

# Audit 3

## KidSecure

von Jaqueline Ganz, Anouk Martinez, Phithaya Khamkaew

# Inhalt

- Proof of Concepts
- Wireframes
- Symbole
- Navigation innerhalb der App

# Iteration

## • Anforderungen

### Anforderungsermittlung

#### Funktionale Anforderungen

Das System muss den Schüler\*innen die Möglichkeit bieten, für ihre Lernfortschritte Feedback zu erhalten.  
Das System muss den Schüler\*innen die Möglichkeit bieten, Aufgaben durch Interaktion mit dem System bearbeiten zu können.  
Der Lernfortschritt sollte immer aktuell gehalten werden.  
Schüler\*innen muss dazu in der Lage sein, bereits erfüllte Aufgaben zu wiederholen.  
Das System muss in der Lage sein, Schüler\*innen sowohl intrinsisch als auch extrinsisch zu motivieren.  
Das System sollte den Lehrer\*innen die Möglichkeit bieten, den Fortschritt ihrer Schüler\*innen betrachten zu können.

#### Qualitative Anforderungen

Das System sollte den Schüler\*innen die Möglichkeit bieten, Spaß bei der Bearbeitung der Aufgaben zu haben, damit diese nicht die Motivation verlieren.  
Das System sollte für die Schüler\*innen einfach und intuitiv zu verstehen sein.  
Das System muss fähig sein, mit aktuellen Informationen relevant zu bleiben.  
Das System muss fähig sein, alte Aufgabentypen an neue Informationen anzupassen.  
Das System sollte mit angesehenen Organisationen zusammenarbeiten, um auf dem neuesten Stand zu bleiben.  
Das System sollte mit angesehenen Organisationen zusammenarbeiten, und den Schüler\*innen sicherheitskritische Informationen zu vermitteln.  
Das System sollte es den Lehrer\*innen einfach machen, das System lauffähig zu machen, zu verstehen und den Schüler\*innen zu erklären.  
Das System sollte in der Lage sein, für so viele Schüler\*innen wie möglich zugänglich zu sein.  
Das System sollte eine Möglichkeit bieten, trotz mangelnden Ressourcen funktional zu bleiben.

#### Organisationale Anforderungen

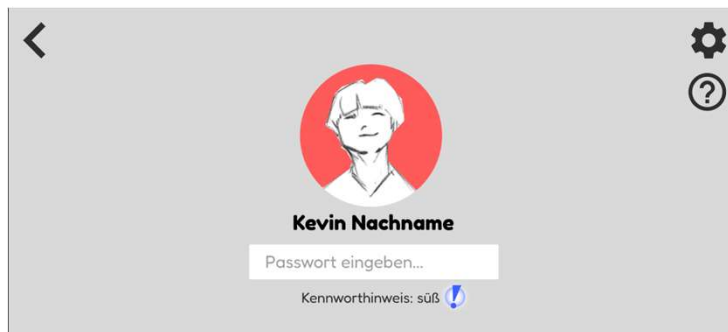
Das System sollte von dem Schulministerium gefördert werden, und von Interesse zu sein in das Curriculum integriert zu werden.  
Die Schule muss fähig sein, Schüler\*innen andere Möglichkeiten zur Verfügung stellen, das System zu benutzen.  
Das System muss in der Lage sein, den Lehrstoff und die Lernziele in der Schule zu unterstützen.

## Proof of Concepts

- 1 Aufgabentyp mit Prototyp
- 2 Levelauswahl
- 3 Erklärungen
- 4 Remote Daten
- 5 Offline Funktionalitäten

## PoC 1 : Aufgabentyp mit Prototyp

- Simple Texteingabe
- Richtig: Positives Feedback
- Falsch: Erklärung



Hier sieht man einen Teil unseres Rapid Prototypes zum Thema Aufgabentyp. Alle Wireframes kann man hier in unserem Repo finden: <LINK>

Oder auf unserem Figma Board. Dort lassen sich auch unsere nicht fertigen Designs und sonstige Komponenten finden:

<https://www.figma.com/file/xpPsJF5V2h76NvHuOkx4mv/EP-Rapid-Prototype-Screens?type=design&node-id=0%3A1&mode=design&t=UGYvEkmxQ5Z3y0q4-1>

## PoC 1 : Risiken/Fail Kriterien

- Applikation crasht während der Bearbeitung
- Nutzer verliert während der Bearbeitung die Motivation und bricht die Bearbeitung ab
  - Kein negatives Feedback sondern Erklärungen
  - Großzügige Tipps
- Es wird kein Feedback gegeben
  - Feedback wird bei jeder Eingabe gegeben
- Nutzer ändert seine Eingabe nachträglich um doch noch positives Feedback zu erhalten
  - Aufgabe ist nach Submit nicht mehr bearbeitbar

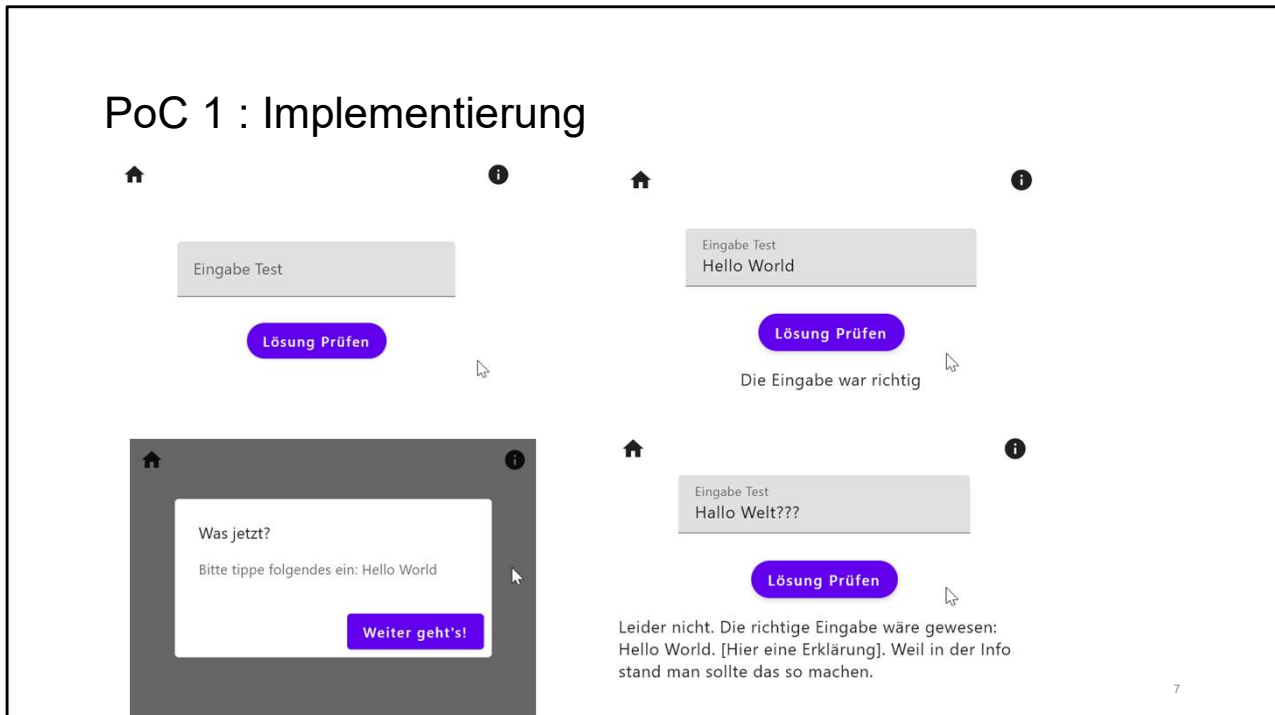
6

Hier unsere Risiken und Fail Kriterien, sowie einige passende Fallbacks, und eventuell noch wie wir das Ganze im Code umgesetzt haben. Mehr dazu unter der Seite PoCs im Wiki:

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#fail-kriterien](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#fail-kriterien)

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#fallbacks](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#fallbacks)

## PoC 1 : Implementierung



Hier einige Screenshots der bereits laufenden Implementierung des PoCs. Spezifische Code Stellen sind hierzu im Wiki verlinkt.

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code)

## PoC 2 : Levelauswahl

- Auswahl der Level
- Klicken auf Haus oder Titel mit Levelnamen
- Weiterleiten auf jeweiliges Level



8

Hier ein Teil von unseren Wireframes aus dem Rapid Prototype, mit einer kurzen Beschreibung des PoCs.

<https://www.figma.com/file/xpPsJF5V2h76NvHuOkx4mv/EP-Rapid-Prototype-Screens?type=design&node-id=0%3A1&mode=design&t=UGYvEkmxQ5Z3y0q4-1>

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#beschreibung-1](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#beschreibung-1)



## PoC 2 : Risiken/Fail Kriterien

- Nutzer klickt außerhalb der Buttons und Bilder
  - Keine Reaktion auf die Eingabe
- System reagiert nicht auf das Klicken
  - Clickable Text und Image
- Angezeigte Location passt nicht zu geklickter Location
  - Jeder Button ist direkt mit der Location verlinkt

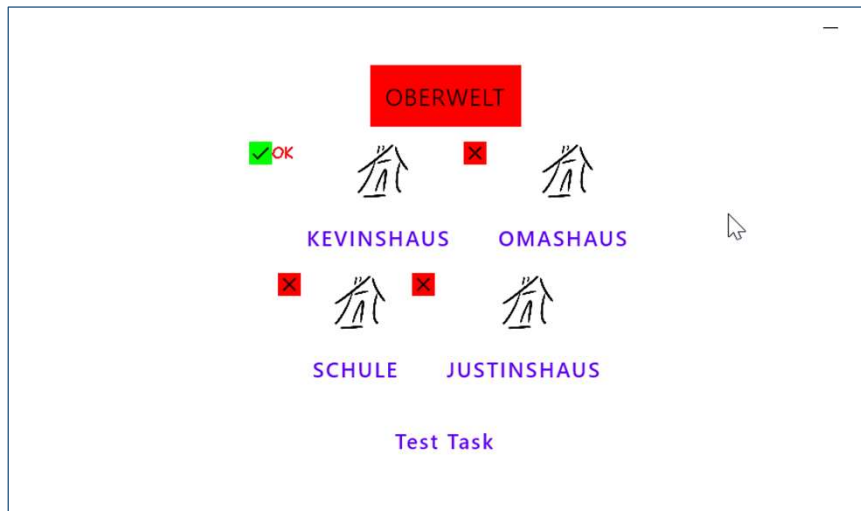
9

Hier noch einmal: Welche Fail Kriterien gibt es? Welche Fallbacks passen jeweils dazu? Wie haben wir diese im Code umgesetzt?

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#fail-kriterium](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#fail-kriterium)

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#fallbacks-1](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#fallbacks-1)

## PoC 2 : Implementierung



10

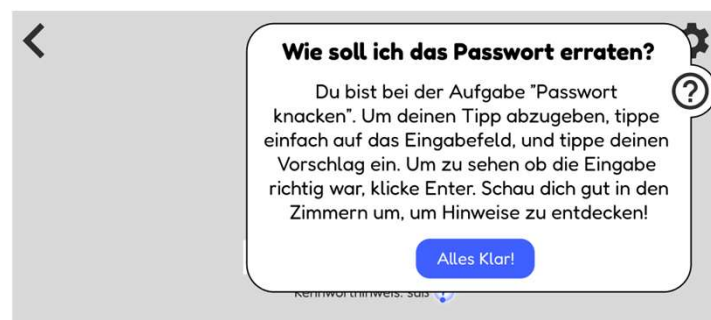
Hier ein Bild davon, wie wir die World Map bisher im Code implementiert haben.

Man kann auf das Bild, oder auf den Text klicken, um zum jeweiligem Level weitergeleitet zu werden. Im Level selber haben wir im Moment schon einige Skript Lines die angezeigt werden, mit einer entsprechenden Emotion der Charaktere

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code-1](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code-1)

## PoC 3 : Erklärungen

- Anklicken eines Fragezeichen für eine Erklärung des Aufgabentypen
- Text wird mit Level „mitgeliefert“ und ändert sich je nachdem ob man im Dialogmodus, im Untersuchungsmodus oder in einer Aufgabe ist



11

So sollten die Erklärungen aussehen. Je nach Level wird eine andere Erklärung angezeigt. Wenn man gerade mit einem anderen Charakter spricht, wird erklärt, wie der Dialogmodus funktioniert. Wenn man sich im Suchmodus befindet, und zum Beispiel durch ein Zimmer geht um etwas zu suchen, wird das dort auch erklärt. Zudem hat jede Aufgabe (Bei uns werden das am Ende wahrscheinlich 4 sein) eine eigene Erklärung.

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#beschreibung-3](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#beschreibung-3)

## PoC 3 : Risiken/Fail Kriterien

- System reagiert nicht auf Klicken
  - Fragezeichen Icon Button ist clickable
- Popup mit Erklärung ist leer/es wird keine Erklärung angezeigt
  - falls Text leer: Hinweis an Nutzer und Bitte um Neustart



Bei uns im Code momentan werden die Hilfe Buttons noch nicht generiert. Wir haben bis jetzt nur einen Platzhalter, in dem immer die selbe Hilfestellung bei allen Levels steht (Siehe Bild rechts). In der einen Test Task wird eine leicht andere Erklärung angezeigt.

Es gibt jetzt allerdings schon einen Check dafür, ob der Tipp leer ist.

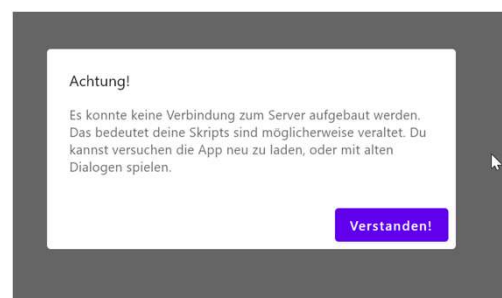
[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#fail-kriterien-1](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#fail-kriterien-1)

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#fallbacks-2](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#fallbacks-2)

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code-2](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code-2)

## PoC 4 : Remote Abfrage von Daten

- Skripte von Remote Server anfragen
- Bei Erfolg: Neue Skripte nehmen
- Sonst: Hinweis an Nutzer und Lokale Scripts verwenden die möglicherweise veraltet sind

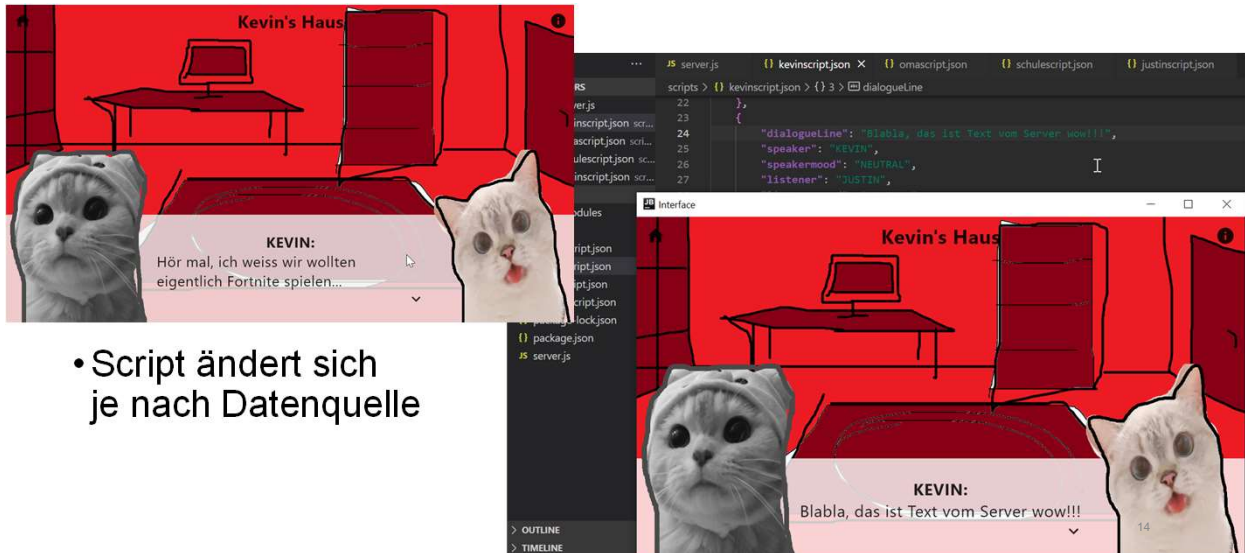


13

Die Idee ursprünglich war, eine remote Datenbank zu haben, spezifischer mit Firebase. Wir hatten dann allerdings Probleme bei der Integration von Firebase in unserem Kotlin Multiplatform Projekt, und haben uns dann entschieden als simpleren Proof of Concept zunächst einen einfachen lokalen Server mit NodeJS zu implementieren, der bei Anfrage eine JSON Datei mit den Dialoglinien zurückschickt. Zum Beginn der App wird ein kleiner Ladescreen angezeigt. Wenn die Verbindung zum Server erfolgreich ist, läuft das Spiel normal. Wenn die Verbindung fehlschlägt, werden die lokalen Scripts aus dem Android Projekt genommen, mit einem Hinweis an den Nutzer dass die Skripts möglicherweise veraltet sind. Die Funktion ist dazu da, dass man Skripts aktualisieren kann, ohne die ganze App zu updaten. Außerdem brauchen wir später einen Server, um den Fortschritt der Schüler und Lehrer zu synchronisieren.

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code-3](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#implementierung-im-code-3)

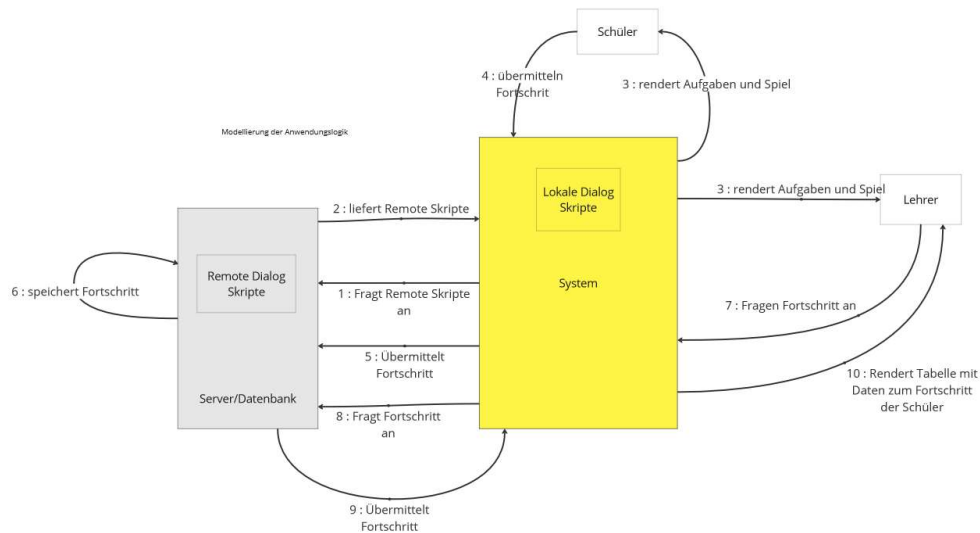
## PoC 5 : Offline Funktionalitäten



Als Fortsetzung des vierten PoCs wird hier geprüft, ob die App auch Offline laufen kann. In unserem Fall noch ja, weil hier nur die Skripts geprüft wurden. Selbst wenn die Verbindung zu einem Server mal nicht hergestellt werden kann, wird der Fortschritt offline gespeichert und die Level werden trotzdem ordnungsgemäß angezeigt.

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/wiki/07-Proof-of-Concept#poc-5--funktionalit%C3%A4ten-des-systems-bleiben-erhalten-nachdem-die-internetverbindung-abgebrochen-wurde](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/wiki/07-Proof-of-Concept#poc-5--funktionalit%C3%A4ten-des-systems-bleiben-erhalten-nachdem-die-internetverbindung-abgebrochen-wurde)

## Anwendungslogik



15

Hier sieht man eine schnelle Modellierung eines Anwendungsfalls, um nochmal als Zusammenspiel des Servers und der App zu highlighten. Lehrer und Schüler bekommen eine visuelle Darstellung der Daten, die von dem Server bereitgestellt werden (Durch die Verarbeitung einer JSON File). Der Fortschritt der Schüler wird von der App an den Server übermittelt, dort persistent gespeichert mithilfe einer Datenbank, und kann dann von den Lehrern bei Bedarf abgerufen werden.

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/blob/main/Artefakte/Anwendungsfall\\_Anwendungslogik\\_Modellierung.png](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/blob/main/Artefakte/Anwendungsfall_Anwendungslogik_Modellierung.png)

## Wireframes

- Login
  - für Lehrer\*innen
  - für Schüler\*innen
- Home
- Fortschritt
  - für Lehrer\*innen
  - für Schüler\*innen
- Dialog
- Oberwelt
- Aufgabe und Feedback

16

Figma Board für das alle Wireframes und Skizzen

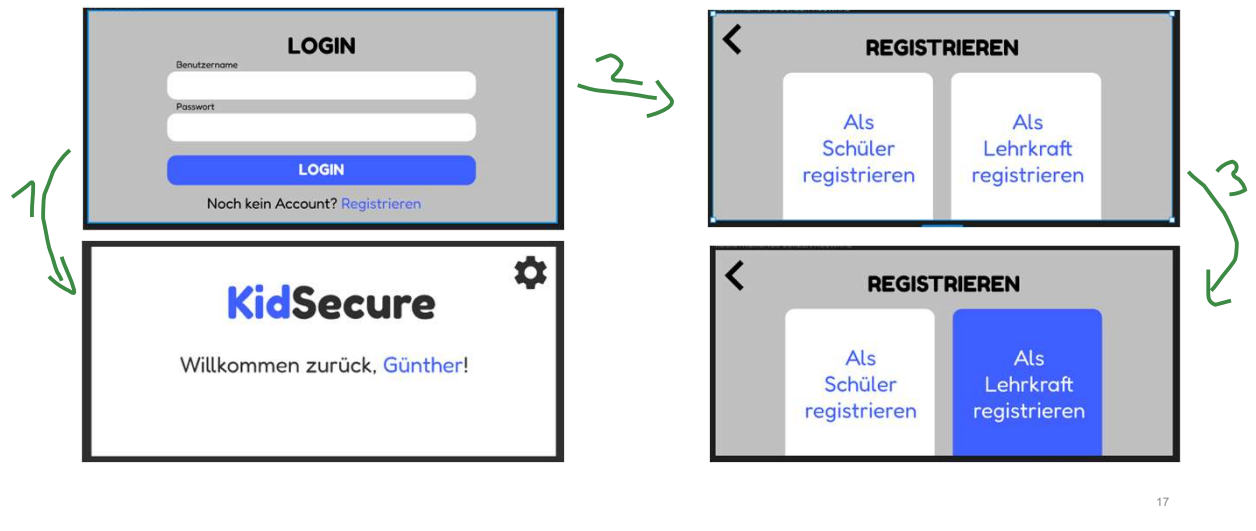
<https://www.figma.com/file/xpPsJF5V2h76NvHuOkx4mv/EP-Rapid-Prototype-Screens?type=design&node-id=0-1&mode=design&t=owOB0hkRZRiOaZGN-0>

RapidPrototype

[https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/blob/main/Artefakte/Rapid%20Prototype/Rapid%20Prototype%20Wireframes.png](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/blob/main/Artefakte/Rapid%20Prototype/Rapid%20Prototype%20Wireframes.png)



## Login und Registrierung



Die Zahlen sind eine Nummerierung der Pfeile und besagen nichts über ihre Reihenfolge

1. Bei einem erfolgreichen Login wird ein Begrüßungsfenster aufgerufen // Schüler\*innen werden mit ihrem Vornamen angesprochen, Lehrer\*innen mit der Anrede und Nachnamen, z.B. Frau Müller.
2. Drückt man auf Registrieren, wird man zu einem Registrierungsfenster weitergeleitet, in dem man auswählen kann, ob man sich als Schüler\*in oder als Lehrer\*in registrieren möchte.
3. Das drücken der Taste gibt ein visuelles Feedback.

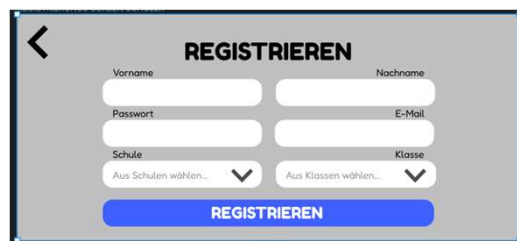
## Registrierung

- Lehrer\*innen



The registration form for teachers is titled "REGISTRIEREN" and features a back arrow in the top left corner. It contains the following fields: "Vorname" (first name), "Nachname" (last name), "Passwort" (password), "E-Mail", "Schule" (school) with a dropdown menu labeled "Aus Schulen wählen...", and "LehrerID" (teacher ID). A blue "REGISTRIEREN" button is at the bottom.

- Schüler\*innen

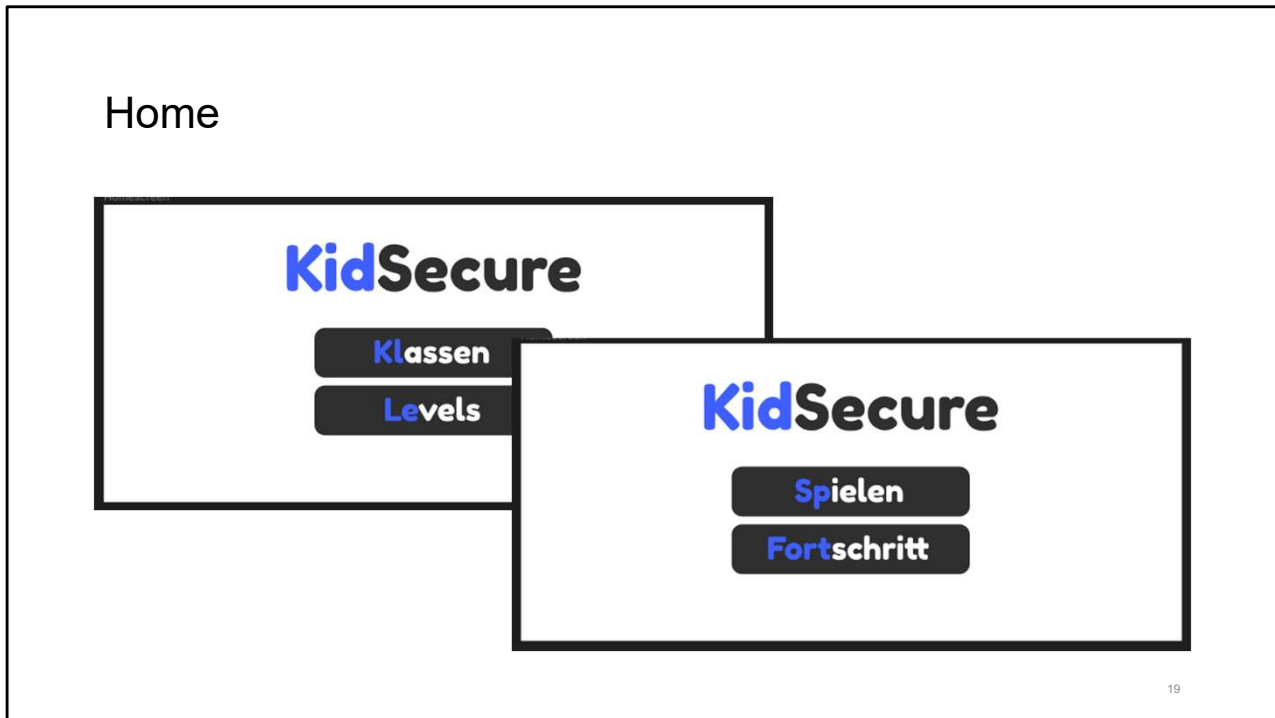


The registration form for students is titled "REGISTRIEREN" and features a back arrow in the top left corner. It contains the following fields: "Vorname" (first name), "Nachname" (last name), "Passwort" (password), "E-Mail", "Schule" (school) with a dropdown menu labeled "Aus Schulen wählen...", and "Klasse" (class) with a dropdown menu labeled "Aus Klassen wählen...". A blue "REGISTRIEREN" button is at the bottom.

18

Das Registrierungsfenster ist fast identisch, den Unterschied macht das letzte Feld. Bei den Lehrer\*innen soll eine eindeutige LehrerID eingegeben werden, damit geprüft werden kann, dass es sich um diesen Lehrer handelt

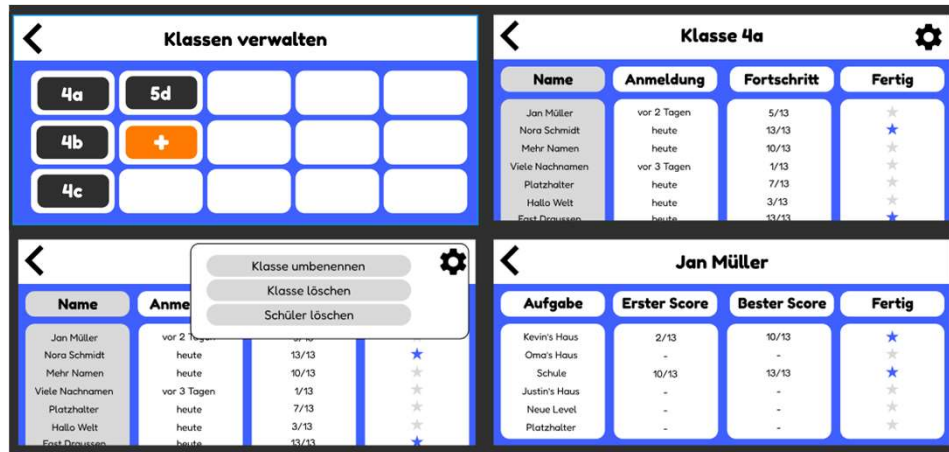
Die Schüler\*innen wählen stattdessen ihre Klasse. Da es bei Lehrer\*innen der Fall sein kann, dass sie von mehreren Klassen die Klassenlehrer\*innen sind, soll es im späteren Verlauf möglich sein, ihre Klassen zu verwalten.



Die Homescreens für Schüler\*innen und Lehrer\*innen haben verschiedene Funktionen. Die Lehrer\*innen sollen in der Lage sein, ihre Klassen zu verwalten, und sich die Levels in einem Admin-Modus anzuschauen, in dem sie Zugriff auf alle Aufgaben und Lösungen haben. Das soll ihnen ermöglichen, den Schüler\*innen Hilfestellung zu geben, wenn diese mit Fragen kommen sollten. Außerdem können sie unter Klassen auch den Fortschritt ihrer Schüler\*innen betrachten.

Die Schüler\*innen haben Zugriff auf das Lernspiel und ihren eigenen Fortschritt.

## Fortschritt Lehrer\*innen



20

Die vier Screens zeigen einmal die Klassenverwaltung, die Klassenliste einer ausgewählten Klasse, die Einstellungen, die in der Klasse vorgenommen werden können und den Fortschritt einer\*s ausgewählten Schüler\*in.

In der Klassenverwaltung können Lehrer\*innen die Klassen verwalten für die sie zuständig sind. Die Löschung einer Klasse ist in diesem Screen nicht möglich, um die versehentliche Löschung einer Klasse zu vermeiden. Es sollte nicht so oft vorkommen, dass eine vollständige Klasse gelöscht werden muss. Die Buttons führen in die Klassenliste.

Die Klassenliste zeigt einen kurzen Überblick des Fortschritts aller Schüler\*innen. Wählt man eine\*n Schüler\*in aus, so wird die detaillierte Fortschrittsanzeige der\*s Schüler\*in aufgerufen. Das Zahnrad öffnet weitere Funktionen.

Der dritte Screen zeigt die Einstellungen, die für eine Klasse vorgenommen werden können. Hier ist die Löschung der Klasse möglich. Übernehmen Lehrer\*innen ihrer Klassen im nächsten Jahr, so kann die Klasse umbenannt werden. Für den Fall, dass Schüler\*innen die Klasse verlassen, können sie

aus der Klasse gelöscht werden. Die Schüler\*innen kommen durch ihre Anmeldung in die Klassenliste.

## Fortschritt Schüler\*innen



The screenshot shows a progress overview for a student named Jan Müller. It features a table with four columns: 'Aufgabe' (Task), 'Erster Score' (First Score), 'Bester Score' (Best Score), and 'Fertig' (Finished). The tasks listed are Kevin's Haus, Oma's Haus, Schule, Justin's Haus, Neue Level, and Platzhalter. The 'Fertig' column uses blue stars to indicate completion status.

Aufgabe	Erster Score	Bester Score	Fertig
Kevin's Haus	2/13	10/13	★
Oma's Haus	-	-	★
Schule	10/13	13/13	★
Justin's Haus	-	-	★
Neue Level	-	-	★
Platzhalter	-	-	★

21

- entspricht der detaillierten Fortschrittsanzeige, die Lehrer\*innen anschauen können
- Aufgaben sollen öfter gelöst werden können
  - der Unterschied zwischen erstem und bestem Score soll die Verbesserung der Schüler\*innen zeigen

## Oberwelt



22

Dieser Screen bildet die Oberwelt ab, die die Stadt darstellt, in der der Spielcharakter lebt und von der aus er mit der Welt interagieren kann. Wir sehen 5 Levels, in die der Spielcharakter gelangt. Von dort aus muss er in die Oberwelt zurück, um ein anderes Level zu besuchen. Die Levels sind Points of Interest, zum Beispiel das Haus eines Freundes, oder aber die Schule, in denen Aufgaben gefunden und Dialoge gestartet werden können.

Die Icons signalisieren den Schüler\*innen, welche Zustände ein Level hat, damit die Navigation einfacher fällt.

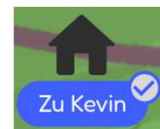
- ein Ausrufezeichen über dem Haus/Gebäudesymbol bedeutet, dass das Level noch nicht besucht worden ist.
- das Ausrufezeichen verschwindet, sobald das Level besucht worden ist
- Ein unabehakter Kreis bedeutet, dass in dem Level noch keine Aufgaben gelöst worden sind
- ein schwarzer, abgehakter Kreis bedeutet, dass mindestens eine Aufgabe erledigt worden ist
- ein blauer, abgehakter Kreis bedeutet, dass alle Aufgaben in dem Level gelöst worden sind.

- Zahnrad ermöglicht das Speichern und Beenden der App



## Oberwelt Zustände

- unentdecktes Level
- keine Aufgaben gelöst
- mindestens eine Aufgabe gelöst
- alle Aufgaben gelöst



23

Es wird noch überlegt, ob das Ausrufezeichen wieder auftaucht, wenn ein neues Ereignis innerhalb des Levels begonnen werden kann, also zum Beispiel eine Aufgabe entdeckt oder ein Dialog gestartet. Dann würde das Ausrufezeichen nicht mehr „unentdeckter Ort“ bedeuten, sondern auf ein „neues Ereignis“ hinweisen. Die Symbole sollen bei der Orientierung helfen. So müssen die Schüler\*innen bei einem blau abgehakten Level nicht mehr nach einer Aufgabe schauen, da sie alle gelöst haben. Es hat außerdem den Effekt, dass Schüler\*innen ein visuelles Feedback erhalten und womöglich dazu angestoßen werden, einen blauen Haken zu erhalten, also alle Aufgaben zu lösen. Es wird noch darüber diskutiert, wie wir den visuellen Effekt verstärken können, evtl. könnten wir statt dem blauen Haken einen Stern benutzen, der eher mit einem Erfolg in Verbindung gebracht.

## Dialog, Levelnavigation



24

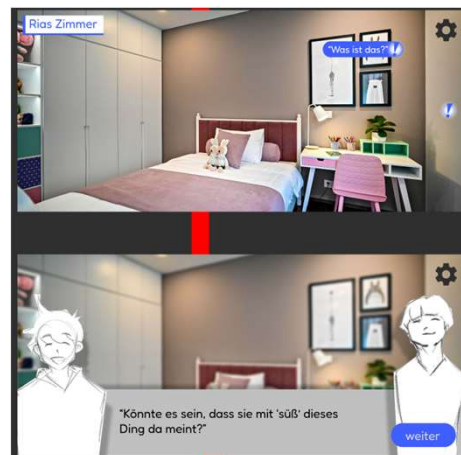
- Dialogpartner sollen vor dem Dialogfenster und dem Hintergrund abgebildet sein und mit einem Schatten oder in Compose mit einem anderen Effekt hervorgehoben werden
- es soll einen Speaker und einen Listener geben. Der Listener wird durch grayscaling in den Hintergrund gerückt —> visuelle Hervorhebung des Sprechers
- Der Hintergrund ist geblurt, wenn eine Unterhaltung stattfindet
- die Buttons zurück und weiter ermöglichen das Nachlesen, wenn man einmal etwas unaufmerksam war oder ungeduldig Dialoge skippt
- Ausrufezeichen weisen auf eine mögliche Aktion hin
- Interaktion mit dem Ausrufezeichen öffnet die Aktion, die ausgeführt werden kann. Diese muss nochmal durch Antippen bestätigt werden.
- Das Zahnrad ermöglicht das jederzeitige Speichern des Fortschritts und das Beenden einer Aufgabe, was einen zurück zur Oberwelt führt
- Der gesamte bisherige Dialog zwischen Kevin und Justin kann auf de

Figma Board nachgelesen werden

- <https://www.figma.com/file/xpPsJF5V2h76NvHuOkx4mv/EP-Rapid-Prototype-Screens?type=design&node-id=0-1&mode=design&t=vL1aUXKfsbp2CAka-0>
- [https://github.com/AnoukMartinez/EP\\_WS23\\_Ganz\\_Khamkaew\\_Martinez/blob/main/Artefakte/Rapid%20Prototype/Rapid%20Prototype%20Wireframes.png](https://github.com/AnoukMartinez/EP_WS23_Ganz_Khamkaew_Martinez/blob/main/Artefakte/Rapid%20Prototype/Rapid%20Prototype%20Wireframes.png)

## Levelinteraktion

- Levelbezeichnung links oben
- Interaktion mit einem Ausrufezeichen kann einen Dialog starten



25

Das Ausrufezeichen rechts führt uns zurück zu Kevins Zimmer, diesmal auf der rechten Seite, da natürlicher.

Die Levelbezeichnung links oben dient zur Navigation

Das Zahnrad beendet entweder den Dialog oder ermöglicht das Zurückkehren auf die Oberwelt

## Aufgabe Hinweis



26

- Interaktion mit dem Computer in Kevins Zimmer ermöglicht das Lösen der Aufgabe
- hier soll ein Passwort eingegeben werden, welches seine Schwester beim letzten Login geändert hatte
- Das Fragezeichen öffnet ein Fenster, das eine Erklärung oder Hinweise enthält

## Negatives Feedback

- Wird bei einer nicht gelösten Aufgabe ausgelöst

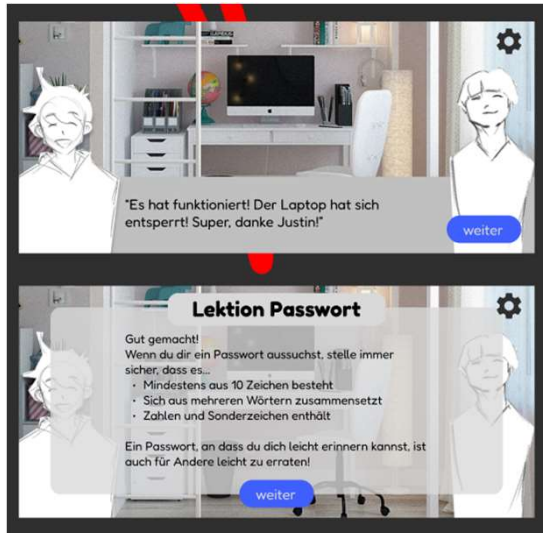


27

- Schüler\*innen haben die Möglichkeit, direkt nochmal die Aufgabe zu machen, sollten sie wissen, wo ihr Fehler lag, oder mit weiter ins Zimmer zurückkehren und weiter Hinweise sammeln, mit Personen reden, die Tipps geben, etc.
- Das Feedback ist positiv formuliert, gibt Anstöße und Hinweise, wie weiterrvorgegangen werden kann
- Soll nicht durch negatives Feedback demotivieren, ermutigt zum Ausprobieren, Erkunden und Weitermachen

## Positives Feedback

- Wird bei einer erfolgreich gelösten Aufgabe ausgelöst



28

Das korrekte Lösen einer Aufgabe löst eine Lektion aus, die mit der beendeten Aufgabe zusammenhängt. Hiermit soll nochmal Stoff vertieft werden.

- vermittelt Wissen durch Wiederholung, vertieft Stoff, erleichtert Lösen ähnlicher Aufgaben

## Artefakte für das 4. Audit

Aufgabe	Deadline	geplanter Start	fertiggestellt am	geplanter Zeitaufwand	tatsächlicher Zeitaufwand	
Exposé	29.10.2023	09.10.2023	20.10.2023	10h		
Working-Title vergeben	29.10.2023	09.10.2023	20.10.2023	1h	0,5h	
Projektplan	29.10.2023	23.10.2023		3h	3h	
Exposé-Abgabe	30.10.2023	29.10.2023				
Domänenanalyse	29.10.2023	09.10.2023	21.10.2023	4h	3h	
Stakeholderanalyse	09.11.2023	09.11.2023	09.11.2023	3h	4h	
Personas	09.11.2023	09.11.2023	09.11.2023	2h	2h	
Anforderungsermittlung	08.12.2023	13.11.2023	23.11.2023	2h	2h	
funktionale Use Cases	08.12.2023	13.11.2023	23.11.2023	1h	1h30	
Interview	<del>23.02.2024</del>					
Projektrisiken	08.12.2023	13.11.2023	30.11.2023	3h	3h	
Proof of Concept Konzept	08.12.2023	13.11.2023	30.11.2023	6h	6h	
Proof of Concept Implementierung	18.01.2024	13.11.2023	19.01.2024	10h	8h	
Überarbeitung nach Audit 1	08.12.2023	13.11.2023	07.12.2023	2h	1h	
Überarbeitung nach Audit 2 (Anforderu	18.01.2024	14.12.2023	19.01.2024	0,5h	0,5h	
Überarbeitung nach Audit 2 (PoC)	18.01.2024	14.12.2023	19.01.2024	4h	4h	
Überarbeitung nach Audit 2 (Risiko)	18.01.2024	14.12.2023	14.12.2023	1h	1h	
Rapid Prototype Wireframes	20.01.2024	01.01.2024	19.01.2024	5h	5h	
Anwendungslogik Skizze	20.01.2024	19.01.2024	19.01.2024	0,1h	0,1h	
Medienkompetenzrahmen Recherche	<del>23.02.2024</del>					
Prototyp	23.02.2024	22.01.2024		50h		
1. Audit	13.11.2023					
2. Audit	11.12.2023					
3. Audit	22.01.2024					
4. Audit	26.02.2024					
Poster	26.02.2024	22.01.2024				
Poster-Abgabe	26.02.2024					