

# 4. Wireflow und Key Screen

Angelehnt an Artikel von Mike Angeles (mike@balsamiq.com) und <https://www.collaborative-uxdesign.com/methods/konzept-wireflows>

## Einführung

Wireflows sind eine Kombination aus Wireframes (grafische Darstellungen von Benutzeroberflächen) und User Flows (sequenzielle Beschreibung von Benutzerinteraktionen), die verwendet werden, um das Zusammenspiel von Seiten und Interaktionen in einer Anwendung zu visualisieren und zu dokumentieren. Sie bieten einen detaillierten Überblick über den Ablauf der Benutzerinteraktionen und dienen als Grundlage für die Entwicklung und das Design von Benutzeroberflächen.

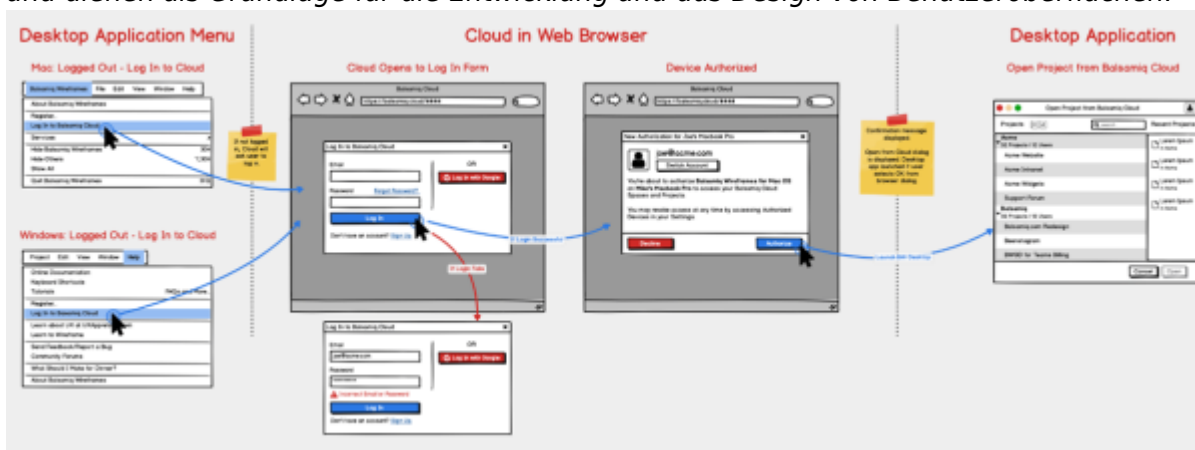


Abb. 6-5: Beispiel eines Wireflows

## Grundlegende Elemente von Wireflows

Mit einem Wireflow können wir die in den Szenarien beschriebenen Interaktionen in Form von miteinander verknüpften Wireframes visualisieren. Abgebildet wird der Fluss der Szenarios. <br/> Durch einfaches Hinzufügen von Pfeilen und Anmerkungen zwischen den Wireframes (Screens) werden die Pfade angezeigt, die ein Benutzer bei der Verwendung einer Anwendung einschlagen kann. Diese kleinen Änderungen an Wireframes kommunizieren die visuellen Änderungen an der Mensch-Maschinen-Schnittstelle und beschreiben gleichzeitig Zustandsänderungen und Sequenzen in der Benutzerinteraktion.

### Wireframes

Wireframes (auch als Key Screens bezeichnet) eignen sich gut für die Darstellung statischer Schnittstellen.

Die Daten und das Layout einer Schnittstelle ändern sich jedoch häufig, wenn Benutzer mit einem Produkt interagieren, so dass ein einzelnes statisches Wireframe die Komplexität dieser Änderungen möglicherweise nicht gut darstellt.

Das folgende Beispiel zeigt eine Seite für ein Produkt auf einer Shopping-Website.

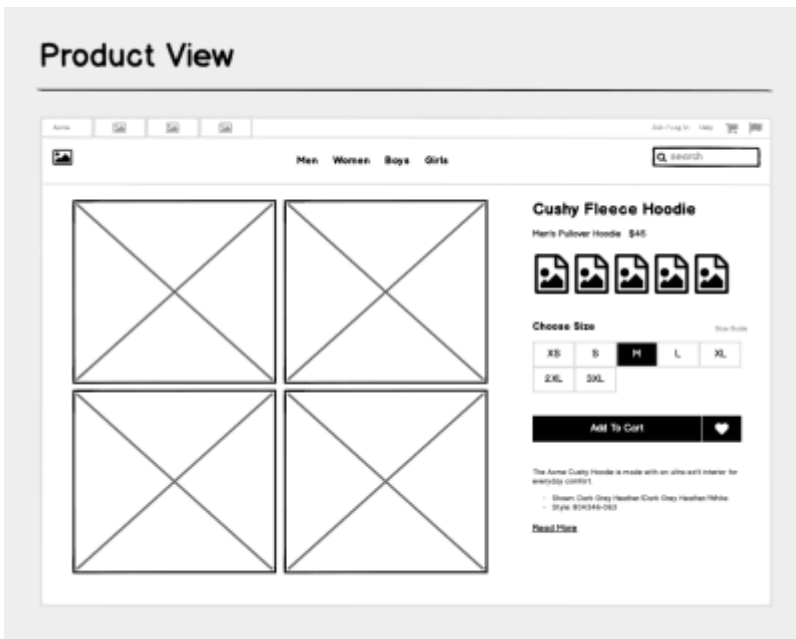


Abb. 6-6: Wireframe einer Anwendung

Dieser Wireframe vermittelt eine gute Vorstellung davon, welche Elemente auf den Produktseiten eines Shops erscheinen und wo sie angeordnet werden sollten. Einige dieser Oberflächenelemente lösen jedoch Aktionen aus. Zum Warenkorb hinzufügen (Add to Cart) ist hier ein offensichtlicher Auslöser. Es muss daher explizit festgelegt werden, was passiert, wenn der Benutzer diese Aktion auslöst. Um dies zu tun, könnte ein Produktdesigner typischerweise jeden geänderten Zustand drahtlos darstellen, wenn der Benutzer die Kaufabwicklung durchläuft.

### User Flow (Nutzerfluss)

Aufgabenabläufe sind nützlich, um zu entwerfen, wie ein Benutzer eine Aufgabe erledigen wird. Sie werden in der Regel als lineare Abfolge von Schritten dargestellt und können auf einer hohen Ebene entworfen werden oder sehr detailliert sein, indem eine Aufgabe in Teilaufgaben unterteilt wird. Sie können das Ergebnis einer Aufgabenanalyse sein, bei der beobachtet wird, wie Benutzer Aufgaben erledigen.

### Wireflows

Der Begriff Wireflow wurde von der Nielsen Norman Group geprägt, nachdem sie beobachtet hatte, wie sich die Praxis in diesem Bereich entwickelte.

Wireflows zeigen visuell, wie sich Teile der Benutzeroberfläche verändern, wenn der Benutzer mit der Anwendung interagiert. Zusätzlich können Anmerkungen hinzugefügt werden, um Faktoren wie die Übertragung von Daten innerhalb des Systems anzuzeigen.

Wireflows können einfach als eine Erweiterung der Wireframing-Praxis betrachtet werden, um Benutzerflussinformationen direkt in das Schnittstellendesign einzufügen.

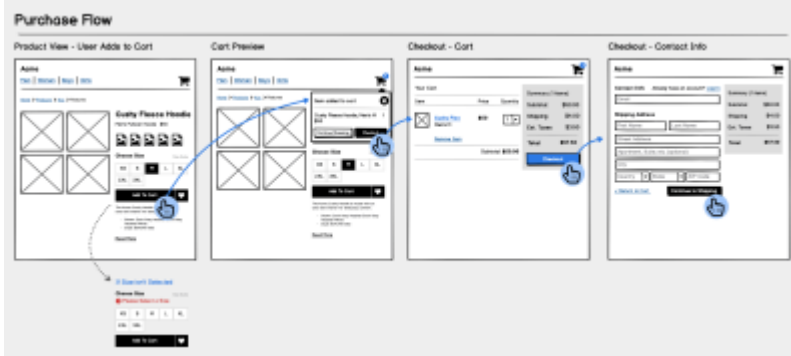


Abb. 6-9: Veranschaulichung des Ablaufs anhand eines Wireflows

Es können auch komplexere Beispiele für Wireflows erstellt werden, um zu zeigen, wie eine Vielzahl von Aktionen innerhalb einer Schnittstelle Änderungen oder Systemreaktionen in einer dynamischen Schnittstelle auslösen können.

Die komplexesten Beispiele können Flussdiagrammen ähneln und zeigen die Art von Entscheidungsverzweigungen, Backend-Prozessen und Datenbewegungen, wie sie in detaillierten Flussdiagrammen erwartet werden.

## Wie man einen Wireflow erstellt.

Es ist üblich, ganze Schlüsselbildschirme (Key Screen) zu reproduzieren und Zustandsänderungen zu zeigen.

Die Erstellung von Wireflows kann in 3 einfachen Schritten durchgeführt werden:

1. Beschreibender Text des Ablaufs oder User Flow
2. Definieren von Key Screens
3. Screens verbinden

### 1. Beschreibender Text

Basierend auf einem Szenario wird eine mögliche Lösung in Worte gefasst und ein einfacher Aufgabenablauf festgehalten.

### 2. Definieren von Key Screens

Als Nächstes werden die wichtigsten Bildschirme (Key Screens) der Schnittstelle erstellt (basierend auf Lösungsskizzen aus der Ideation Phase). Eine gute Möglichkeit, dies zu tun, ist es, in Form von Start- und Endpunkten zu denken. Im Beispiel der Aufgabe „Kauf eines Hoodies“ ist der Startpunkt die Produktseite und der Endpunkt die Bestätigungsseite der WEB-Anwendung.

Ziel ist es, Bildschirme zu identifizieren, bei denen die Schnittstelle ihren Zustand ändert, z. B. wenn Daten während eines Prozesses wie dem Auschecken aus einem Geschäft dynamisch aktualisiert werden. Es könnte ein Formular verwendet werden, das nach und nach mehr Eingaben erfordert, z.B. Kontaktinformationen, Versandinformationen und Rechnungsinformationen in einer Ansicht. Jedes dieser Formulare müsste entworfen werden.

### 3. Bildschirme verbinden

Als Nächstes wird die Verbindung der wichtigsten Bildschirme (Key Screens) skizziert. Dazu werden Pfeile zwischen die Kästchen gezeichnet, um den Raum zwischen den Key Screens zu definieren und den Benutzer durch seine Aufgabe voranzubringen. Es können Entscheidungspunkte eingefügt werden, um zu zeigen, was in verschiedenen Fällen passiert, wenn der Benutzer eine Option wählt, und was passiert, wenn er eine andere wählt.

---

### Beispiel 14: App für Katzenfotos

Nehmen wir an, wir entwerfen eine App zur gemeinsamen Nutzung von Katzenfotos und möchten die Anmeldung und das erste Benutzererlebnis gestalten. Dazu kann folgender Flow skizziert werden:

1. Aufgabe die durchzuführen sind
  - Anmeldung: Der Benutzer meldet sich an und legt einen Benutzernamen, eine E-Mail und ein Passwort fest.
  - Erster Beitrag: Die erste Aufgabe des Nutzers ist es, sein erstes Foto direkt nach der Anmeldebestätigung zu machen.
2. Key Screens
  - Unregistrierte Startseite

- Anmeldungscreens
- Anmeldebestätigung
- Anmeldungscreens
- Erster Beitrag
- Feed (Bestätigung)

### 3. Wireflow

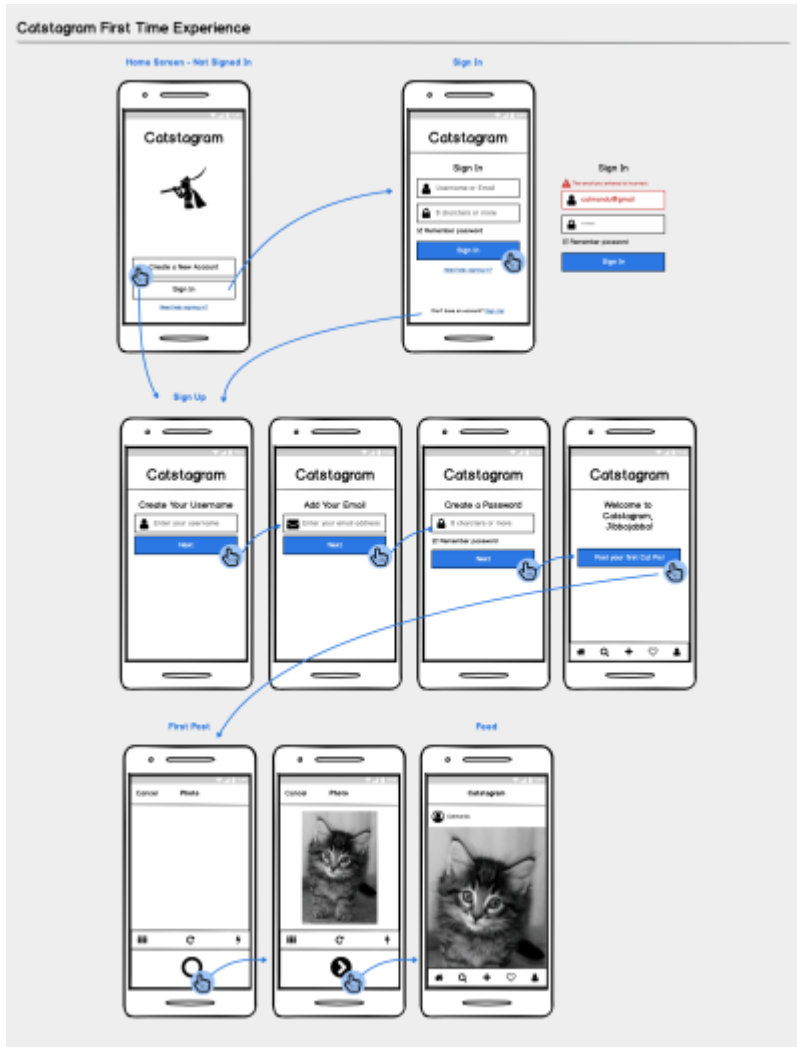


Abb. 6-10: Illustration mit Wireframes zur Veranschaulichung des Ablaufs

## Ergebnis und Schlussfolgerung

Ein Wireflow ist nicht mehr lösungsneutral. Er spiegelt erste Designentscheidungen insbesondere im Bezug zur Informationsarchitektur und den verwendeten Patterns wider: Welche Interaktionskontexte gibt es, wie wird navigiert, welche Interaktionen sind notwendig?

Wie wir im obigen Vergleich gesehen haben, bieten uns Wireflows eine Reihe von Vorteilen gegenüber der Verwendung jeder einzelnen dieser Entwurfstechniken. Der Hauptgrund für die Verwendung von Wireflows besteht darin, dass Sie als Produktdesigner die Möglichkeit haben, die Schnittstelle und den Ablauf auf die effektivste Art und Weise mit einem einzigen Design-Artefakt visuell zu kommunizieren.

(1) Vergleiche ISO 9241: Aufgabenangemessenheit

- (2) Vergleiche ISO 9241: Fehlerrobustheit (-toleranz)
- (3) Vergleiche ISO 9241: Steuerbarkeit

### Werkzeuge

Das Erstellen von Key Screens und Wireflows kann am einfachsten mit Papier und Bleistift realisiert werden. Um einen ersten Eindruck eines Konzepts zu erhalten, genügt dies vollauf.

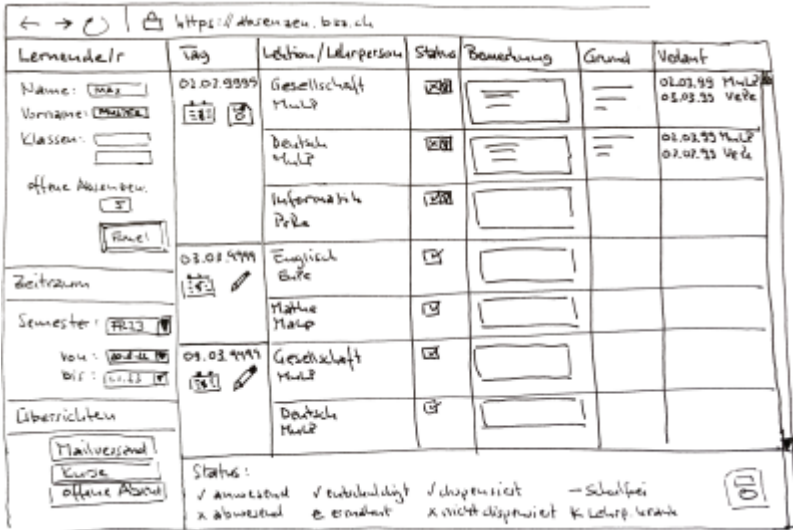


Abb. 6-11: Handskizze eines Key Screen

Wer sich mit skizzieren schwer tut, kann auch Anwendungen wie Visio, Balsamiq usw. nutzen.

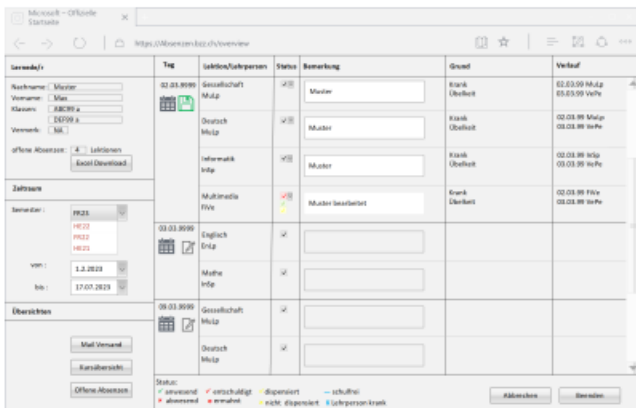


Abb. 6-12: Key Screen mit Visio erstellt

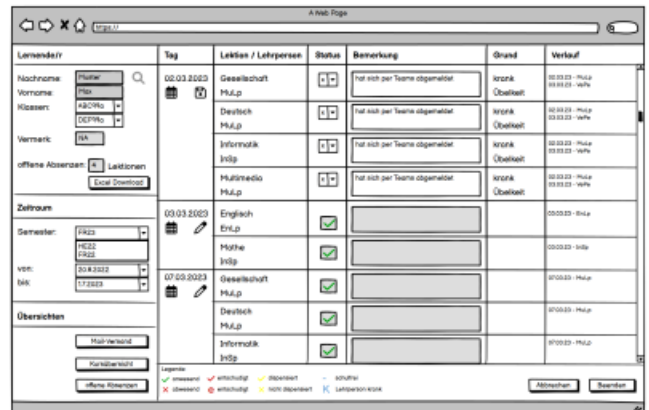


Abb. 6-13: Key Screen mit Balsamiq erstellt

Selbst mit einer Anwendung wie z.B. Excel lassen sich Key Screens entwerfen. Man muss dabei gut Überlegen, wie die Zeilenhöhen und Spaltenbreiten zu wählen sind.

Elemente wie Pulldown, Scrollbar oder gewisse Icons können per copy-paste aus z.B. Visio eingefügt werden.

Lernender	Tag	Lektion/Lehrperson	Status	Bemerkung	Grund	Verlauf
Max Master	02.03.9999	Gesellschaft MuLp	<span style="color:red">■</span>	per Teams abgemeldet	krank Überkeit	02.03.99 - MuLp
Klassen: ABC22e, XY222e		Deutsch MuLp	<span style="color:red">■</span>		krank Überkeit	02.03.99 - MuLp
offene Absenzen: 5 Excel Export		Informatik InSp	<span style="color:red">■</span>	Arztzeugnis nachgereicht (Prüfung)	krank Überkeit	02.03.99 - MuLp 05.03.99 - Verfe
Zeitraum	03.03.9999	Englisch LpEn	<span style="color:green">■</span>			05.03.99 - LpEn
PR23 30.01.2023 17.07.2023		Mathe InSp	<span style="color:green">■</span>			02.03.99 - InSp
Überichten	03.03.9999	Gesellschaft MuLp	<span style="color:green">■</span>			09.03.99 - MuLp
Mitversand		Deutsch MuLp	<span style="color:green">■</span>			09.03.99 - MuLp
Kurse						
offene Absenzen						
Status: <span style="color:green">■</span> anwesend <span style="color:red">■</span> entschuldig <span style="color:yellow">■</span> dispensiert <span style="color:blue">■</span> schulfrei <span style="color:red">■</span> abwesend <span style="color:red">■</span> erkrankt <span style="color:yellow">■</span> nicht dispensiert <span style="color:blue">■</span> Lehrpers. Krank						
						Abbrechen    Beenden

Abb. 6-14: Key Screen mit Excel erstellt

Eine günstige Alternative für Smartphone Screens bietet auch das Miro-Board an.



Abb. 6-15: Library des Miro-Boards mit Wireframes für Smartphones

## Wireflow

Ein Wireflow zeigt den Ablauf, den eine Anwendung bei unterschiedlichen Aktionen nehmen kann. Um einen Wireflow zu skizzieren, ist es sinnvoll, wenn die jeweiligen Key Screens in einer minimierten Form präsentiert werden. Alle Elemente die nicht oder nur wenig von Bedeutung für den Ablauf der Anwendung sind, können als Linie dargestellt werden. Dadurch ergibt sich ein kompakteres Bild.

Beispiel 15: Wireflow zu Absenztool

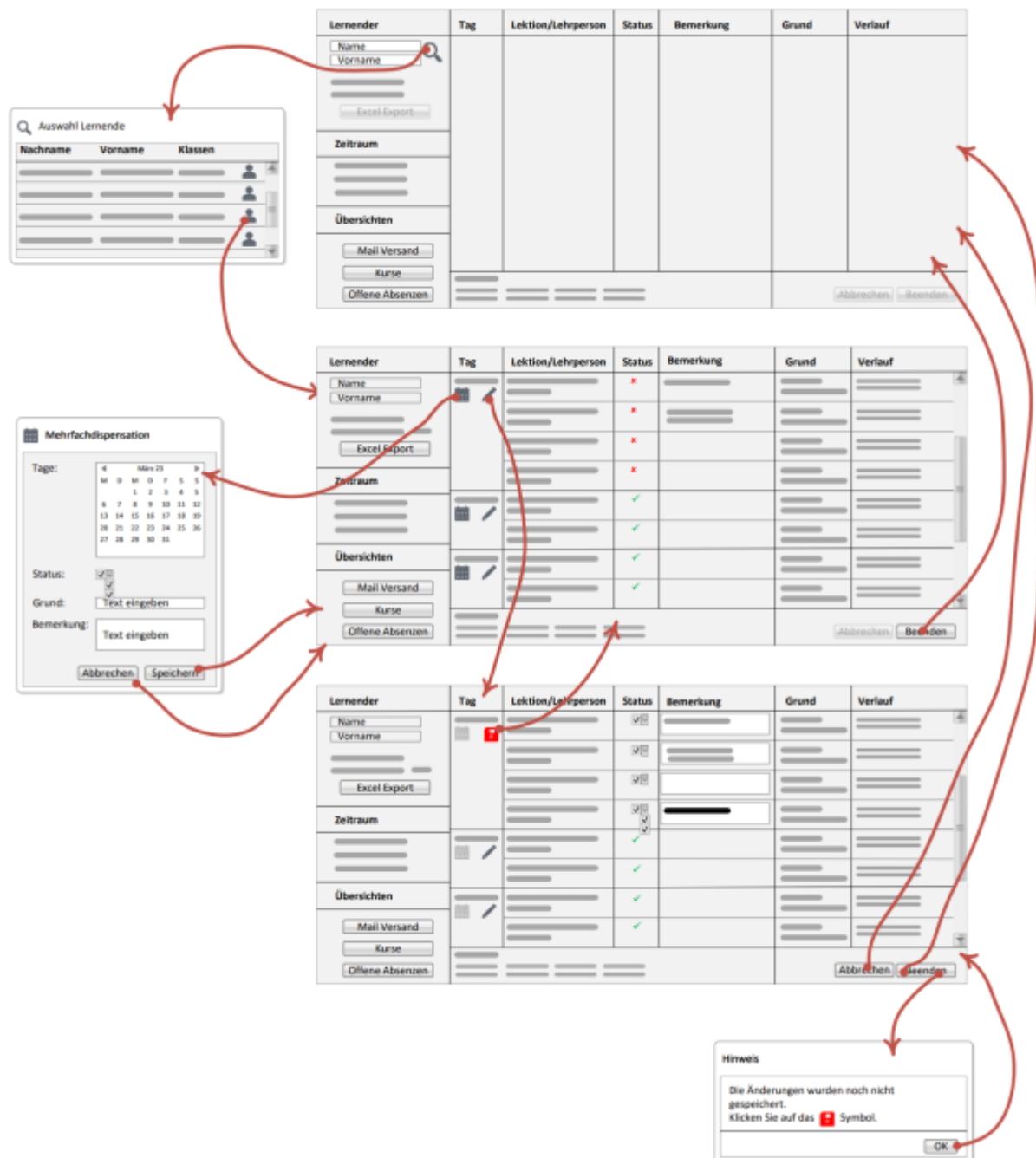


Abb. 6-16: Wireflow einer Anwendung

Hier findet sich auch ein Beispiel für die Lernförderlichkeit einer Anwendung, in dem der Benutzer über die Bedeutung des Disketten-Symbols aufgeklärt wird.



© René Probst

From: <https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link: [https://wiki.bzz.ch/modul/m322/learningunits/lu06/theorie/lu6-kapitel\\_4?rev=1741957600](https://wiki.bzz.ch/modul/m322/learningunits/lu06/theorie/lu6-kapitel_4?rev=1741957600)

Last update: 2025/03/14 14:06

