	6,6
SD24	Eignung GUI für Tablets	6,5
SD25	Notwendigkeit Interface für Tablets	6,3

In Tabelle 5.7 ist zu erkennen, dass das Interface für Tablets sehr gut abgeschnitten hat. Es wurde deutlich, dass eine Trennung der Slideshow-UIs für Smartphones und Tablets notwendig war, um die unterschiedlichen Bildschirmgröße optimal nutzen zu können (SD25, EZ7).

Die Einfachheit einen bestimmten Zeichenschritt zu finden (SD15) soll ebenfalls durch die Verwendung von **Zusammengefassten Gruppen** erreicht werden (EZ1). Fünf der Testpersonen machten unabhängig voneinander diesen Vorschlag für das Interface. Die andere Hälfte der Testpersonen gab auf Nachfrage des Testleiters an, dass die ausklappbaren Gruppen die Einfachheit (SD15), sowie die Übersichtlichkeit der Timeline (SD21) deutlich verbessern würde.

Zwei der Befragten gaben an, dass sie die Variante des Slideshow-UI für Tablets für beide Arten von Mobilgeräten bevorzugen würden. Einer der Befragten würde als alleiniges Layout die Variante für Smartphones unterstützen.

Somit wird eine Unterscheidung im Layout für Tablets und Smartphones ausdrücklich befürwortet (EZ7).

5.3.3. VIDEOPLAYER-UI

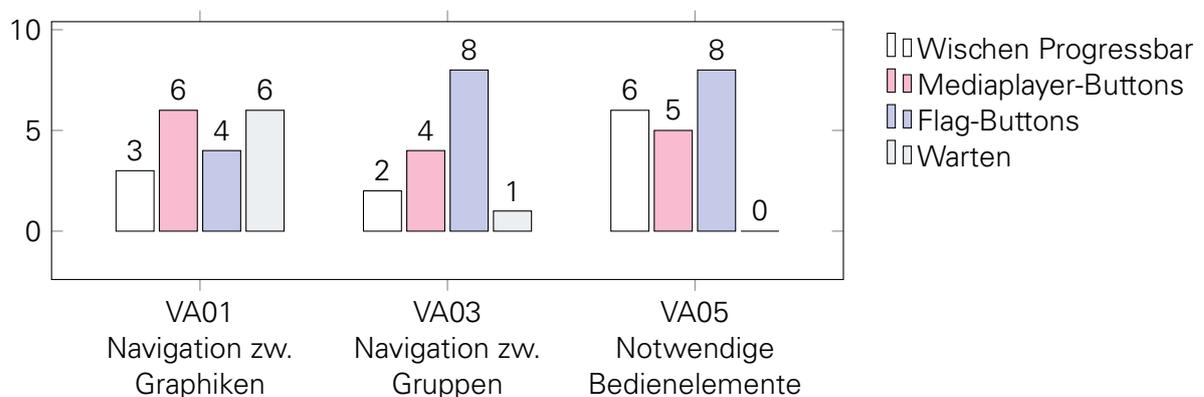


Abbildung 5.5.: Antworten Videoplayer-UI MF

In Tabelle 5.5 für das Videoplayer-UI ist zu sehen, dass alle Bedienelemente als notwendig empfunden wurden (EZ3).

Nur die Hälfte der Testpersonen empfand die Mediaplayer-Buttons als notwendig. Zurückzuführen ist diese Bewertung auf die häufigere Benutzung der *Progressbar*, die in den meisten bekannten Videoplayern umgesetzt wurde, während die Mediaplayer-Buttons weniger einheitliche Funktionen implementieren.

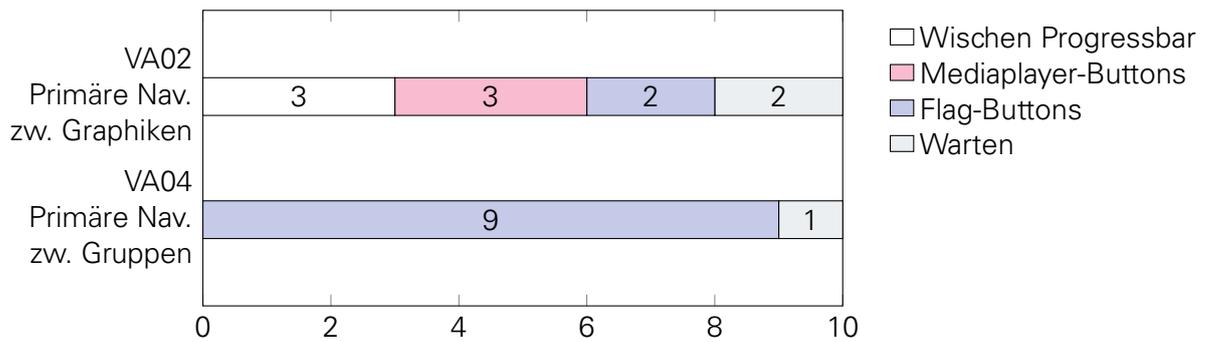


Abbildung 5.6.: Antworten Videoplayer-UI EF

In Tabelle 5.6 wird ebenfalls die Notwendigkeit aller Bedienelemente ersichtlich (EZ3). Zudem ist erkennbar, dass die Flag-Buttons das primäre Element zur Navigation zwischen benachbarten Gruppen darstellt.

Tabelle 5.8.: Bewertung Flag-Buttons Videoplayer-UI

Item	Beschreibung	Wert
VD01	Positionierung	4,6
VD02	Notwendigkeit	6,6
VD03	Selbsterklärend	4,4
VD04	Optik (Größe, Farbe, Icon)	4,5

Tabelle 5.8 verdeutlicht, dass die Positionierung (VD01) der Flag-Buttons nicht ideal gestaltet worden ist (EZ5). Dabei gaben 7 der Testpersonen an, dass sie eine Trennung von Mediaplayer und Flag-Buttons bevorzugen, um eine schnellere Navigationsmöglichkeit zu unterstützen, da der Mediaplayer nicht immer eingeblendet wird (VD07).

Die Bewertung der Positionierung ist durch die Platzierung der Buttons auf dem Video entstanden. Da innerhalb des gezeigten Videos auch Texteingaben angezeigt worden sind, wurden diese oberhalb der Flag-Buttons dargestellt. Somit existierte keine feste Trennung der Buttons und des Videos.

Außerdem ist zu erkennen, dass die Funktion der Flag-Buttons nicht selbsterklärend (VD03) und die Optik (VD04) nicht optimal gewählt worden ist. Zum einen wurde von vier der Probanden angemerkt, dass sie das Icon einer Flagge nicht als geeignet empfinden. Außerdem lenkt die Gestaltung der *Floating Action Buttons* in einer Signalfarbe, dem *colorAccent* der Anwendung, vom Video ab.

Toolbar Wie im Slideshow-UI traten Probleme bezüglich der Verständlichkeit der Buttons auf, die Sprünge im Zeichenverlauf unter Einbezug der festgelegten Gruppen ermöglichen. Um die Flag-Buttons selbsterklärend zu gestalten, sollte sie daher mit den Informationen über die aktuelle Gruppe und den aktuell dargestellten Zeichenschritt eine optische Einheit bilden. Dazu könnte eine *Toolbar* unterhalb der *Actionbar* am oberen Bildschirmrand platziert werden. Da die Texteingabe ohnehin als weniger relevant empfunden worden ist (VD10), könnte diese dementsprechend ausgeblendet werden. Um dahingehend auch die Optik zu verbessern, sollte die *Toolbar* in einer weniger grellen Farbgebung gestaltet werden.

Tabelle 5.9.: Bewertung Mediaplayer Videoplayer-UI

Item	Beschreibung	Wert
VD05	Positionierung	7,0
VD06	Erkennbarkeit als Navigationselement	6,1
VD07	Ständige Präsenz	3,5

Eine ständige Präsenz des Mediaplayers wurde mehrheitlich abgelehnt (VD07). Die Bewertung ergab sich, da auch die Probanden, die es bevorzugen, den Mediaplayer nach einem bestimmten Zeitintervall auszublenden, ein längeres Zeitintervall für das Einblenden des Mediaplayers unterstützen würden. Die kurze Dauer des Einblendens wurde als störend empfunden. Trotzdem soll vor allem auf dem Layout für Smartphones mehr Fläche zur Darstellung des Canvas bereitgestellt werden.

Die Erkennbarkeit der Navigationselemente (VD06) erhielt Abzug in der Bewertung, da vier der Testpersonen nicht bewusst war, dass ein Mediaplayer existiert.

Sechs der Probanden hätten erwartet, dass man mittels des Fastforward- und des Rewind-Buttons spulen könnte.

Die Frage VA06 nach der gewünschten Belegung dieser Buttons ergab außerdem, dass sieben der Testpersonen einen Sprung über eine feste Anzahl an Zeichenschritten erwartet hätten. Sechs Personen gaben an, dass sie es für sinnvoll halten würden, einen Zeichenschritt vor- bzw. zurückzuspulen. Eine Testperson wählte fünf Zeichenschritte. Drei Testpersonen wählten einen Sprung von einem festen Zeitintervall über fünf Sekunden.

Tabelle 5.10.: Bewertung Videoplayer-UI Allgemein

Item	Beschreibung	Wert
VD08	Erkennbarkeit der Inhalte	5,9
VD09	Zeitliche Informationen über Erstellungshistorie als Mehrwert	4,8
VD10	Zeichenverlauf der Objekte als Mehrwert	4,0
VD11	Geschwindigkeit des Videos	5,3
VD12	Einfachheit, bestimmten Zeichenschritt zu finden	5,4
VD13	Berechenbarkeit	5,0
VD14	Eignung GUI für Smartphones	5,9
VD15	Eignung GUI für Tablets	6,1

Das Videoplayer-UI erhielt im Allgemeinen (Tab. 5.10) eine schlechtere Bewertung als das Slideshow-UI. Die passivere Haltung des betrachtenden Nutzers wurde von einem Großteil der Probanden als störend empfunden, da das Interaktionsspektrum als zu eingeschränkt wahrgenommen wurde. Folglich ist die Implementierung von weiteren Bedienelementen zur Kompensation nötig.

Die Bewertung Erkennbarkeit der Inhalte (VD08) ist durch eine nicht optimale Qualität des abgespielten Videos zu erklären. Durch eine Implementierung, die die Zeichenschritte mit ihren zeitlichen Informationen speichert und im betrachtenden Modus entsprechend auf den Canvas zeichnet, kann die Auflösung und somit Erkennbarkeit gesteigert werden (EZ4).

Insbesondere sahen die Testpersonen die bereitgestellten zeitlichen Informationen über die

Erstellungshistorie (VD09) und den Zeichenverlauf der Objekte (VD10) weniger als notwendige Komponente (EZ3). Diese stellten den größten Unterschied zum Slideshow-UI dar. Die zeitlichen Informationen über die Erstellungshistorie (VD09), also bspw. die Dauer des Zeichenvorgangs und der Zeitpunkt der Erstellung der einzelnen Primitive wurden von acht der Testpersonen zwar als in Einzelfällen relevant eingestuft, jedoch ist die Darstellung innerhalb eines Videos dafür nicht optimal. Es könnten lange Wartezeiten auftreten. Der Zeichenverlauf der Objekte (VD10) wurde weniger gut bewertet. Als Begründung wurde hier angegeben, dass sich der zeichnende Nutzer vermutlich schon bevor er zur Zeichnung ansetzt, darüber bewusst ist, wie er das Primitiv zeichnen möchte. Trotzdem kann eine Darstellung des Zeichenverlaufs von Primitiven vom betrachtenden Nutzer positiv wahrgenommen werden. Der Touchinput wird durch einen Kreis veranschaulicht und so vereinfacht die Bewegung auf dem Canvas die Wahrnehmung neu entstandener Primitive. Allerdings wären für eine Darstellung des Zeichenverlaufs keine detailgetreuen Informationen über den Zeichenverlauf nötig. Der Zeichenverlauf der Primitive könnte vereinfacht und in einem festgelegten Zeitintervall dargestellt werden.

Acht Testpersonen gaben zudem in VA7 an, dass sie Teile des Videos übersprungen haben (VD11).

Fastforward-/Rewind-Button Um die Geschwindigkeit des Videos (VD11) selbst beeinflussen zu können, sollte durch Gedrückthalten des Fastforward- und des Rewind-Buttons Spulen, also ein schnelleres Abspielen ermöglicht werden. Außerdem könnte der Mediaplayer um die Funktion erweitert werden, das Video schneller als in Echtzeit abzuspielen.

Die Eignung für Smartphones (VD14) wurde schlechter bewertet, als die Eignung für Tablets (VD15), da es in Videos nicht üblich ist, eine Zoomfunktion zu implementieren. Bei größeren Graphiken könnte die Fläche des Displays eines Smartphones zu klein sein. Trotzdem unterscheiden sich die beiden Werte nur geringfügig. Außerdem befürwortete keiner der Probanden eine Trennung der Layouts des Videoplayer-UI für Smartphones und für Tablets (EZ7).

Ein grundlegendes Problem der **Berechenbarkeit** stellte zudem die Tatsache dar, dass fünf der Testpersonen nicht bewusst war, dass sie sich im betrachtenden Modus befinden. Sie versuchten zunächst, selbst auf dem Canvas zu zeichnen.

Einschränkung der bereitgestellten Informatoinen Durch die gering bewertete Berechenbarkeit wurde deutlich, dass nicht alle Informationen aus dem zeichnenden Modus für den betrachtenden Nutzer relevant sind. Alle Befragten gaben an, dass der Verlauf der Texteingabe und das gewählte Primitiv keine relevanten Informationen darstellen. Zusätzlich könnte die Übersichtlichkeit (EZ6) und die Berechenbarkeit (VD13) erhöht und damit die Komplexität des UIs (EZ1) verringert werden.

5.3.4. VERGLEICHE DER UIS

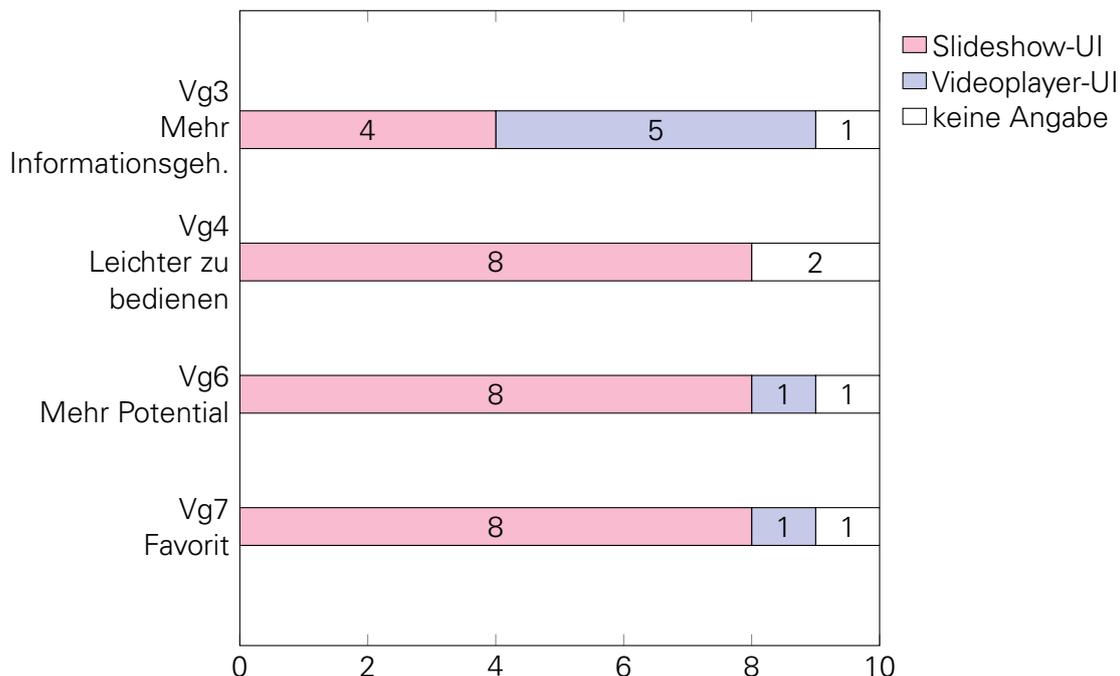


Abbildung 5.7.: Antworten Vergleiche der UIS

Im letzten Fragebogen zum Vergleich von Slideshow-UI und Videoplayer-UI sollte festgestellt werden, ob eines der beiden Interfaces besser für die zukünftige Anwendung geeignet ist. Um eine Bewertung unabhängig von bisherige Entwurfsentscheidungen erreichen zu können, wurden die beiden Interfaces hinsichtlich mehrerer Kriterien in Abb. 5.7 verglichen. Obwohl sich das Videoplayer-UI durch mehr Informationsgehalt auszeichnet, waren die dort präsentierten Informationen für die Nutzer weniger relevant (Vg3). Bezüglich der Komplexität (Vg4) und des Potentials (Vg6) der Interfaces überzeugte das Slideshow-UI deutlich. Auch für die endgültige Software würde ein Großteil der Probanden das Slideshow-UI bevorzugen.

5.4. ANDROID GUIDELINES

In diesem Kapitel werden die bisher getroffenen Entwurfsentscheidungen auf Basis der von Apache veröffentlichten Guidelines für Android von Google [Goo17], sowie den von Apache bereitgestellten Androiddokumentationen [Apa17b] objektiv bewertet.

5.4.1. SLIDESHOW-UI

Actionbar Die verwendete *Actionbar* im oberen Teil des Bildschirms, auch App Bar genannt, gilt laut Android als eine Form von *Toolbar*. Sie dient u.a. zur Navigation und zum Ausführen von Aktionen innerhalb der Anwendung.

Es erfolgt demnach ein sinnvoller Einsatz der *Actionbar* in den UIS, da der primäre Zweck der Implementierung vor allem der Navigation innerhalb der Zeichenschritte dient.

Der Navigation-Button befindet sich im linken Teil der *Actionbar*. In seinen Einsatzzwecken wird direkt das Öffnen eines *Navigation Drawers* erwähnt, was im Slideshow-UI genauso umgesetzt ist.

Der Titel in der Mitte einer *Actionbar* reflektiert laut den Guidelines die aktuelle Seite. Diese Umschreibung ist wenig spezifisch und daher ist keine Bewertung zu erkennen, ob die Platzierung der aktuell dargestellten Gruppe und der Nummer des Zeichenschrittes an dieser Stelle geeignet erscheint.

Auf der rechten Seite der *Actionbar* ist ein Menü-Button vorgesehen. Dort sind sekundäre Aktionen vorgesehen, wie Hilfe oder Einstellungen. Mittels dieses Buttons kann im Slideshow-UI eine Auswahl zwischen Smartphone- und Tabletmodus geöffnet werden.

Die Android Guidelines ordnen die Nutzung der *Actionbar* in der Slideshow-UI demnach als geeignet ein.

Navigation Drawer Mit dem Slideshow-Menü im Slideshow-UI für Smartphones wird ein *temporary* (temporärer) *Navigation Drawer* implementiert. Dieser Typ überdeckt im geöffneten Zustand Inhalt des Bildschirms, in diesem Fall den dargestellten Canvas. Die Verwendung von *Navigation Drawer* auf Smartphones benötigt einen temporären Typ.

Die Breite, die Höhe und das Verhalten des *Navigation Drawer* für Smartphones werden nicht verändert und genügen deshalb den Anforderungen der Guidelines.

Mit der Slideshow-Timeline im Slideshow-UI für Tablets wird ein *permanent Navigation Drawer* implementiert. Dieser Typ kann nicht geschlossen werden. Er wird für Desktopanwendungen empfohlen, was aber der Verwendung auf Tablets nicht widerspricht.

Die Timeline entspricht der Variante, eines *Navigation drawer clipped under the app bar*. Das bedeutet, dass die *Actionbar* nicht von der *Timeline* überdeckt wird.

Die Höhe und das Verhalten des *Navigation Drawers* für Smartphones wurden nicht verändert und entsprechen deshalb den Anforderungen der Guidelines. Zur Breite des permanenten *Navigation Drawers* entspricht für Smartphones und für Tablets der *Side nav width*. Diese definiert sich aus der Breite des Bildschirms abzüglich der Höhe der *Actionbar*. Da in diesem Layout lediglich eine Breite von 20-25% des Displays zur Darstellung der *Timeline* genügen, wurde diese Guideline nicht befolgt.

Im Android Design Guide wird angegeben, dass *Navigation Drawer*, die links bzw. am Start der Anwendung aufgezogen werden kann, ausschließlich Inhalt zur Navigation innerhalb der Anwendung beinhalten sollen. Diese Anforderung wird in der Slideshow-UI für Smartphones und Tablets umgesetzt, da der *Drawer* ausschließlich zur Navigation zu Zeichenschritten im *ViewPager* dient.

Floating Action Buttons (FAB) Im Slideshow-UI werden FABs durch Rewind-, Last-, Next- und Fastforward-Button umgesetzt. Diese sind zeitgleich sichtbar und dicht beieinander angeordnet, da sie eine Einheit zur Navigation zu weiteren Zeichenschritten in der Erstellungshistorie symbolisieren sollen.

Das Aussehen der Buttons wird bezüglich Farbe und Größe nicht manuell verändert. Sie können ausschließlich mit einem einfachen Klicken bedient werden. Navigation ist zudem ein direkt in den Guidelines erwähnter Einsatzzweck. Diese Eigenschaften widersprechen somit nicht ihrem Konzept.

Allerdings wird in den *Guidelines* beschrieben, dass pro Screen nur ein FAB für eine grundlegende Interaktionsmöglichkeit des UIs eingesetzt werden soll. Im Slideshow-UI werden mehrere Buttons verwendet, was der Richtlinie direkt widerspricht. Mehrere FABs mit ähnlichen Aktionen sollten daher laut den *Guidelines* in einer *Toolbar* zusammengefasst werden, was sich ebenfalls aus der Evaluationsbefragung (Kap. 5.2) ergeben hat.

Im Layout existieren bereits zwei *Toolbars* in Form der horizontalen, permanenten *Actionbar* am oberen Bildschirmrand und des vertikalen, temporären *Navigation Drawers*. Bei einer Platzierung der *Toolbar*, die die FABs vereint, am unteren Bildschirmrand erfolgt also keine Überschneidung der *Toolbars*.

Wischgesten Im Slideshow-UI werden mehrere Arten von Wischgesten kombiniert, *Edge*

Swipes und *Paging Swipes*, bei denen es festzustellen gilt, ob eine gleichzeitige Verwendung empfehlenswert ist.

Edge Swipes können eingesetzt werden, um den *Navigation Drawer* zu öffnen. *Edge Swipes* beschreiben eine Wischgeste, die off-screen, also innerhalb des am Bildschirmrand definierten *EdgeSlop*, beginnen, um auf Inhalte zugreifen zu können, die zum Zeitpunkt der Ausführung nicht sichtbar bzw. *off-screen* liegen. Ein *Navigation Drawer* wird standardmäßig und auch in dieser Implementierung durch einen horizontalen *Edge Swipe* vom linken Bildschirmrand aus geöffnet.

Der Canvas, auf dem ein bestimmter Fortschritt des Zeichenverlaufs dargestellt wird, kann mittels *Paging Swipes* bedient werden. Diese *Paging Swipes* beschreiben eine Wischgeste, die horizontal ausgeführt werden und innerhalb des Bildschirms auf dem zu bewegenden Objekt beginnen. Sie dienen dazu, eine weitere Seite oder einen weiteren Tab anzuzeigen. In diesem UI kann damit die Anzahl der dargestellten Zeichenschritte auf dem Canvas verändert werden. Dazu ist eine horizontale Wischgeste auf dem implementierten *Viewpager* und damit auf dem dargestellten Canvas notwendig.

Ist keine Aktion für einen auftretenden *Edge Swipe* definiert, wird dieser als *Paging Swipe* interpretiert. Das bedeutet, dass eine Wischgeste die off-screen vom rechten Bildschirmrand aus beginnt, im Slideshow-UI den *Viewpager* bewegt, obwohl der Touchinput nicht auf dem Canvas begonnen hat.

Es liegt somit keine Überschneidung des Touchinputs vor. Die Guidelines schließen zudem eine parallele Verwendung von *Edge Swipes* und *Paging Swipes* nicht aus. Somit können *Navigation Drawer* und *Viewpager* gleichzeitig eingesetzt werden.

Außerdem wurde in der Evaluationsbefragung (Kap. 5.2) deutlich, dass die Testpersonen die Nutzung beider Elemente nicht als störend empfinden.

5.4.2. VIDEOPLAYER-UI

Floating Action Buttons (FAB) Im Videoplayer-UI wurden FABs zur Umsetzung der Flag-Buttons implementiert. Diese sind horizontal nebeneinander angeordnet und dienen zur Navigation zu Endpunkten von Gruppen der dargestellten Graphik. Pro Gruppe existiert also ein FAB. Die Optik der Flag-Buttons besteht aus zwei kombinierten FABs. Ein bezüglich der Größe unveränderter FAB enthält ein Flaggen-Icon. Ein angepasster, kleinerer Button beinhaltet die Gruppennummer. Die beiden FABs werden überlappend angeordnet, um Zusammengehörigkeit zu signalisieren. Die individuell angepasste Größe widerspricht zwar den Guidelines, jedoch ist eine solche Kombination Nutzern mobiler Geräte bekannt. Beispielsweise bei Nachrichten Anwendungen wird durch eine kleine, kreisförmige Anzeige, die das App-Icon überlappt, angezeigt, wie viele ungelesene Nachrichten vorhanden sind.

Lediglich der größere FAB reagiert auf einfaches Klicken. Da die Flag-Buttons zur Navigation innerhalb der Graphik eingesetzt werden, entsprechen die umgesetzten Bedienmöglichkeiten den Richtlinien.

Wie im Slideshow-UI werden allerdings im Videoplayer-UI mehrere FABs angezeigt. Anderenfalls würde nur eine Gruppe innerhalb der Graphik existieren und somit ein Widerspruch zur Prämisse entstehen, dass Gruppen durch den Nutzer in sinnvoller Größe gewählt werden.

Mehrere FABs sollen zu einer *Toolbar* zusammengefasst werden. In diesem UI existieren bereits zwei horizontale *Toolbars* in Form einer permanenten *Actionbar* am oberen Bildschirmrand und eines horizontalen Mediaplayers am unteren Bildschirmrand. Um eine Mehrfachplatzierung zu vermeiden, ist also entweder eine vertikale Anordnung oder eine Integration in die bestehenden *Toolbars* notwendig.

Bei einer vertikalen Anordnung könnte der bestehende Bezug zur horizontal angeordneten *Progressbar* im Mediaplayer verloren gehen. Da die Flag-Buttons einen direkt den Zustand

des Mediaplayers beeinflussen, sollten die Flag-Buttons ebenfalls im Mediaplayer angeordnet werden.

Einer dortigen Verwendung wurde in der Konzeption ausgeschlossen, um die Bedienung des UIs zu vereinfachen und es zu vermeiden, die Höhe des Mediaplayers verändern zu müssen, da dieser anderenfalls mehr Fläche des Canvas überdeckt.

Mediaplayer Zur Umsetzung des Mediaplayers wird die Klasse *android.media.MediaPlayer* verwendet. Jener wird nicht manuell verändert.

Die Evaluationsbefragung (Kap. 5.2) ergab, dass eine Veränderung der Belegung von Rewind- und Fastforward-Button notwendig sind. Durch langes Drücken, einen *Long Press*, soll Spulen ermöglicht werden. Außerdem sollte bei einem einfachen Klicken das Zeitintervall verändert werden.

Da die Klasse *MediaPlayer* ein einfaches Werkzeug zum Umgang mit Videodateien darstellt, existieren für die Änderung der in ihr enthaltenen Items keine Richtlinien. Bei der Anpassung des Mediaplayers ist demnach eine erneute Evaluation durch Testpersonen notwendig.

5.5. FAZIT

Selbstevaluation Während der Implementierung wurde deutlich, dass die bisherige Speicherung der Graphiken nicht optimal umgesetzt worden ist. Da bisher lediglich lokale Daten für die Umsetzung des Prototypen verwendet worden sind, führten diese zu *OutOfMemoryError*. Vor allem bei der Benutzung des *Slideshow-Menüs* und der *Slideshow-Timeline* werden alle angezeigten Graphiken beim Scrollen erneut geladen, somit erneut gezeichnet und beanspruchen erneut Speicherplatz.

Im Vorfeld der Befragung wurde daher der der Anwendung zugewiesene Speicher auf dem Mobilgerät maximiert. Dazu wurde in der *AndroidManifest.xml* ein *largeHeap* aktiviert [Apa17a], was das Auftreten des Problems nicht behob, aber deutlich verzögerte.

Während der Testphase führte das beschriebene Problem einmal zur Beendigung des Prototypen. Durch den Testleiter wurde der Testperson erklärt, dass der Absturz keinen Zusammenhang mit dessen Interaktionen hatte. Daher beeinflusste das Speichermanagement die Befragung und die damit erzielten Ergebnisse nicht.

Durch die später vorgesehene Anbindung eines Backends kann die Speicherung optimiert und dadurch das Problem behoben werden.

Auswahl eines UIs Da sich die Testpersonen deutlich für das Slideshow-UI entschieden, wird ausschließlich dieses im nächsten Iterationsschritt weiter untersucht. Eventuelle Vorteile des Videoplayer-UI, wie die zeitlichen Informationen über den Zeichenverlauf, könnten zusätzlich in das Slideshow-UI integriert werden. Alle Testpersonen vertraten die Meinung, dass die zeitlichen Details nur in Einzelfällen wichtige Informationen über den Erstellungsprozess darstellen. Außerdem würden die zeitlichen Details einen bedeutend größeren Aufwand bezüglich der Speicherung eines Canvas bedeuten. Deshalb wurde dieser Ansatz nicht weiter implementiert (EZ8).

6. OPTIMIERUNG NACH EVALUATION

Aus der Evaluation für das Slideshow-UI für Smartphone (Kap. 5.3.1) und Tablet (Kap. 5.3.2) ergaben sich konkrete Kritikpunkte und somit neue Entwurfsentscheidungen für die Layouts, die die Benutzerfreundlichkeit erhöhen sollen. In diesem Kapitel werden die Änderungen im Design im Prototypen in einem weiteren Iterationsschritt umgesetzt.

Die zugehörigen Abbildungen wurden zur Verdeutlichung der umgesetzten Optimierung mit einem * gekennzeichnet.

Auswahl eines UIs Die wichtigste Entscheidung für die Weiterentwicklung des Prototypen war die Auswahl zwischen *Slideshow-UI* und *Videoplayer-UI*. Die Evaluationsbefragung ergab beim Vergleich der beiden Interfaces (Kap. 5.3.4) eine Bevorzugung des *Slideshow-UI*. Die Änderungen für das *Videoplayer-UI* werden deshalb in diesem Iterationsschritt nicht umgesetzt.

6.1. PROTOTYPOPTIMIERUNG

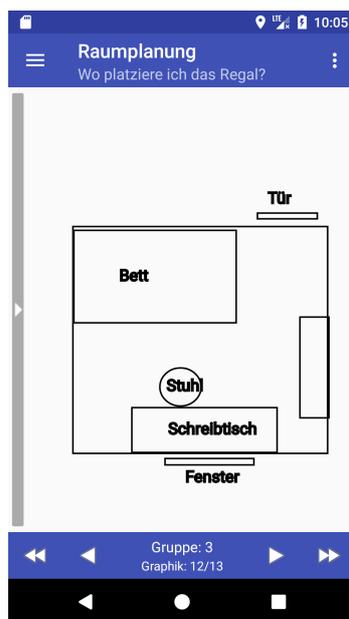


Abbildung 6.1.: Slideshow-UI Smartphone*

In diesem Kapitel werden die Änderungen in der Implementierung des Slideshow-UI angewandt.

6.1.1. SLIDESHOW-UI

Zunächst werden die Änderung der Implementierung erläutert, die sowohl das Layout für Smartphones, als auch das Layout für Tablets betraf.

Toolbar Zum besseren Verständnis der Funktionsweise des Fastforward- und Rewind-Buttons wurde eine *Toolbar* entworfen. Diese befindet sich am unteren Bildschirmrand in Abb. 6.1 und 6.4. Ihre Gestaltung wurde der *Actionbar* am oberen Bildschirmrand nachempfunden, da beide Bedieneinheiten zur Navigation innerhalb der Erstellungshistorie dienen. Die Höhe und die Farbe der *Actionbar*, die *colorPrimary* einer Androidanwendung, wurden nicht manuell verändert. Beide Parameter wurden für die *Toolbar* übernommen.

Die Farbgestaltung soll darauf hinweisen, dass mittels der *Toolbar* ebenfalls eine Navigation ermöglicht wird. Außerdem fasst ein einheitlicher Hintergrund die ähnlich agierende

Bedienelemente und zugehörige Informationen optisch zusammen. Während der Evaluation (Kap. 5.3.1) wurde außerdem angemerkt, dass eine Farbgebung in einem Pinkton zu grell und daher nicht für die Buttons geeignet ist.

Die Toolbar beinhaltet alle Slideshow-Buttons, welche den Rewind-, den Last-, den Next-, sowie den Fastforward-Button (v.l.n.r.) umfassen (Kap. 3.3). Die Informationen über die zugehörige Gruppe des aktuell im *Viewpager* angezeigten Zeichenschritts, sowie die Nummer des Zeichenschritts werden in der Mitte der *Toolbar* aufgeführt. Am Anfang bzw. Ende der Erstellungshistorie werden die beiden linken bzw. die beiden rechten Buttons weiterhin ausgeblendet.

Da diese Informationen nun nicht mehr innerhalb der *Actionbar* platziert werden, können dort andere, weniger dynamische Informationen angezeigt werden. In Abb. 6.1 und 6.4 wurde ein beispielhafter Titel der Zeichnung und eine kurze, zugehörige Fragestellung eingeblendet. Diese Art von Information ist vielen potentiellen Nutzern aus Emailprogrammen bekannt sein, bei denen in der *Actionbar* Betreff und Sender der Email dargestellt werden.

6.1.2. SMARTPHONE: SLIDESHOW-MENÜ

Fortfolgend werden die konkret implementierten Änderung am *Slideshow-UI* für Smartphones im zweiten Iterationsschritt erläutert.

Drag-Element Eine wichtige Entwurfsentscheidung des Slideshow-Menüs stellte die Implementierung eines Drag-Elementes dar, um die Interaktion mit dem Menü einfacher und die Existenz des Menüs offensichtlicher zu gestalten. Dieses ist in Abb. 4.3 an der linken Seite des Bildschirms zu sehen. Das Design wurde ähnlich des Designs des *Adobe Acrobat Reader DC* (Abb. 3.1) entworfen.

Die Breite des Drag-Elementes wurde analog zur Breite des Bereiches gewählt, in dem das Slideshow-Menü geöffnet werden kann. Diese Breite entspricht dem *EdgeSlop* der *ViewConfiguration* der jeweiligen Androidanwendung. In Abb. 4.3 beträgt der *EdgeSlop* 36 Pixel. Als Icon wurde ein Pfeil gewählt, der die Richtung der Interaktion symbolisiert.

Das Slideshow-Menü kann weiterhin durch horizontales Wischen geöffnet werden. Der Touchinput der Wischgeste muss innerhalb des *EdgeSlop* beginnen.

Eine neue Interaktionsmöglichkeit zum Öffnen des Menüs wird durch das *Drag-Element* gegeben, das zusätzlich angeklickt werden kann.

Der zukünftige Nutzer soll in jedem Fall ein Bedienelement realisieren. Mit dem Drag-Element kann also auf mehrere Arten interagiert werden, um das Menü zu Öffnen.

Das Drag-Element verändert sich im geöffneten Zustand des Menüs, wie in Abb. 6.2 sichtbar. Der Pfeil des Buttons ändert dabei die Richtung. Durch ein Betätigen des Buttons kann das Menü zusätzlich zu den bestehenden Möglichkeiten geschlossen werden.

Übersichtlichkeit des Slideshow-Menüs Um die Übersichtlichkeit des Slideshow-Menüs zu steigern, wird das Layout verändert. Die *HorizontalScrollView*, die alle Items der Erstellungshistorie einer Gruppe beinhalten, werden nun in

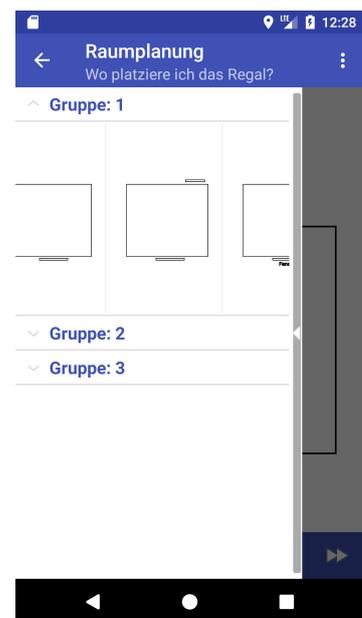


Abbildung 6.2.: Slideshow-Menü* Smartphone

einer *ExpandableListView* angeordnet, wie in Abb. 6.2 zu sehen ist. Das bedeutet, dass die Gruppen durch Klicken auf den Gruppenname ausgeklappt werden können. Beim ersten Öffnen des Menüs werden nur die Namen der Gruppen angezeigt.

Links neben den Gruppennamen befindet sich ein Pfeil nach unten, falls die Gruppe nicht ausgeklappt worden ist (Abb. 6.2 Gruppe 3). Anderenfalls zeigt er nach oben, um zu symbolisieren, dass die Gruppe durch Klicken auf den Gruppenname geschlossen werden kann (Abb. 6.3 Gruppe 3).

Der Text der Gruppennamen wird zudem in der Farbe *colorPrimary* der App und in fett angezeigt, um die Lesbarkeit zu verbessern.

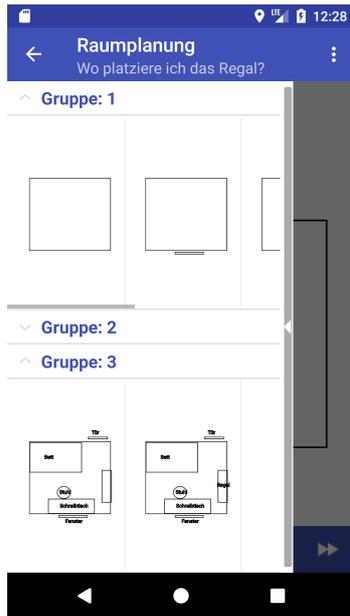


Abbildung 6.3.: Slideshow-Menü* erweitert

Divider Um die Übersichtlichkeit im Slideshow-Menü zu steigern, werden sogenannte *Divider* implementiert. Der Begriff umschreibt die hellgrauen Trennlinien zwischen den Graphiken.

*ListView*s und *LinearLayout* implementieren diese *Divider* standardmäßig. Sie werden üblicherweise in der Farbe *Color.LTGRAY* (hellgrau) gestaltet und haben bei einem vertikal angeordneten Layout eine Höhe von 1 Pixel, da diese die kleinste darstellbare Höhe ist. Die Breite der *Divider* passt sich der Breite des zu unterteilenden Layouts an.

Im Slideshow-Menü werden zur Unterteilung der *ExpandableListView* *Divider* mit einer Höhe von 3 dp (*Density-independent Pixels*) verwendet, um eine stärkere Unterscheidung zu signalisieren. Die Höhe wird in dp gewählt, da diese unabhängig von der Auflösung auf jedem Display die gleiche Größenangabe erzielen.

Im horizontal angeordneten *LinearLayout* in der *HorizontalScrollView* werden *Divider* zwischen allen Zeichenschritten mit einer Breite von 1 Pixel gesetzt.

Die optische Trennung der einzelnen Zeichenschritte soll aufzeigen, dass die Miniaturansichten der Graphiken auch als Buttons fungieren und damit angeklickt werden können.

6.1.3. TABLET: SLIDESHOW-TIMELINE

Fortfolgend werden die konkret implementierten Änderungen am *Slideshow-UI* für Tablets im zweiten Iterationsschritt erläutert.

Übersichtlichkeit der Slideshow-Timeline Im *Slideshow-UI* für Tablets wurde in der Evaluation festgestellt, dass die Gruppen zunächst zusammengefasst werden sollten und dann vom Nutzer nach Bedarf durch Klicken ausgeklappt werden können. Dazu wird statt einer *ListView* eine *ExpandableListView* implementiert. Ein Eintrag der *ExpandableListView* enthält die zu jedem Zeichenschritt korrespondierende *ImageView*, sowie eine *TextView*, die den Gruppennamen repräsentiert.

Der Text der Gruppennamen wird zudem in der Farbe *colorPrimary* der App und in fett angezeigt, um die Lesbarkeit zu verbessern.

Präsenz der *Slideshow-Timeline* Die wichtigste Entwurfsentscheidung durch die Evaluation (Kap. 5.3.2) im *Slideshow-UI* für Tablets stellt die Änderung der Präsenz der *Slideshow-Timeline*

line dar.

Die Unterschiede der Slideshow-UIs für Smartphone und Tablet sollen zu Gunsten der Benutzerfreundlichkeit so gering wie möglich gehalten werden. Aufgrunddessen wurde beschlossen, dass der Nutzer selbst entscheiden kann, ob die Slideshow-Timeline sichtbar ist oder nicht. Während der Implementierung (Kap. 4.2.2) war eine ständige Präsenz zwingend.

Das Grundprinzip des Layouts soll weiterhin erhalten bleiben. Der Nutzer kann also auch während die *Timeline* geöffnet ist mittels Wischgesten mit dem *Viewpager* im rechten Teil des Bildschirms interagieren.

Zur Umsetzung dieses Prinzips wird ein benutzerdefiniertes *SlidingPaneLayout* implementiert. Dieses ermöglicht es, analog zum Layout für Smartphones, die Timeline über horizontales Wischen, das von der linken Kante in Breite des *EdgeSlop* ausgeht, zu öffnen. Zusätzlich kann das *SlidingPaneLayout* über den Navigation-Button geöffnet und geschlossen werden. Dieser erhält das gleiche Symbol, wie standardmäßig für *Navigation Drawer* vorgesehen, den *DrawerIconDrawable* (auch „Burger-Icon“). Dieser transformiert sich während der Bewegung des *SlidingPaneLayouts* von drei parallelen, horizontalen Strichen (geschlossen, Abb. 6.4) zu einem Pfeil (offen, Abb. 6.6).

Zum Schließen der Timeline dient außerdem der Back-Button des Mobilgerätes.

Die Größe des *Viewpagers* und der *Toolbar* passt sich je nach Status der Timeline der restlichen Größe des Displays an.

Die Bedienmöglichkeiten, sowie die Gestaltung von Slideshow-Menü und Slideshow-Timeline werden so ähnlich wie möglich umgesetzt. Interaktionen wie das Öffnen und das Schließen der Elemente soll konform möglich sein, um den Nutzern den Umgang mit beiden Layouts zu vereinfachen.

Drag-Element Außerdem ist ein Öffnen über das ebenfalls implementierte Drag-Element möglich, welches in Abb. 6.4 am linken Bildschirmrand zu sehen ist. Wie im Layout für Smartphones entspricht die Breite des Drag-Elements der des *EdgeSlop*, die Farbgebung erfolgt in neutralem Hellgrau (*Color.LTGRAY*) und das Icon besitzt die Form eines Pfeils, der in die Richtung der korrespondierenden Wischgeste verweist.

Ist die Timeline geöffnet, ändert sich die Richtung des Pfeils. Das Drag-Element dient dazu, das *SlidingPaneLayout* durch Klicken zu schließen. Wischgesten zum Schließen des Layouts werden, wie im UI für Smartphones, gesperrt. Anderenfalls würden sich die Wischgesten zum Schließen der Timeline und zum Bewegen des *Viewpagers* überlappen. Außerdem könnte eine vertikale Bedienung der Timeline erschwert werden, da auch bei Wischgesten zum Scrollen auch unbeabsichtigt Wischgeste von rechts nach links beinhalten kann.

Breite der Slideshow-Timeline Die Breite der Slideshow-Timeline wird auf die Breite der dort dargestellten Graphiken reduziert. Die Breite beträgt nun inklusive des Buttons zum Schließen der Timeline 165 dp. Diese Breite ist unabhängig von der Auflösung und wurde in der Evaluation durch die Testpersonen als ideal eingestuft, um einen Kompromiss zwischen

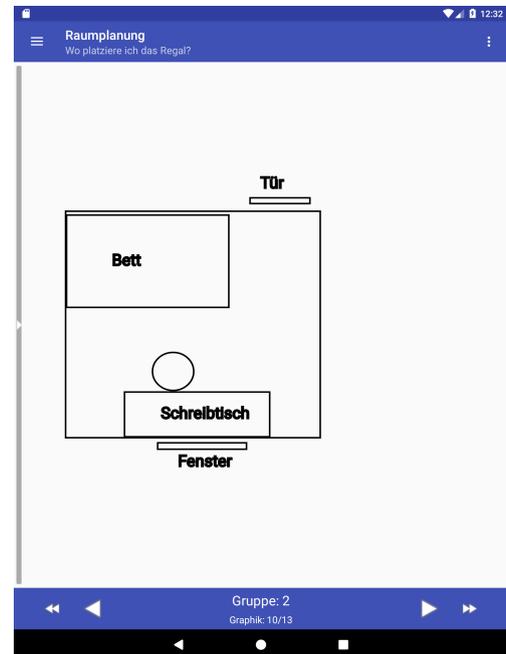


Abbildung 6.4.: Slideshow-UI* Tablet

zusammengefassten Informationen und Erkennbarkeit der Graphiken zu erreichen.

Divider Um die Übersichtlichkeit in der Slideshow-Timeline zu erhöhen, wurden sogenannte *Divider* analog zum Design des Slideshow-Menüs angepasst (Abb. 6.6).

Bereits in der Implementierung waren *Divider* zwischen den Gruppennamen und den Zeichenschritten vorhanden. Diese wurden in der Höhe von 1 Pixel auf 3 dp erhöht.

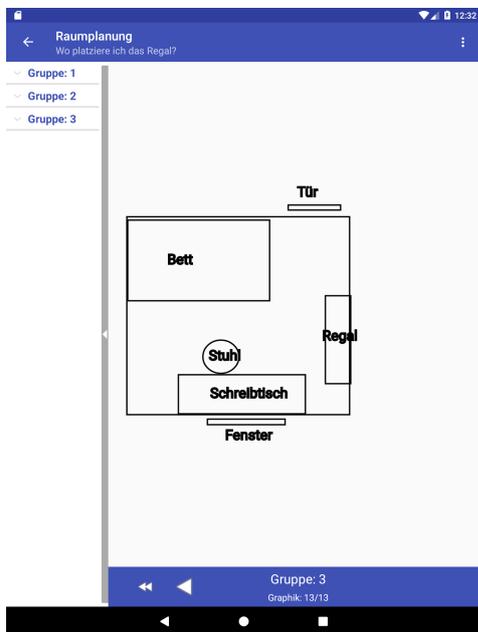


Abbildung 6.5.: Slideshow-Timeline*
Tablet minimal

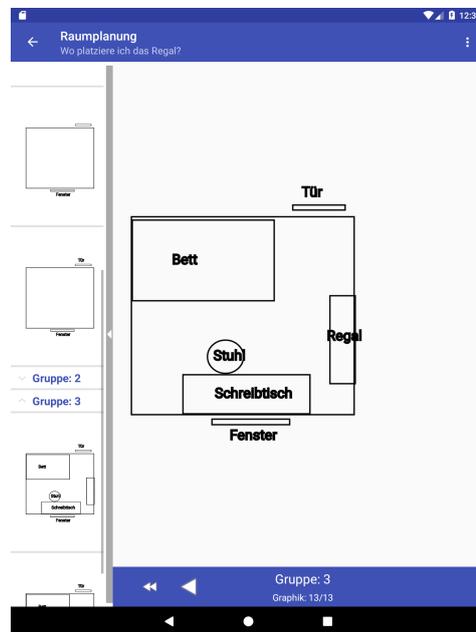


Abbildung 6.6.: Slideshow-Timeline*
Tablet erweitert

6.2. VERIFIKATION DER OPTIMIERUNG

In diesem Kapitel sollen die in der Evaluation gefunden Mängel am Slideshow-UI und in der Prototypoptimierung umgesetzten Änderungen im Design evaluiert werden.

6.2.1. VERIFIKATIONSZIELE

Im Fokus dieser Evaluation liegt also die detaillierte Untersuchung von bisher als mangelhaft eingestuft Bedienelementen.

Weiterhin sollen die bisher erreichten Bewertungen der Interfaces nicht negativ beeinflusst werden.

Zunächst wird aufgelistet, welche Entwurfsentscheidungen aus der Evaluation auf welche Variante des *Slideshow-UI* angewendet wurde:

- *Slideshow-UI* für Smartphone
 - Toolbar
 - Drag-Element
 - Übersichtlichkeit des *Slideshow-Menüs*

- Divider
- *Slideshow-UI* für Tablet
 - Toolbar
 - Übersichtlichkeit der *Slideshow-Timeline*
 - Präsenz der *Slideshow-Timeline*
 - Drag-Element
 - Breite der *Slideshow-Timeline*
 - Divider

Fortfolgend werden die genauen Bestimmungen der Evaluationsziele definiert. Die Nummerierung erfolgt fortlaufend von den Zielen der Evaluation (Kap. 5.1):

EZ9: Toolbar Für die in der Toolbar verwendeten Buttons muss festgestellt werden, ob sie als eindeutig, sowie optisch geeignet eingestuft werden.

EZ10: Drag-Element Für das *Drag-Elementes* muss festgestellt werden, ob die Offensichtlichkeit des *Slideshow-Menüs* durch die Implementierung erhöht werden kann. Außerdem gilt es festzustellen, inwiefern die Interfaces unter Verwendung eines *Drag-Elementes* bezüglich Berechenbarkeit und Komplexität beeinflusst werden.

EZ11: Divider Es gilt festzustellen, ob die verwendeten *Divider* zur Übersichtlichkeit von *Slideshow-Menü* und *Slideshow-Timeline* beitragen.

EZ12: Übersichtlichkeit von *Slideshow-Menü* und *Slideshow-Timeline* Eine erhöhte Übersichtlichkeit von *Slideshow-Menü* und *Slideshow-Timeline* sollte durch die Verwendung von *ExpandableListViews*, sowie der optischen Hervorhebung der Gruppennamen umgesetzt werden. Es gilt, festzustellen, ob die Maßnahmen die Navigation zu beliebigen Elementen der Erstellungshistorie vereinfachen und die Übersichtlichkeit erhöhen.

EZ13: Präsenz der *Slideshow-Timeline* Es muss festgestellt werden, ob durch die Möglichkeit, die *Slideshow-Timeline* auszublenden, den Umgang mit dem Interface erschwert NUTZER. Im Vordergrund stehen bei dieser Untersuchung die Komplexität, Notwendigkeit und Optik der ergänzten Bedienungsmöglichkeiten für Interaktionen, die die Präsenz der *Timeline* beeinflussen.

EZ14: Breite der *Slideshow-Timeline* Es gilt, festzustellen, ob die Breite der *Timeline* ideal gewählt worden ist. Die Darstellungsgröße hat direkten Einfluss auf die Übersichtlichkeit der *Timeline*, sowie die Komplexität des Umganges mit dem gesamten Interface für Tablets.

6.2.2. VERIFIKATIONSBEFRAGUNG

Die Verifikationsbefragung zur Evaluation des *Slideshow-UI* wird analog zur Evaluationsbefragung (Kap. 5.2) aufgebaut. Der Versuchsaufbau und die Erhebungsmethoden unterscheiden sich nur insofern, dass ausschließlich das *Slideshow-UI* untersucht wird.

Die Auswahl der Probanden erfolgt nach den gleichen Kriterien und beinhaltet, dass die Testpersonen bisher keine Vorkenntnisse zu *Graphicuss* besitzen und mindestens seit 3 Jahren

ausschließlich Mobilgeräte mit dem Betriebssystem Android verwenden. Die Zusammenstellung der Probanden wird in Tabelle 6.1 ersichtlich.

Tabelle 6.1.: Einordnung der Befragten Verifikationsbefragung

Gruppe	Geschlecht	Altersgruppe	Berufsfeld	Anzahl der Befragten
IT2	weiblich	20 - 39	Informatik	1
IT3	männlich	40 - 60	Informatik	1
DI1	männlich	20 - 39	Anderes	2
DI2	weiblich	20 - 39	Anderes	1

Es wurden weiterhin auditive Mitschnitte und Screencasts erstellt, sowie der *AB-Test* für die Interfaces für Smartphones und Tablets angewendet.

Den Probanden wird zu jedem gezeigten Layout jeweils ein Fragebogen vorgelegt, die jeweils 17 Detailitems beinhalten (Anhang A.4). Dabei werden alle Detailitems (SD) aus den Fragebögen in Evaluation I (Kap. 5.2, Anhang A.3) übernommen. Der Fragebogen für das *Slideshow-UI* wird zusätzlich um vier Items erweitert, die sich auf das Öffnen und Schließen der *Slideshow-Timeline* beziehen.

6.2.3. AUSWERTUNG DER VERIFIKATIONSBEFRAGUNG

Wie in der Evaluationsbefragung (Kap. 5.3) wird ein Grenzwert von 6,0 definiert. Die Zeilen mit den Items, die diesen Wert nicht erreicht werden, wurden grau markiert. Zudem wurden die Zeilen markiert, bei denen sich der Wert im Vergleich zu Evaluationsbefragung um 0,5 oder mehr verschlechtert hat. Eine Schwankung von +/- 0,5 kann durch persönliche Einstellung der Befragten entstehen.

Bei den markierten Zeilen gilt es, Gründe für die Verschlechterung bzw. den Wert an sich zu finden und ggf. erneut Entwurfsentscheidungen zur Behebung zu treffen, um den Zielwert von mehr als 6,0 in der nächsten Iteration zu erreichen.

SLIDESHOW-UI SMARTPHONE

Tabelle 6.2.: Bewertung Slideshow-Buttons für Smartphone II

Item	Beschreibung	Wert	Änderung Evaluation
SD01	Positionierung	7,0	+ 0,6
SD02	Notwendigkeit	6,5	- 0,5
SD03	Eindeutigkeit	6,8	+ 1,3
SD04	Optik (Farbe, Größe, Icon)	6,5	+ 0,6

In Tabelle 6.2 ist bei allen Items zur Bewertung der Slideshow-Buttons der Grenzwert deutlich erreicht worden. Damit wurde gezeigt, dass die implementierte *Toolbar* dazu führte, dass der Fastforward- und der Rewind-Button in ihrer Funktion leichter nachvollziehbar wurden (SD03, EZ9). Zudem hat sich die Bewertung der Optik durch die weniger grelle Farbe verbessert (SD04).

Die Verschlechterung des Wertes zu Item SD02 ist durch einen einzelnen Befragten entstanden, der die Implementierung ausdrücklich Wischgesten den Slideshow-Buttons vorzieht.

Tabelle 6.3.: Bewertung Slideshow-Menü für Smartphone II

Item	Beschreibung	Wert	Änderung Evaluation
SD05	Positionierung	6,8	+ 0,3
SD06	Notwendigkeit	6,3	- 0,1
SD07	Erkennbarkeit der Navigationsmöglichkeit mit dem Menü	6,8	+ 1,8
SD08	Übersichtlichkeit	6,5	+ 1,2
SD09	Erkennbarkeit der Graphiken	7,0	+ 2,0
SD10	Offensichtlichkeit Existenz	6,5	+ 1,9
SD11	Leicht zu Öffnen	6,5	- 0,5
SD12	Leicht zu Schließen	6,8	+ 0,4
SD13	Erwartungskonform, Menü mit horizontalem Wischen zu öffnen	6,0	+ 2,4

Die Bewertung des Slideshow-Menüs in Tabelle 6.3 zeigt auf, dass alle Kriterien den Grenzwert erreicht haben.

Auffällig bei dieser Gruppe von Befragten war die deutliche Verbesserung von SD13. Den Testpersonen der Verifikationsbefragung war das Prinzip, mit horizontalem Wischen ein Menü zu öffnen, eindeutig im Vorfeld bekannt. Zurückzuführen ist diese Tatsache eventuell auf die im Durchschnitt jüngeren Probanden. Durch den veränderten Stand der Vorkenntnisse ist keine Verfälschung der Ergebnisse zu erwarten, da mit dem Drag-Element ein neues Bedienelement zur Veranschaulichung der Möglichkeit implementiert worden ist. Dieses Vorwissen hat also für die weitere Implementierung an Relevanz verloren.

Deutlich verbessert hat sich die Offensichtlichkeit der Existenz (SD10) durch das Drag-Element und die Erkennbarkeit der Navigationsmöglichkeit (SD07) durch die implementierten *Divider* (EZ10, EZ11).

Durch die implementierte *ExpandableListView* konnten Übersichtlichkeit (SD08), sowie Erkennbarkeit der Graphiken (SD09) eindeutig nutzerfreundlicher gestaltet werden (EZ12).

Eine Verschlechterung der Wahrnehmung, wie leicht das Slideshow-Menü geöffnet werden kann (SD11), ist ebenfalls auf das Drag-Element zurückzuführen. Zwei der Probanden empfanden es nicht als einfach, das Menü durch horizontales Wischen bzw. Klicken auf das Drag-Element zu öffnen, da der Bereich des *EdgeSlop* relativ klein ist. Aus diesem Grund existiert weiterhin die Möglichkeit, das Menü über den Navigation-Button zu öffnen. Die Bewertung von SD11 fiel aufgrunddessen trotzdem sehr gut aus.

Tabelle 6.4.: Bewertung Slideshow-UI für Smartphone Allgemein II

Item	Beschreibung	Wert	Änderung Evaluation
SD14	Erwartungskonforme Reaktion auf Wischgesten	6,8	+ 1,0
SD15	Einfachheit, bestimmte Zeichenschritte zu finden	6,0	- 0,5
SD16	Berechenbarkeit	6,0	+ 0,6
SD17	Eignung GUI für Smartphones	7,0	+ 0,5

Die Bewertung des Slideshow-UIs für Smartphones zeigte im Allgemeinen ebenfalls, dass alle Items den Grenzwert erreicht haben.

Die Bewertung der Einfachheit, einen bestimmten Zeichenschritt zu finden (SD15) hat sich lediglich in einem kleinen Rahmen verschlechtert. Der Wert ist durch das persönliche Empfinden eines Nutzers entstanden, der Wischgesten der Benutzung von Buttons vorzieht. Wischgesten innerhalb des *Viewpagers* stellen ein relativ umständliche Interaktionsmöglichkeit zum Erreichen von weit entfernten Zeichenschritten dar.

Wichtig ist die Verbesserung der Berechenbarkeit, da diese ein ausschlaggebendes Kriterium für die wahrgenommene Komplexität durch die Nutzer darstellt.

SLIDESHOW-UI TABLET

Tabelle 6.5.: Bewertung Slideshow-Buttons für Tablet II

Item	Beschreibung	Wert	Änderung Evaluation
SD01	Positionierung	6,8	+ 2,7
SD02	Notwendigkeit	6,3	- 0,7
SD03	Eindeutigkeit	6,8	+ 0,5
SD04	Optik (Farbe, Größe, Icon)	6,3	+ 1,8

In Tabelle 6.5 wird ersichtlich, dass die Slideshow-Buttons im UI für Tablets in allen Kriterien sehr gut bewertet worden sind.

Eine wichtige Verbesserung wurde in der Bewertung der Optik (SD04) und der Eindeutigkeit (SD03) durch die Implementierung der Toolbar erzielt (EZ9). Kritisiert wurde von zwei der Probanden, dass die Icons des Last- und des Next-Button größer dimensioniert worden sind als die Icons des Rewind- und des Fastforward-Buttons. Die Größe muss entsprechend angepasst werden. Im Interface für Smartphones ist der Unterschied demnach auch aufgetreten, wurde nur durch das kleinere Display nicht bemerkt.

Die Verschlechterung des Wertes der Notwendigkeit der Buttons ist ebenfalls auf einen Nutzer zurückzuführen, der Wischgesten deutlich bevorzugt. Dennoch wurden die Slideshow-Buttons von den Testpersonen eindeutig als notwendig eingestuft.

Tabelle 6.6.: Bewertung Slideshow-Timeline für Tablet II

Item	Beschreibung	Wert	Änderung Evaluation
SD19	Notwendigkeit	7,0	+/- 0,0
SD20	Offensichtlichkeit der Navigationsmöglichkeit	7,0	+ 0,5
SD21	Übersichtlichkeit	7,0	+ 2,2
SD22	Erkennbarkeit der Graphiken	7,0	+ 0,5
SD23	Einfachheit Bedienung	7,0	+ 0,1
SD26	Offensichtlichkeit Existenz	7,0	-
SD27	Leicht zu Öffnen	7,0	-
SD28	Leicht zu Schließen	5,8	-

In Tabelle 6.6 wird deutlich, dass alle Items, die in der Evaluation (Kap. 5.3.2) ebenfalls im Fragebogen enthalten waren (SD19 bis SD23), den Maximalwert von 7,0 erreicht haben.

Eine deutliche Verbesserung wurde in der Übersichtlichkeit (SD21) durch die Verwendung von *ExpandableListView*, die Hervorhebung der Gruppennamen und die vergrößerten *Divider* erzielt (EZ12, EZ11). Die schmalere Gestaltung der Timeline beeinflusste diese Kriterien nicht und wurde von keinem der Testperson als störend empfunden (EZ14).

Weiterhin galt es, die Änderung innerhalb der Präsenz der Timeline zu evaluieren (EZ13). Diese wurde in den neu definierten Items SD26, SD27 und SD28 hinterfragt. Hinsichtlich der Offensichtlichkeit der Existenz (SD26) wurde der Idealwert von 7,0 erreicht, da die Timeline beim Öffnen des Slideshow-UI zunächst geöffnet dargestellt wird und im geschlossenen Zustand ein Drag-Element eingeblendet wird (EZ10).

Aufgrund der analog implementierten Bedienmöglichkeiten zum Interface für Smartphones zum Öffnen der Timeline wurde ebenfalls im Item SD27 der Maximalwert erreicht. Das *SlidingPaneLayout* wurde dem eines *Navigation Drawer* so ähnlich gestaltet, wie möglich, weshalb die Bedienung des relativ unbekanntes Layouts trotzdem leicht fiel.

Die Bewertung, wie leicht es möglich ist, die Timeline zu schließen (SD28), hat den Grenzwert von 6,0 knapp nicht erreicht. Zurückzuführen ist diese Einschätzung der Testpersonen auf die Breite des Drag-Elementes, das der Breite des *EdgeSlop* entsprach. Der *EdgeSlop* beträgt auf beiden Mobilgeräten, auf denen die Testpersonen den Prototypen untersuchten, 36 Pixel. Somit erfolgte keine Vergrößerung des Drag-Elementes bei der Tabletversion. Die Probanden empfanden es als schwierig, das Drag-Element anzuklicken. Daraus ist zu schlussfolgern, dass das Drag-Element im geöffneten Zustand der Timeline verbreitert werden muss. Trotz des schmal gestalteten Drag-Elementes erfolgte eine sehr gute Bewertung von Item SD27, da zum Öffnen der Timeline zusätzlich Wischgesten verwendet werden können. Eine Verbreiterung des Drag-Elementes im geschlossenen Zustand der Timeline ist also nicht notwendig.

Tabelle 6.7.: Bewertung Slideshow-UI für Tablet Allgemein II

Item	Beschreibung	Wert	Änderung Evaluation
SD14	Erwartungskonforme Wischgesten	7,0	+ 0,5
SD15	Einfachheit, bestimmten Zeichenschritt zu finden	6,5	+ 1,2
SD16	Berechenbarkeit	6,8	+ 0,2
SD24	Eignung GUI für Tablets	6,8	+ 0,3
SD25	Notwendigkeit Interface für Tablets	4,0	- 2,3

In Tabelle 6.7 wird die allgemeine Bewertung des Slideshow-UI dargestellt.

Die Einfachheit, einen bestimmten Zeichenschritt zu finden (SD15) hat sich deutlich erhöht, was darauf hinweist, dass die verwendete *ExpandableListView* die Übersichtlichkeit der Timeline und somit ebenfalls die Benutzerfreundlichkeit erhöht.

Die Notwendigkeit einer Tabletversion (SD25) hat bedeutend an Zuspruch verloren und liegt bei dem indifferenten Wert von 4,0. Die Bewertung gibt also keine Information darüber, ob die Testpersonen ein eigenes Interface für Tablets für notwendig empfinden oder nicht. Zwei der fünf Testpersonen wählten direkt den Wert 4,0 und damit die Mitte der Skala. Drei der Testpersonen wiesen also eine Tendenz auf. Um die Tendenz jedes einzelnen Nutzer umsetzen zu können, wird zwar standardmäßig bei einem als Tablet erkannten Gerät zunächst das korrespondierende Interface angezeigt, jedoch ist es auf jedem Mobilgerät möglich, zu wählen, ob das Slideshow-UI für Smartphones oder für Tablets dargestellt wird.

Durch die Implementierung einer Timeline, die mittels Wischgesten geöffnet werden kann, hat sich die Konformität bezüglich Wischgesten (SD14) zum Positiven verändert. Auch die Berechenbarkeit (SD16) erhöhte sich minimal. Diese Items weisen darauf hin, dass die Timeline benutzerfreundlich zu bedienen ist, der Workflow des gesamten UIs erwartungskonform

abläuft und somit die das Slideshow-UI für Tablets keine größeren Mängel mehr aufweist.

6.2.4. ANDROID GUIDELINES

Es erfolgt anschließend eine Auswertung der in der Optimierung implementierten Objekte nach den von Google veröffentlichten Guidelines für Android [Goo17], sowie den von Apache bereitgestellten Androiddokumentationen [Apa17b] bewertet.

Toolbar Die *Toolbar* entstand in der Optimierung aus der Vereinigung der *Floating Action Buttons*. Sie beinhaltet vier Buttons zur Navigation innerhalb der Erstellungshistorie, sowie die Nummer des aktuell dargestellten Zeichenschrittes und der zugehörigen Gruppe. Sie enthält somit fünf Elemente.

Eine *Bottom Bar* sollte laut den Guidelines drei bis fünf Elemente beinhalten, die jeweils einen direkten Zugriff erlauben. Sie sollte bei einem Überlauf der *Actionbar* am oberen Bildschirmrand verwendet werden. Es existieren keine Spezifizierungen zu der verwendeten Farbe, jedoch in keinem der Beispiele der *colorAccent* der App (in diesem Fall pink) verwendet. Die Höhe der *Toolbar* sollte der Höhe der *Actionbar* entsprechen.

Alle Anforderungen an eine *Bottom Bar* wurden demnach von der implementierten *Toolbar* im *Slideshow-UI* umgesetzt.

SlidingPanelLayout (SPL) Zur Umsetzung der Slideshow-Timeline wurde ein *SlidingPanelLayout* implementiert. Es dient primär dazu, ein Layout, das zwei Fenster beinhaltet, an größere Displaygrößen anpassen zu können.

Das SPL wurde im *Slideshow-UI* für Tablets zur Umsetzung der Slideshow-Timeline implementiert, das konvergierend zum *Slideshow-Menü* im Layout für Smartphones verwendet werden soll. Damit wurde ein gleicher Verwendungszweck und eine ähnliche Interaktionsweise vorausgesetzt.

In den Guidelines wird das SPL direkt von einem *Navigation Drawer* abgegrenzt. Sie sollten nicht in den gleichen Szenarien verwendet werden, da die Verwendung eines SPLs einen direkte Informationshierarchie zwischen den interagierenden Komponenten des Layouts voraussetzt. Diese Hierarchie ist bei dem *Slideshow-UI* nicht gegeben.

Dennoch wurde das SPL zur Umsetzung einer Navigationsmöglichkeit implementiert. Dabei wurde es in wesentlichen Bestandteilen manuell verändert, was das Auftreten und somit den Einsatzzweck des *SPL* ändert. Ziel der Änderung war die Implementierung eines *Navigation Drawer*, der nicht permanent auftritt und im geöffneten Zustand parallel zum Canvas bedienbar ist.

Vor allem die Interaktionen zum Öffnen und Schließen des SPL wurden so stark verändert, dass der Nutzer das SPL nicht mehr als solches wahrnimmt, sondern als ein Navigationsmenü. Diese Vermutung wurde in der Verifikationsbefragung (Kap. 6.2.3) durch die wahrgenommene Erwartungskonformität und Benutzerfreundlichkeit bestätigt.

6.3. FAZIT

Selbstevaluation Durch die Verifikationsbefragung wurde deutlich, dass die Optimierung des Prototypen erfolgreich stattgefunden hat. Alle B

Während der Benutzung des *SlidingPanelLayout* im *Slideshow-UI* für Tablets ist momentan lediglich eine unsaubere Anpassung des auf der rechten Seite dargestellten *Viewpagers* mit *Toolbar* implementiert worden.

Das bedeutet, dass sich zwar die Slideshow-Timeline, während sie geöffnet oder geschlossen wird, in einem Bewegungsfluss befindet, allerdings wird die Breite des *Viewpagers* und

damit des dargestellten Canvas, sowie der *Toolbar* in zu großen und damit für den Nutzer erkennbaren Zeitintervallen angepasst. Der Vorgang erscheint stockend. Keiner der befragten Nutzer hat diese abrupte Adaption angemerkt oder kritisiert. Für den Workflow ist sie nicht ausschlaggebend. Dennoch widersprechen sich die sehr flüssige Bewegung des *SlidingsPaneLayout* und die stoßweise Anpassung des *Viewpagers* mit *Toolbar*.

In der Auswertung der Verifikationsbefragung (6.2.3) wurde deutlich, dass alle implementierten Bedienelemente in ihren Bewertungswerten mindestens einen Wert von 5,8 erreicht haben. Das entspricht einem Prozentsatz von 82,9% und zeigt somit eindeutig, dass die Testpersonen die getroffenen Entwurfsentscheidungen als intuitiv und benutzerfreundlich zu verwenden empfinden.

Dabei wurde Item SD25 nicht einbezogen, da die Entscheidung, welches der beiden Varianten des *Slideshow-UIs* für welches Mobilgerät verwendet wird, von jedem Nutzer individuell während der Laufzeit getroffen werden kann.

Resultierende Schnittstellen Bei der Weiterentwicklung des Prototypen ist zur Datenspeicherung eine Anbindung an ein Backend notwendig. Im parallel stattfindenden Komplexpraktikum am *Lehrstuhl für Rechnernetze* an der *TU Dresden* wurden bereits zugehörige Schnittstellen definiert.

Weiterhin gilt es, das bisher verwendete Datenformat weitestgehend zu übernehmen bzw. die notwendigen Attribute beizubehalten.

Primitive bestehen aus ID, einer Art (Viereck, Oval, Linie oder Pfeil), einem Startpunkt (x- und y-Koordinate), einem Endpunkt (x- und y-Koordinate), einer Farbe, einer Gültigkeit (relevant für eventuell Undo-Schritte) und einer Gruppe. Eventuell kann auch der Zeitpunkt der Erstellung gespeichert werden. Die zugehörigen Daten wurden im Prototyp im *Slideshow-UI* allerdings bisher nicht verwendet.

Text besteht aus einer ID, einem Inhaltstext, einem Startpunkt (x- und y-Koordinate), einer Farbe, einer Größe, einer Gültigkeit und einer Gruppe. Eventuell kann ebenfalls ein Zeitstempel ergänzt werden.

Graphiken bestehen aus einer ID, mehreren Primitiven und/oder Textelementen, einem erstellenden Nutzer, einem Titel und einer Fragestellung. Die Speicherung eines Zeitstempels ist hier zwingend notwendig.

7. ZUSAMMENFASSUNG UND AUSBLICK

Innerhalb dieser Bachelorarbeit wurden mehrere Varianten eines Prototypen von *Graphicuss* für Mobilgeräte konzipiert und anschließend implementiert. Die Android-Prototypen durchliefen zunächst einen Iterationsschritt, nach welchem in einer Evaluation die Auswahl einer Variante folgte.

Der gewählte Prototyp durchlief einen zweiten Iterationsschritt, in dem er durch konkrete Änderungen in der Implementierung, die sich aus der Evaluation ergaben, optimiert wurde. In der abschließenden Validierung konnte gezeigt werden, dass der Prototyp, sowie seine Bedienelemente als intuitiv empfunden werden und keine erheblichen Mängel mehr aufweisen.

Zusätzlich enthält der vertikale Prototyp ein Zeichentool, in dem Graphiken aus definierten Primitiven und/oder Text erstellt werden können. Die gezeichneten Graphiken können anschließend im zweifach evaluierten Slideshow-UI ihrer Erstellungshistorie schrittweise nachvollzogen werden. Das Interface verbindet Slider- und Slideshow-Metapher. Es wurden getrennte Layouts für Smartphone und Tablet entwickelt, um die Gegenbenheiten des Mobilgeräts bestmöglich zu nutzen.

Während der Erarbeitung wurden zudem einige für die Weiterentwicklung des Prototypen relevante Informationen gesammelt:

Das Slideshow-UI sollte zunächst auf die Eignung bei Verwendung der Mobilgeräte im Querformat geprüft werden.

Außerdem bedarf es einer Festlegung bezüglich der Maße des verwendeten Canvas. In diesem Prototyp wurde dabei die Größe des Canvas durch die des Mobilgerätes festgelegt.

Durch die Evaluation wurde außerdem deutlich, dass bei Betrachtung des Canvas eine Zoom-Funktion implementiert werden sollte, um umfangreiche Graphiken detailliert genug betrachten zu können.

Im Slideshow-UI für Tablet bedarf es außerdem einer flüssigeren Anpassung beim Öffnen bzw. Schließen der Timeline.

Es existierte außerdem der Bedarf, Änderungen im Canvas beim vor- bzw. zurückblättern zu verdeutlichen (AI1). Während der Evaluation gab es daher den Vorschlag, mittels einer Animation des hinzugefügten bzw. gelöschten Elementes die Anpassung des Canvas zu visualisieren.

eva 2: mitscrollen von timeline - menüleiste sollte wenn sie offen ist mit weiter nach unten

rutschen wurde implementiert

breitere schließen button wurde implementiert (muss noch gemacht werden)

LITERATURVERZEICHNIS

- [Apa17a] APACHE: *Android App Manifest File*. <https://developer.android.com/guide/topics/manifest/application-element.html#largeHeap>, 2017. – [letzter Zugriff 03.04.2018]
- [Apa17b] APACHE: *Android Developer*. <https://developer.android.com>, 2017. – [letzter Zugriff 03.04.2018]
- [Che16] CHEN, Kaijun: *Graphical Discussion System*, Technische Universität Dresden, Masterarbeit, Juli 2016
- [eMa16] EMARKETER: *Growth Trails Off in Western Europe Tablet Market*. <https://www.emarketer.com/Article/Growth-Trails-Off-Western-Europe-Tablet-Market/1013785/>, April 2016. – [letzter Zugriff 25.02.2018]
- [Goo17] GOOGLE: *Material Design, Guidelines*. <https://material.io/guidelines/>, 2017. – [letzter Zugriff 03.04.2018]
- [Ilj17] ILJASSOVA, Anastasia: *Nutzerschnittstellenentwurf für unterschiedliche Nutzungskontexte zur Visualisierung von Zeitinformationen innerhalb canvasbasierter Diskussionsbeiträge in Graphicuss*, Technische Universität Dresden, Bachelorarbeit, Oktober 2017
- [Kub16] KUBICA, Tommy: *Entwicklung eines Prototypen zur Auswahl und zum Einsatz technischer Werkzeuge/Werkzeugkombinationen in unterschiedlichen Lehrformen*, Technische Universität Dresden, Masterarbeit, August 2016
- [Maz96] MAZUR, Eric: *Peer Instruction: A User's Manual*. first. Pearson, 1996
- [MKH17] MAZUR, Eric ; KURZ, Günther ; HARTEN, Ulrich: *Peer Instruction: Interaktive Lehre praktisch umgesetzt*. Springer Spektrum, 2017
- [SB16] SARODNICK, Florian ; BRAU, Henning: *Methoden der Usability Evaluation, Wissenschaftliche Grundlagen und praktische Anwendungen*. 3. Auflage. Hogrefe, 2016
- [She17] SHEN, Tong: *Visualizing the temporal Creation Process of Graphical Discussion Contributions*, Technische Universität Dresden, Masterarbeit, Januar 2017
- [Typ17] TYPE/CODE: *Screen Sizes*. <http://screensiz.es/>, 2017. – [letzter Zugriff 24.03.2018]

A. ANHANG

A.1. VERWANDTE ARBEITEN

A.1.1. CHEN, KAIJUN: *GRAPHICAL DISCUSSION SYSTEM*

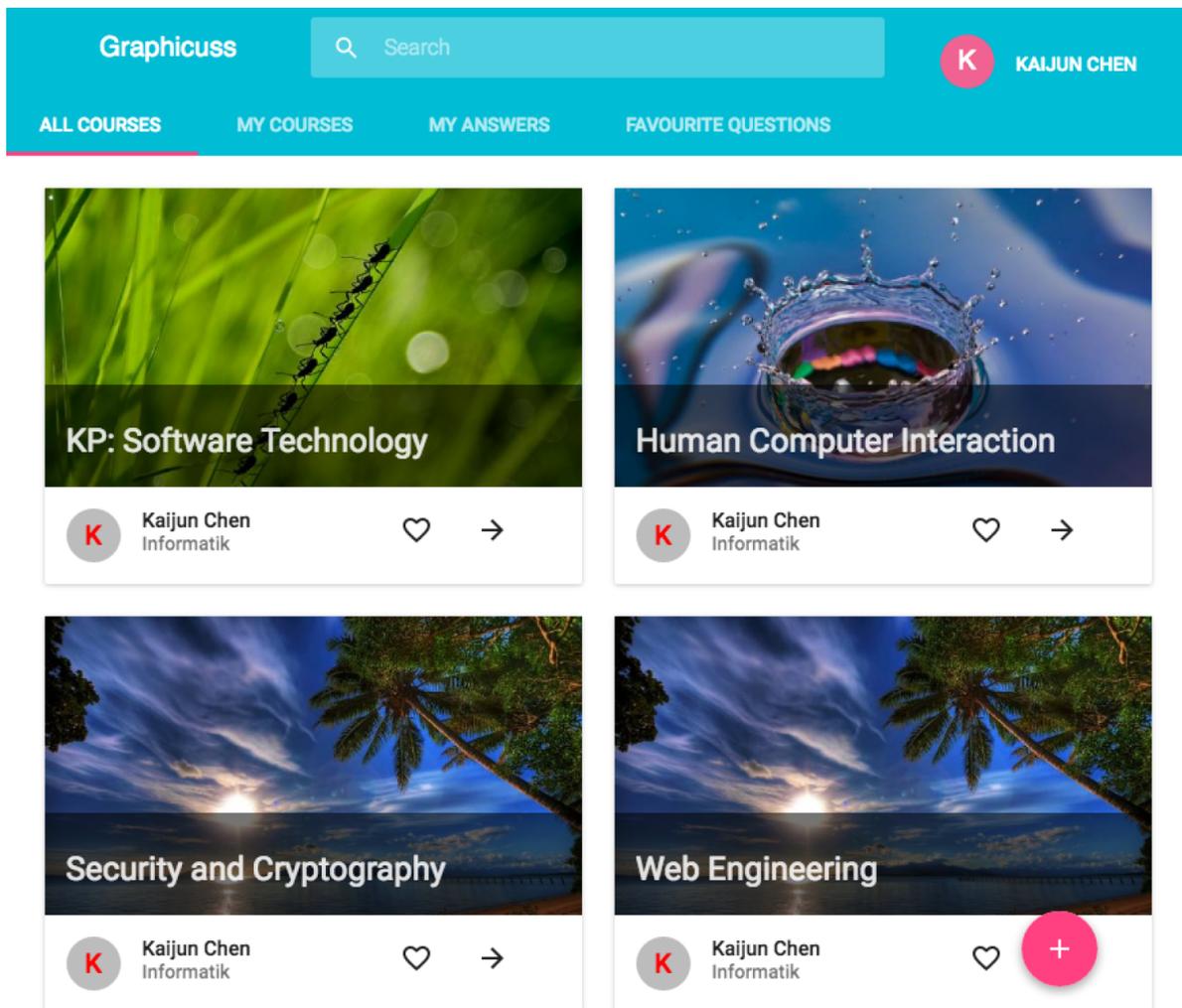


Abbildung A.1.: Kursübersicht, *Graphicuss* als Webanwendung [Che16]



Cloud Computing

BJI_LGVu

NEW QUESTION

Kaijun Chen Informatik 2016-07-25

Questions: 2 Course Detail

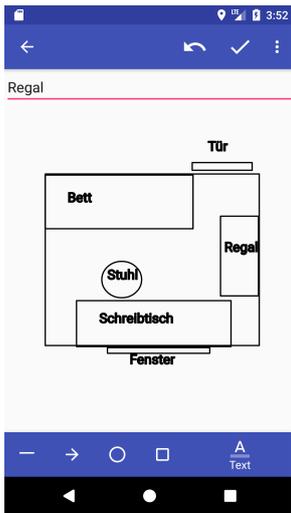
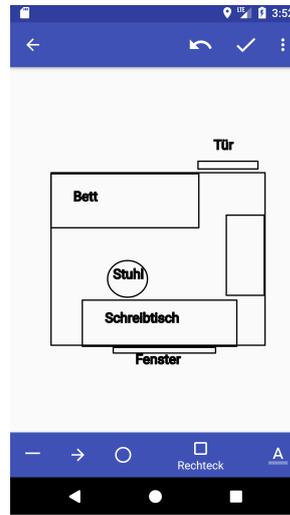
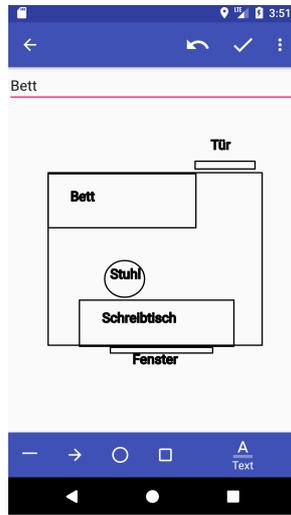
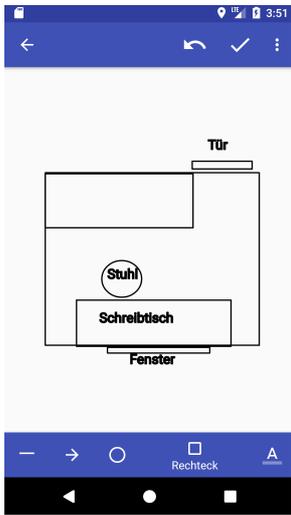
Title	Votes	Answers	Author	Date
Major could serving vendor?	1	0	Kaijun Chen	2016-07-20
Difference of SaaS Paas and laas?	0	0	Kaijun Chen	2016-07-20

Abbildung A.2.: Fragenübersicht, *Graphicuss* als Webanwendung [Che16]

A.2. ANALYSEBEFRAGUNG

A.2.1. BEISPIEL ANALYSEBEFRAGUNG





A.3. EVALUATION

A.3.1. EINLEITUNG UND LEGENDE

Einleitung Nutzerbefragung

In der heutigen Zeit sind mobile Geräte zu allgegenwärtigen Begleitern geworden. Sie erfüllen heute einen Nutzen weit über die Freizeit hinaus und spielen eine zunehmend wichtige Rolle im Bildungs-, sowie Arbeitsbereich.

Aus diesem Grund soll mit „**Graphicuss**“ eine App umgesetzt werden, in der man aus Lernzwecken Graphiken erstellen und sie mit anderen Nutzern in einem Forum teilen kann.

Insbesondere soll auch der **Verlauf** der gezeichneten Graphiken ersichtlich werden, d.h. Sie sollen nachvollziehen können, in welcher Reihenfolge die Primitive (Linien, Pfeile, Ovale, Rechtecke und Text) gezeichnet worden sind.

In dem zu testenden Prototypen wurde das Forum noch nicht komplett implementiert. Sie dienen lediglich zur Veranschaulichung einer gegebenen Graphik, die bereits im Forum gepostet wurde.

Zur Umsetzung dieses Zeitverlaufs werden Ihnen im Folgenden Test zwei Darstellungsmodi vorgestellt: das **Slideshow User Interface** und das **Video User Interface**.

Es werden Ihnen zu beiden währenddessen und danach Fragen gestellt, in denen Sie die beiden User Interfaces bewerten sollen.

Bitte treffen Sie die Bewertung ausschließlich darauf, wie gut Sie den Zeitverlauf nachvollziehen und mit der Software interagieren können und ignorieren Sie dabei etwaig auftretende Fehlermeldungen, schlechtere Auflösung, sowie persönliches Empfinden zur Farbgestaltung.

Die Benutzung erfolgt außerdem ausschließlich im Hochformat, da die Beispiele dafür ausgelegt wurden.

Weiterhin wird darauf hingewiesen, dass das Gespräch und Ihre Bewegungen auf den mobilen Geräten aufgezeichnet und später anonym ausgewertet werden.

Stellen Sie sich nun folgende **Ausgangssituation** vor:

Ein anderer Nutzer hat bereits eine Frage in diesem Forum gepostet. Darin wird die Einrichtung eines Zimmers diskutiert. Er erwähnt, dass er noch einen Kleiderschrank besitzt, der die gleichen Maße, wie der gezeigte Schreibtisch besitzt und bei der bisherigen Einrichtung keinen Platz findet. Der Nutzer bittet um Vorschläge für eine anderweitige Gestaltung.

Versuchen Sie nun, seine Gedankengänge innerhalb der Applikation mit beiden User Interfaces nachzuvollziehen.

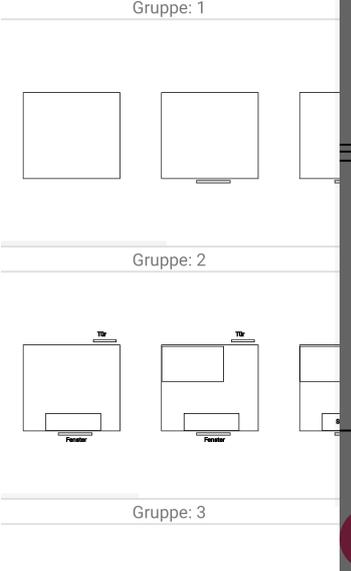
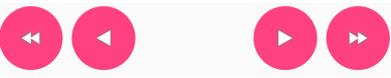
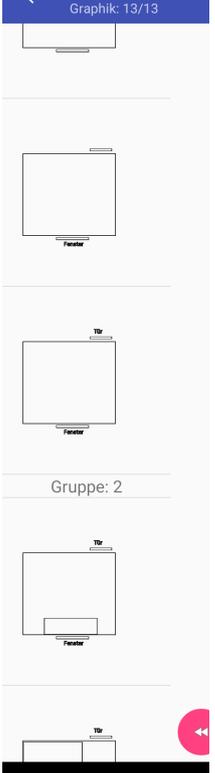
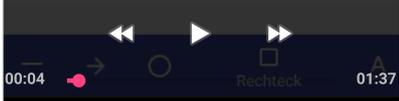
Jede Befragung hilft dabei, den Prototypen nutzerfreundlicher zu gestalten.

Bei dieser Befragung geht es vor allem um **Ihre eigene Meinung**. Es gibt weder richtige, noch falsche Antworten.

Nehmen Sie sich bitte so viel Zeit, wie Sie benötigen.

Natürlich können Sie **jederzeit Fragen stellen**.

Vielen Dank für Ihre Teilnahme

Bezeichnung	Bedeutung	Bezeichnung	Bedeutung
MF	Mehrfachantworten möglich	Slideshow-Menü (Smartphone)	 <p>Gruppe: 1</p> <p>Gruppe: 2</p> <p>Gruppe: 3</p>
UI	User Interface (Nutzerschnittstelle/-oberfläche)		
Slideshow-Buttons			
Flag-Buttons		Slideshow-Timeline (Tablet)	 <p>Graphik: 13/13</p> <p>Gruppe: 2</p>
Mediaplayer			
Mediaplayer-Buttons			

A.3.2. FRAGEBOGEN SLIDESHOW-UI FÜR SMARTPHONE I

Slideshow User Interface Smartphone

Nr	How				
SA01	Welche Bedienelemente haben Sie genutzt, um zwischen benachbarten Graphiken zu navigieren? (MF)	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Menü	
SA02	Wie haben Sie bevorzugt zwischen benachbarten Graphiken navigiert?	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Menü	<input type="radio"/> keine Angabe
SA03	Welche Bedienelemente haben Sie genutzt, um zwischen Gruppen zu navigieren? (MF)	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Menü	<input type="radio"/> keine
SA04	Wie haben Sie bevorzugt zwischen Gruppen navigiert?	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Menü	<input type="radio"/> keine Angabe
SA05	Welche Bedienmöglichkeiten halten Sie für notwendig? (MF)	<input type="radio"/> Wischgesten	<input type="radio"/> Buttons	<input type="radio"/> Menü	<input type="radio"/> andere
SA06	Wie haben Sie das Menü geöffnet? (MF)	<input type="radio"/> Wischen	<input type="radio"/> Menü-Button	<input type="radio"/> gar nicht	

1 - trifft gar nicht zu
7 - trifft vollkommen zu

Nr	Slideshow-Buttons	1	2	3	4	5	6	7
SD01	Die Slideshow-Buttons sind richtig positioniert.							
SD02	Alle Slideshow-Buttons sind notwendig.							
SD03	Die Slideshow-Buttons sind eindeutig.							
SD04	Die Optik der Slideshow-Buttons ist ideal. (Größe, Farbe, Icon)							

Nr	Slideshow-Menü	1	2	3	4	5	6	7
SD05	Das Slideshow-Menü ist richtig positioniert.							
SD06	Das Slideshow-Menü ist notwendig.							
SD07	Mir war bewusst, dass ich mithilfe des Slideshow-Menüs navigieren kann.							
SD08	Die Aufteilung des Slideshow-Menüs ist übersichtlich.							
SD09	Die Inhalte im Slideshow-Menü sind gut genug zu erkennen.							
SD10	Es war offensichtlich, dass ein (Slideshow-) Menü existiert.							
SD11	Es fiel mir leicht, das Slideshow-Menü zu öffnen.							
SD12	Es fiel mir nicht leicht, das Slideshow-Menü zu schließen.							
SD13	Ich finde es intuitiv, mit horizontalem Wischen ein Menü zu öffnen.							

Nr	Allgemein	1	2	3	4	5	6	7
SD14	Die Reaktion auf Wischgesten verlief erwartungskonform.							
SD15	Es war kompliziert , den gesuchten Zeichenschritt zu finden.							

Nr	Allgemein	1	2	3	4	5	6	7
SD16	Ich konnte die Reaktion der Anwendung auf meine Eingabe zu jedem Zeitpunkt genau einschätzen.							
SD17	Ich finde das Interface ist für Smartphones geeignet .							

A.3.3. FRAGEBOGEN SLIDESHOW-UI FÜR TABLET I

Slideshow User Interface Tablet

Nr	How				
SA01	Welche Bedienelemente haben Sie genutzt, um zwischen benachbarten Graphiken zu navigieren? (MF)	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Slideshow-Timeline	
SA02	Wie haben Sie bevorzugt zwischen benachbarten Graphiken navigiert?	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Slideshow-Timeline	<input type="radio"/> keine Angabe
SA03	Welche Bedienelemente haben Sie genutzt, um zwischen Gruppen zu navigieren? (MF)	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Slideshow-Timeline	<input type="radio"/> keine
SA04	Wie haben Sie bevorzugt zwischen Gruppen navigiert?	<input type="radio"/> Wischen	<input type="radio"/> Buttons	<input type="radio"/> Slideshow-Timeline	<input type="radio"/> keine Angabe
SA05	Welche Bedienmöglichkeiten halten Sie für notwendig? (MF)	<input type="radio"/> Wischgesten	<input type="radio"/> Buttons	<input type="radio"/> Slideshow-Timeline	<input type="radio"/> andere

1 - trifft gar nicht zu
7 - trifft vollkommen zu

Nr	Slideshow-Buttons	1	2	3	4	5	6	7
SD01	Die Slideshow-Buttons sind richtig positioniert.							
SD02	Alle Slideshow-Buttons sind notwendig.							
SD03	Die Slideshow-Buttons sind eindeutig.							
SD04	Die Optik der Slideshow-Buttons ist ideal. (Größe, Farbe, Icon)							

Nr	Slideshow-Timeline	1	2	3	4	5	6	7
SD19	Die Slideshow-Timeline ist notwendig.							
SD20	Mir war bewusst, dass ich mithilfe der Slideshow-Timeline navigieren kann.							
SD21	Die Aufteilung des Slideshow-Timeline ist übersichtlich.							
SD22	Die Inhalte der Slideshow-Timeline sind gut genug zu erkennen.							
SD23	Es fiel mir nicht leicht, die Slideshow-Timeline zu bedienen.							

Nr	Allgemein	1	2	3	4	5	6	7
SD14	Die Reaktion auf Wischgesten verlief erwartungskonform.							
SD15	Es war kompliziert , den gesuchten Zeichenschritt zu finden.							
SD16	Ich konnte die Reaktion der Anwendung auf meine Eingabe zu jedem Zeitpunkt genau einschätzen.							
SD24	Ich finde das Interface für Tablets nicht geeignet .							
SD25	Ich empfinde ein eigenes Interface für Tablets als notwendig ,							

A.3.4. FRAGEBOGEN VIDEOPLAYER-UI FÜR SMARTPHONE UND TABLET I

Videoplayer User Interface Smartphone

How

VA01	Welche Bedienelemente haben Sie genutzt, um zwischen benachbarten Graphiken zu navigieren? (MF)	<input type="checkbox"/> Wischen <input type="checkbox"/> Progressbar	<input type="checkbox"/> Buttons <input type="checkbox"/> Mediaplayer	<input type="checkbox"/> Buttons <input type="checkbox"/> Flaggen	<input type="checkbox"/> Warten	
VA02	Wie haben Sie bevorzugt zwischen benachbarten Graphiken navigiert?	<input type="checkbox"/> Wischen <input type="checkbox"/> Progressbar	<input type="checkbox"/> Buttons <input type="checkbox"/> Mediaplayer	<input type="checkbox"/> Flag-Buttons	<input type="checkbox"/> Warten	<input type="checkbox"/> keine Angabe
VA03	Welche Bedienelemente haben Sie genutzt, um zwischen Gruppen zu navigieren? (MF)	<input type="checkbox"/> Wischen <input type="checkbox"/> Progressbar	<input type="checkbox"/> Buttons <input type="checkbox"/> Mediaplayer	<input type="checkbox"/> Flag-Buttons	<input type="checkbox"/> Warten	<input type="checkbox"/> keine Angabe
VA04	Wie haben Sie bevorzugt zwischen Gruppen navigiert?	<input type="checkbox"/> Wischen <input type="checkbox"/> Progressbar	<input type="checkbox"/> Buttons <input type="checkbox"/> Mediaplayer	<input type="checkbox"/> Flag-Buttons	<input type="checkbox"/> Warten	<input type="checkbox"/> keine Angabe
VA05	Welche Bedienmöglichkeiten halten Sie für notwendig? (MF)	<input type="checkbox"/> Wischen <input type="checkbox"/> Progressbar	<input type="checkbox"/> Buttons <input type="checkbox"/> Mediaplayer	<input type="checkbox"/> Flag-Buttons		
VA06	Haben Sie Teile des Videos übersprungen?	<input type="checkbox"/> Ja	<input type="checkbox"/> Nein			
VA07	Was wäre die für Sie sinnvollste Belegung für die Fastforward/Rewind- Buttons im Mediaplayer?	<input type="checkbox"/> feste Zeit: _____s	<input type="checkbox"/> feste Anzahl: ____ Schritte			

1 - trifft gar nicht zu
7 - trifft vollkommen zu

Flag-Buttons

1 2 3 4 5 6 7

VD01	Die Flag-Buttons sind richtig positioniert.							
VD02	Die Flag-Buttons sind notwendig.							
VD03	Die Flag-Buttons sind selbsterklärend.							
VD04	Die Optik der Flag-Buttons ist ideal. (Größe, Farbe, Icon)							

Mediaplayer

1 2 3 4 5 6 7

VD05	Der Mediaplayer ist richtig positioniert.							
VD06	Mir war bewusst, wie ich mithilfe des Mediaplayers navigieren kann. (Bedeutung der Mediaplayer-Buttons)							
VD07	Der Mediaplayer sollte immer eingeblendet werden.							

Allgemein

1 2 3 4 5 6 7

VD08	Die Inhalte sind gut zu erkennen .							
VD09	Die zeitlichen Details (Zeichengeschwindigkeit, Wartezeiten, Dauer der Erstellung der Graphik) bringen einen Mehrwert .							
VD10	Es ist nicht hilfreich , den ganzen Zeichenverlauf jedes Objektes zu sehen. (Startpunkt bis Endpunkt Primitiv, Eingabe von Text)							
VD11	Das Video hatte die richtige Geschwindigkeit .							

Allgemein		1	2	3	4	5	6	7
VD12	Es war kompliziert , den gesuchten Zeichenschritt zu finden.							
VD13	Ich konnte die Reaktion der Anwendung auf meine Eingabe zu jedem Zeitpunkt genau einschätzen.							
VD14	Ich finde das Interface für Smartphones geeignet .							
VD15	Ich finde das Interface für Tablets geeignet .							

A.3.5. FRAGEBOGEN VERGLEICH UND PERSÖNLICHES I

Nr	Vergleich der Varianten des Slideshow User Interfaces	Smart phone	Tablet	nein
Vg1	Bevorzugen Sie eine der Varianten für beide Arten von Mobilgeräten?			
Vg2	Empfinden Sie eine Variante als ungeeignet ? (MF)			

Nr	Vergleich der User Interfaces	Slide show	Video player	keine Angabe
Vg3	Welches UI hat für Sie mehr Informationsgehalt ?			
Vg4	Welches UI war für Sie leichter zu bedienen ?			
Vg5	Erschien ein UI für Sie überladen oder unübersichtlich ? (MF)			
Vg6	Welches Konzept hat für Sie das meiste Potential , wenn Sie Ihre Verbesserungsvorschläge einbeziehen?			
Vg7	Welches UI würden Sie für die vollständige Anwendung bevorzugen ?			

Einordnung in Personengruppen (Angabe freiwillig)

GruppenID	Geschlecht	Altersgruppe	Berufsfeld	
IT1	m	20 - 39	IT	<input type="radio"/>
IT2	w	20 - 39	IT	<input type="radio"/>
IT3	m	40 - 60	IT	<input type="radio"/>
IT4	w	40 - 60	IT	<input type="radio"/>
DI1	m	20 - 39	Anderes	<input type="radio"/>
DI2	w	20 - 39	Anderes	<input type="radio"/>
DI3	m	40 - 60	Anderes	<input type="radio"/>
DI4	w	40 - 60	Anderes	<input type="radio"/>

A.4. VERIFIKATIONSBEFRAGUNG

A.4.1. FRAGEBOGEN SLIDESHOW-UI FÜR SMARTPHONE II

Slideshow User Interface Smartphone

1 - trifft gar nicht zu
7 - trifft vollkommen zu

Nr	Slideshow-Buttons	1	2	3	4	5	6	7
SD01	Die Slideshow-Buttons sind richtig positioniert.							
SD02	Alle Slideshow-Buttons sind notwendig.							
SD03	Die Slideshow-Buttons sind eindeutig.							
SD04	Die Optik der Slideshow-Buttons ist ideal. (Größe, Farbe, Icon)							
Nr	Slideshow-Menü	1	2	3	4	5	6	7
SD05	Das Slideshow-Menü ist richtig positioniert.							
SD06	Das Slideshow-Menü ist notwendig.							
SD07	Mir war bewusst, dass ich mithilfe des Slideshow-Menüs navigieren kann.							
SD08	Die Aufteilung des Slideshow-Menüs ist übersichtlich.							
SD09	Die Inhalte im Slideshow-Menü sind gut genug zu erkennen.							
SD10	Es war offensichtlich, dass ein (Slideshow-)Menü existiert.							
SD11	Es fiel mir leicht, das Slideshow-Menü zu öffnen.							
SD12	Es fiel mir nicht leicht, das Slideshow-Menü zu schließen.							
SD13	Ich finde es intuitiv, mit horizontalem Wischen ein Menü zu öffnen.							
Nr	Allgemein	1	2	3	4	5	6	7
SD14	Die Reaktion auf Wischgesten verlief erwartungskonform.							
SD15	Es war kompliziert , den gesuchten Zeichenschritt zu finden.							
SD16	Ich konnte die Reaktion der Anwendung auf meine Eingabe zu jedem Zeitpunkt genau einschätzen.							
SD17	Ich finde das Interface ist für Smartphones geeignet .							

A.4.2. FRAGEBOGEN SLIDESHOW-UI FÜR TABLET II

Slideshow User Interface Tablet

1 - trifft gar nicht zu
7 - trifft vollkommen zu

Nr	Slideshow-Buttons	1	2	3	4	5	6	7
SD01	Die Slideshow-Buttons sind richtig positioniert.							
SD02	Alle Slideshow-Buttons sind notwendig.							
SD03	Die Slideshow-Buttons sind eindeutig.							
SD04	Die Optik der Slideshow-Buttons ist ideal. (Größe, Farbe, Icon)							

Nr	Slideshow-Timeline	1	2	3	4	5	6	7
SD19	Die Slideshow-Timeline ist notwendig.							
SD20	Mir war bewusst, dass ich mithilfe der Slideshow-Timeline navigieren kann.							
SD21	Die Aufteilung des Slideshow-Timeline ist übersichtlich.							
SD22	Die Inhalte der Slideshow-Timeline sind gut genug zu erkennen.							
SD23	Es fiel mir nicht leicht, die Slideshow-Timeline zu bedienen.							
SD26	Es war offensichtlich, dass eine Slideshow-Timeline existiert.							
SD27	Es fiel mir leicht, die Slideshow-Timeline zu öffnen.							
SD28	Es fiel mir nicht leicht, die Slideshow-Timeline zu schließen.							

Nr	Allgemein	1	2	3	4	5	6	7
SD14	Die Reaktion auf Wischgesten verlief erwartungskonform.							
SD15	Es war kompliziert , den gesuchten Zeichenschritt zu finden.							
SD16	Ich konnte die Reaktion der Anwendung auf meine Eingabe zu jedem Zeitpunkt genau einschätzen.							
SD24	Ich finde das Interface für Tablets nicht geeignet .							
SD25	Ich empfinde ein eigenes Interface für Tablets als notwendig ,							