

# KidSecure Projektbericht

## 1 Einleitung

### 1.1 Projektidee

Im Rahmen des Projektes war eine Lernapplikation geplant, mit der Schüler\*innen auf die Gefahren und Strukturen des Phishing aufmerksam gemacht werden.

Hierzu soll ein Spiel entwickelt werden, in dem ein\*e Protagonist\*in verschiedene Situationen in unterschiedlichen Welten begegnet, und dann Aufgaben zu der Thematik lösen muss.

In späteren Iterationen, insbesondere nach der Risikoanalyse wurden noch weitere Features der App ausgearbeitet. Zu diesen zählen beispielsweise, dass Lehrkräfte den Fortschritt von Schüler\*innen verfolgen können, um diese bei dem Bearbeiten der Aufgaben zu unterstützen und ihren Unterricht dementsprechend zu planen. Dabei sollen Lehrkräfte nicht nur den Fortschritt von jedem\*r Schüler\*in einsehen, sondern auch eine Liste von Klassen erstellen und verwalten können.

Schüler\*innen sollen ebenfalls ihren eigenen Fortschritt einsehen können, und mit einem simplem Punktsystem motiviert werden, sich gegebenenfalls mehrmals an den einzelnen Levels zu versuchen.

### 1.2 Artefakte

#### 1.2.1 Personas/Stakeholderanalyse, Domänenmodell und Use Cases

Im Rahmen der Initialen Überlegung haben wir Personas unser möglichen Stakeholder angefertigt.

Die Überlegung hierbei war, dass das System nicht nur Kindern direkt helfen könnte, sondern auch den Lehrkräften, die möglicherweise nicht die Zeit und Ressourcen haben, um sich selber mit dem Thema zu beschäftigen. Eltern und Großeltern können ebenfalls von dem Wissen der Kinder profitieren, da diese möglicherweise gemeinsam Phishing Versuche verhindern können.

Dieser Schritt half uns die Interessen der einzelnen Personengruppen genau zu verstehen, und Features wie zum Beispiel den Klassenfortschritt für die Lehrer\*innen zu erarbeiten.

#### 1.2.2 Risikoanalyse und Anforderungsermittlung

Als Planung für die Implementierung haben wir uns intensiv mit den Anforderungen des System beschäftigt. Einer der wichtigsten Aspekte der Planung generell war hierbei die Risikoanalyse. Durch gemeinsames Brainstorming haben wir in mehreren Iterationen die wichtigsten Risiken der Projektes erarbeitet, und diese kategorisiert und priorisiert.

Durch die Analyse haben sich mehrere Features der App ergeben. Zusätzlich zu den Aufgabenstellungen in jedem Schritt der App ebenfalls eine Hilfestellung gegeben. Lehrkräfte können ebenfalls den Fortschritt von Schüler\*innen überwachen.

Ebenfalls wichtig war für uns, dass Schüler\*innen nicht von einer schlechten Internetverbindung beeinflusst werden, und Aufgaben trotzdem bearbeiten können, selbst wenn keine stabile Verbindung zum Server besteht. Hierbei wurden lokale Skripte in Betracht gezogen, die möglicherweise leicht veraltet sein können, es Schüler\*innen allerdings trotzdem erlaubt, Aufgaben zu bearbeiten. Das Feature wurde in den Proof of Concepts umgesetzt.

Vorher hatten wir recht wenig die technischen Limitierungen in Schulen gedacht. So ist auch die Idee entstanden, die App Multiplattform zu machen, damit möglichst viele Schüler\*innen das System nutzen können, und Kinder ohne Smartphone nicht diskriminiert werden.

### **1.2.3 Rapid Prototype Wireframes**

Ein ebenfalls sehr wichtiger Aspekt der Planung waren die Wireframes, beziehungsweise der Rapid Prototype. Dieser Schritt half uns die Navigation zwischen den einzelnen Screens richtig zu visualisieren, und ein konkretes Bild davon zu haben, wie genau der Flow der App aussehen soll.

## 2 Umsetzung und Implementierung

### 2.1 Planung und Multiplatform Tests

Bevor die eigentliche Bearbeitung am Projekt startete, haben wir zunächst einige Prototypen mit Compose Multiplatform erstellt, um uns in das Thema einzuarbeiten. Hierbei wurde in einem Test Repository gearbeitet, welches später mit dem Haupt Repository „gemerged“ wurde.

Zunächst wurde eine relativ simple Worldmap erstellt, auf der man mit Hilfe von vier Buttons auf ein Bild, passend zu der jeweiligen Location weitergeleitet wird. Hierbei gab es zunächst einige Probleme mit Multiplatform, da das Verarbeiten von Bildressourcen auf Mobile und Desktop unterschiedlich gehandelt wird.

In unseren nächsten Schritten wurde ein simples Dialogsystem entwickelt, welches ein Skript mit DialogueLine Objekten verarbeitet, und durch Klicken der Textbox weiter im Text fortschreitet. Man kann außerdem eine kleine Hilfestellung bekommen, oder zurück auf die Levelübersicht, also die Worldmap kommen.

An diesem Punkt haben wir die Arbeit an dem eigentlichen Prototypen begonnen. Hierzu haben wir den bisherigen Testcode mit dem Haupt Repository vereint, um auf dessen Basis weiter aufzubauen. Um uns ein besseres Bild von den benötigten Tools zu machen, haben wir ein simples Architekturdiagramm erstellt (Im Repository als Anwendungsfall\_Anwendungslogik\_Modellierung).

### 2.2 Proof of Concepts

In den PoCs haben wir bereits fast alle unsere Fallbacks für mögliche Risiken implementiert. Hierzu zählen die Hilfestellungen für Schüler\*innen, die Skripte die zunächst vom Server geladen, und dann falls nötig lokal geladen werden, und eine Test Aufgabe, die je nach richtiger oder falscher Beantwortung ein Feedback gibt. Alle PoC Codes wurden im finalen Prototypen weiter verwendet, und nur noch ergänzt oder minimal abgeändert.

### 2.3 Tools und Bibliotheken

#### 2.3.1 JSON Prozessor

Aufbauend auf dem Test Repository haben wir uns überlegt, dass eine generelle Verarbeitung der Skripte im JSON Format über einen längeren Zeitraum mehr Sinn machen könnte. Hierzu implementierten wir einen simplen JSON Parser mit Hilfe der Jackson Bibliothek.

#### 2.3.2 Navigation

Außerdem wollten wir einen NavController implementieren, um den Code übersichtlicher und leichter maintainable zu machen. Hierbei stießen wir auf das Problem, das der NavController als solches, wie wir ihn in herkömmlichen Android Projekten verwendet haben, noch nicht in Compose Multiplatform verfügbar ist.

Wir haben uns demnach auf eine Bibliothek einer dritten Partei verlassen, die in der Android Dokumentation verlinkt war. Die Implementierung dieser war recht simpel und hat nur ein Problem mit dem finalen Build verursacht, welches hier nicht weiter relevant ist und gegebenenfalls in unserem Coding Diary noch einmal weiter erläutert ist.

#### 2.3.3 Datenbank und HTTP Requests

Eines der Risiken die wir im Verlaufe der Vorbereitung ausgearbeitet haben ist, dass die Skripte möglicherweise nicht aktuell bleiben, bei einem sich ständig weiterentwickelndem Thema wie Phishing. Wir wollten demnach im besten Fall eine Datenbank implementieren, die nicht nur Skripte auf dem neuesten Stand hält, sondern auch den Fortschritt von Schüler\*innen bei

Internetverlust so lange speichert, bis wieder eine Verbindung hergestellt werden kann. Hierzu viel uns Firebase auf, welches genau diese Funktionalitäten bereitstellt.

Allerdings war die Integrierung von Firebase in Kotlin Multiplatform sehr problematisch, und wir haben letztendlich auf eine einfachere Version einer Remote Datenbank im Prototypen zurückgegriffen.

Hierbei wurde einfach ein simpler Server mit NodeJS, Express und Javascript geschrieben, der über den Localhost läuft. Das Kotlin Backend kommuniziert über simple HTTP Requests mit dem Server, erhält Skripte und Profildaten im JSON Format zurück, und verarbeitet diese lokal mit dem bereits genannten JSON Prozessor.

Zum Verschicken der HTTP Requests haben wir die Ktor Bibliothek verwendet.

Ein wichtiger Aspekt der genannt werden sollte, ist dass die Profilabfrage und die Skripte Abfrage an den Server separat passieren, allerdings beide zu Beginn bevor der Login Screen geladen wird.

Die Profildaten werden benötigt, um zu checken ob der Login valide ist, und muss daher noch vor dem Login Screen abgefragt werden. Was die Skripte angeht, ursprünglich wurden diese nur abgefragt, wenn ein\*e Schüler\*in auf den Play Button im Hauptmenü klickt. Allerdings hat die Navigation nach der Implementierung des Login Screens nicht mehr funktioniert, und hat einfach unendlich lange geladen statt danach auf die Worldmap weiterzuleiten. Ich habe leider nicht mehr herausgefunden woran dieser Fehler lag, aber ich vermute dass die NavController Bibliothek die wir benutzt haben möglicherweise nur eine limitierte Anzahl von Routen definieren kann. Ich habe deswegen in dem finalen Prototypen beide Loading Screens in einer Route zusammengefasst und lade beide Datensätze gleichzeitig.

## **2.4 Overhaul des Level Designs**

Nachdem alle wichtigen Prozesse mithilfe der Tools implementiert waren, haben wir uns Gedanken über unsere Architektur gemacht. Hierbei haben wir eine sehr lange Brainstorming Session gehabt, in der wir genau überlegt haben, welche Objekte wir für den Level Screen brauchen, und wie diese miteinander interagieren sollen. Zum Teil haben wir unsere Gedanken mit Pseudocode dokumentiert.

Genauere Details, sowie die Grafik selbst hierzu lassen sich in unserem Coding Diary finden, welches in unserem Github Wiki ist.

Wir haben hierbei unsere bisherige Architektur komplett neu aufgesetzt, und bisherige Aspekte wie zum Beispiel den Level Screen in viele kleinere Objekte aufgeteilt. Unter anderem wurde dies gemacht, um einen Investigation Mode zu implementieren, in dem sich der\*die Nutzer\*in frei zwischen verschiedenen Räumen hin und herbewegen, und mit Objekten interagieren kann.

## **2.5 Berücksichtigung der Artefakte**

In diesem Abschnitt möchte ich noch einmal auf das Zusammenspiel zwischen den Artefakten, die wir zu Beginn der Projektes erstellt haben, und der eigentlichen Implementierung eingehen. In vielen Fällen werden Artefakte und ursprüngliche Planungen außen vor gelassen, aber wir haben uns Mühe gegeben, damit unsere gesamte Arbeit im finalen Prototypen zum Vorschein kommen kann. Ich würde demnach gerne noch einmal unseren Gedankenprozess bei der Implementierung transparent machen.

### **2.5.1 Stakeholderanalyse**

Besonders wichtig hat sich die Stakeholderanalyse sowie das Domänenmodell bei dem Entwickeln der Features der App gemacht. Zu Beginn des Projektes war nur eine App geplant, mit der\*die Schüler\*in Aufgaben in einer Art Quiz bearbeiten sollen. Durch die Arbeit an unseren Artefakten wurde uns klar, dass dies im Kontext der Unterrichts wahrscheinlich weniger sinnvoll ist. Beispielsweise ist es nicht möglich, für Lehrkräfte nachzuvollziehen, wie gut

Schüler\*innen die Aufgaben absolvieren, und ob sie diese überhaupt machen. Die Lehrer\*in Funktionen zur Übersicht der Schüler\*innen Fortschritte wurden demnach ebenfalls nach dieser Analyse erarbeitet.

Außerdem waren wir der Meinung, dass das Format besonders für jüngere Schüler\*innen nicht motivieren genug ist, und möglicherweise zu „trocken“ ist um ein wahres Interesse zu wecken. Wir haben uns nach reichlicher Überlegung dafür dann dazu entschieden, eher in die Richtung Lernspiel statt Lernquiz zu gehen, mit einer simplen Story, lustigen Charakteren und Dialogen, damit sich die Schüler\*innen involvierter fühlen können.

### **2.5.2 Risikoanalyse und Anforderungsermittlung**

Wie bereits erwähnt war eine konstante Sorge von uns, dass Schüler\*innen kein Interesse daran haben, das System zu benutzen, oder frühzeitig das Bearbeiten abbrechen. Wir haben uns demnach reichlich Überlegungen dazu gemacht, welche Gründe ein frühzeitiges Abbrechen haben könnten, und wie wir diesen entgegenwirken können.

Im finalen Prototypen zeigt sich dies unter Anderem darin, dass bei jedem Schritt eine Hilfestellung verfügbar ist. Hinweise der Charaktere auf die Lösungen für bestimmte Aufgaben sind sehr großzügig. Außerdem gibt es ein Punktesystem, je nachdem wie viele Versuche ein\*e Schüler\*in braucht, um eine Aufgabe richtig zu beantworten. Bei einer „perfekten“ Aufgabe wird das Level auf der Worldmap mit einem goldenem Stern versehen.

### **2.5.3 Prototypes und Wireframes**

Der Overhaul des Leveldesigns orientierte sich stark an den Wireframes, die wir für den dritten Audit angefertigt hatten. Zum Beispiel hatten wir jetzt ein klares Bild davon, wie der Investigations Modus aussehen soll. Hierbei haben wir die sogenannten „LookHere“s entwickelt, kleine Ausrufezeichen, mit denen man interagieren kann um Dialogsequenzen auszulösen.

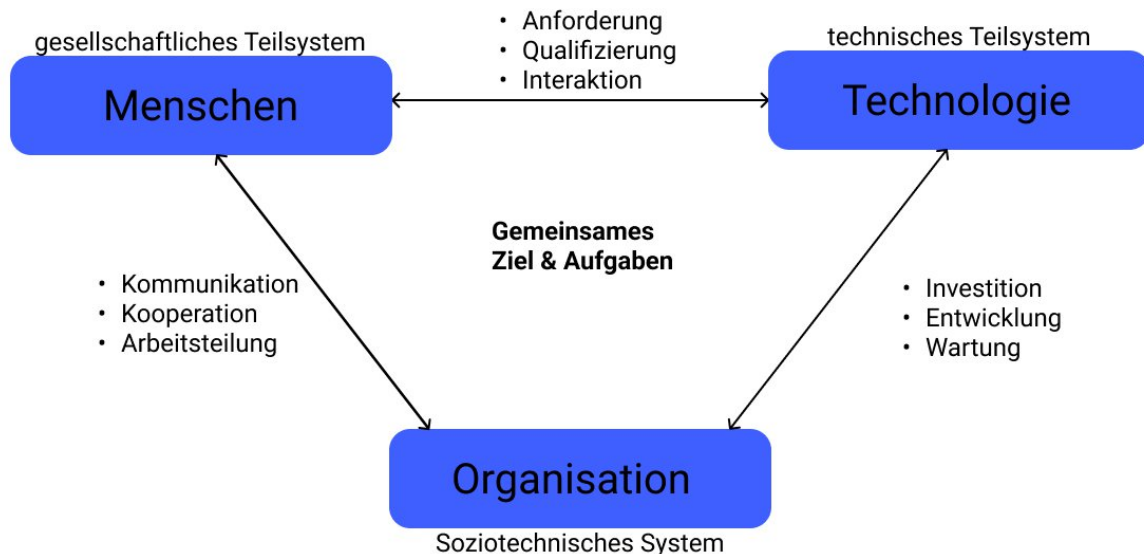
Der Login und Registrierungsscreen wurden ebenfalls anhand der Wireframes modelliert, besonders der Screen auf dem man sich zwischen einer Schüler- vs Lehrerregistrierung entscheiden kann.

### 3. Soziotechnische Perspektive

#### 3.1. Warum Social Computing als Schwerpunkt?

Unter Social Computing fasst die TH Köln Campus Gummersbach u.a. die Module der soziotechnischen Systeme und Gamification zusammen.

Es wird von einem soziotechnischen System gesprochen, wenn das vorliegende System die Beziehungen und Wechselwirkungen zwischen Technik und Gesellschaft abbildet. Die unten stehende Grafik soll dies verdeutlichen.



Bei dem vorliegenden Projekt „KidSecure“ bedeutet dies ausformuliert, dass mittels technologischer Hilfsmittel das Bewusstsein geschaffen und Kompetenzen im Bereich der IT-Sicherheit vermittelt werden. In einem Austausch privater Erfahrungen wurde besonders in der Zeit der Corona-Lockdowns sichtbar, dass weder alle Schüler\*innen noch Lehrkräfte über ein fundiertes Wissen in diesem Bereich verfügen und so schadhafte Software, die für den Unterricht benötigt wurde, eine starke Verbreitung erfuhren. Hierzu wäre eine empirische Datenerhebung sicherlich interessant gewesen, um zu ermitteln, in welchem Bereich welcher Personengruppe Defizite aufweisen, was aus zeitlichen Gründen leider nicht möglich war.

Aufgrund der sich nahezu täglichen Neuentwicklungen der Gesellschaft und Technik, ist es unerlässlich sich mit Themen wie Ethik, Recht und Psychologie ebenfalls auseinander zusetzen und dies mit bei der Planung und Gestaltung miteinzubeziehen. Eine gute Richtlinie können hierfür die „Ziele für nachhaltige Entwicklung“ sein. Die nachfolgende Grafik zeigt die direkten und indirekten Ziele auf, die das Projekt KidSecure aufgreift und berücksichtigt. Aus ethischer Sicht, darf ein System keine Menschen ausgrenzen. Sei es aufgrund von Hautfarbe, Religion oder, so banal das auch klingen mag, aufgrund des finanziellen Backgrounds. Ganz besonders dann, wenn sich das System, so wie KidSecure, sich im Bereich der Bildung sieht. Dies kann z.B. Erreicht werden, in dem das System als OpenSource Software angeboten wird, oder Schulen, die ohnehin nur wenige Gelder zur Verfügung haben, einen geringen, dennoch fairen Betrag für entsprechend viele Lizenzen bezahlen. In beiden Fällen fällt den Schüler\*innen keine Kosten an, so dass diese aus finanziellen Gründen nicht ausgeschlossen werden können. Bildung sollte immer für alle frei zugänglich sein

#### indirekte Ziele



Wissen ist frei und für jede Person zugänglich



Das System ist unabhängig vom Geschlecht nutzbar



Schüler\*innen sind über den richtigen Gebrauch von Medien informiert und benutzen diese nachhaltiger

#### direkte Ziele



Phishin, Identitätsklau etc. können hohen finanziellen Schaden anrichten, Gerichtskosten etc.



Opfer von CyberCrime zu kann psychischer Belastung führen, was weitreichende Folgen haben kann



Das System ist in DE entwickelt worden und hält sich an die vorgeschriebenen Gesetze (Mindestlohn, Urlaub etc), gute Arbeitsbedingungen

Bei der rechtlichen Betrachtung der vorliegenden Systems, muss sich z.B. Die Frage bzgl. Datenschutz gestellt werden. Werden Daten gesammelt und wenn ja, welche und wie?

### 3.2. Gamification

Gamification stellt die psychologische und spielerische Umsetzung eines oben erklärten Systems dar. Genauer formuliert bedeutet Gamification, dass alltägliche Handlungen, die keinen spielerischen Kontext vorweisen, mit Spielelementen verbunden werden.

Aus psychologischer Sicht hat jeder Mensch Bedürfnisse, die, für ein mental gesundes Leben, befriedigt werden müssen. Einige diese Bedürfnisse können im Rahmen der Gamification aufgegriffen und befriedigt werden. Dabei können Herausforderungen, Ziele und Motivation hilfreich sein. Letzteres wird noch einmal in intrinsisch und extrinsisch untergliedert, wozu auch Belohnungen und Bestrafungen gezählt werden können.

#### 3.2.1. Motivation

Wie bereits geschrieben, unterscheidet man zwischen zwei Arten der Motivation: intrinsisch und extrinsisch. Intrinsisch bedeutet, dass die Motivation von einem selbst kommt und der/die Anwenderin aus eigenem Antrieb KidSecure z.B. anwendet. Dies wirkt sich wiederum positiv unter anderem auf das Bedürfnis „Selbstverwirklichung“ aus und mit Erreichen eines Zieles, meistern aller Aufgaben im vorliegenden Projekt, werden Probleme und Herausforderungen positiv wahrgenommen und angegangen. Der/ die Anwender\*in befindet sich im Flow und erlebt Niederlagen bzw. Herausforderungen nicht als etwas Negatives, sondern befriedigt das Bedürfnis der Lösung des vorliegenden Problems. Dadurch, dass sich der/ die Anwender\*in

intensiver mit einem Thema befasst, werden z.T. auch andere Quellen herangezogen und das Wissen ohne große Mühe erweitert.

Bei der extrinsischen Motivation ist der/ die Nutzer\*in nur solange an dem vorliegenden Spiel interessiert, wie sie für die Benutzung dessen von Außen belohnt wird. Zum Beispiel durch Lob und / oder gute Noten der Lehrkraft. Der Fokus hierbei liegt dann nicht auf das Lösen der Probleme / Aufgaben, sondern im Erhalten der Aufmerksamkeit. Sobald diese wegfällt, verliert sich das Interesse am Spiel und somit schwindet auch die Motivation.

### **3.2.2. Spielelemente**

Bei der Gestaltung von System, die Anwender\*innen intrinsisch motivieren, empfiehlt es sich auf bestimmte Spielelemente zurückzugreifen. Können sich Benutzer\*innen einen individuellen Charakter erstellen, können sich diese eher mit dem Spiel und der Situation näher identifizieren und bauen im Idealfall eine Beziehung zu dem Spiel auf. Dadurch wird der oben genannte, innere Wunsch der Problemlösung angestachelt. KidSecure verfügt, im Rahmen von Fortschrittsbalken, eine minimale extrinsische Belohnung, da die Anwender\*innen so einen visuellen Überblick und somit Reiz von außen haben, selbst ihre Stärken und Schwächen zu erkennen. Mittels Storytelling ist es möglich eine Welt, Szenarien etc. zu generieren, die ebenfalls das Interesse der Nutzer\*innen wecken können bzw. die ihnen selbst vertraut sind, wodurch eine weitere Identifikation mit dem System hergestellt werden kann.

Der Oberbegriff des Spielelements um die intrinsische Motivation zu fördern, nennt sich **RAMP** und ist ein Synonym für **Relatedness** (die oben beschriebene Verbundenheit mit dem System), **Autonomy** (Freiheit – in KidSecure können die Anwender\*innen frei entscheiden, welches Level zu spielen möchten und ob sie mit dem Ergebnis zufrieden sind, oder dieses wiederholen möchten), **Mastery** (Fähigkeiten erlernen bzw. ausbauen im Bereich der IT-Sicherheit ist das Konzept des Systems, dies erfüllt den Baustein demnach von selbst) und **Purpose** (Bedeutung – die Nutzer\*innen sammeln Fähigkeiten, die sie in ihrem jetzigen Privatleben anwenden können und später in ihrem Berufsalltag nicht an Bedeutung oder gar Wichtigkeit verlieren).



## 4 Fazit und Retrospektive

### 4.1 Ausgelassene oder anders implementierte Features

Wie bereits in abschnitt 2.2.3 geschildert, war ursprünglich eine volle Remote Datenbank mit persistenten Daten geplant. Im finalen Prototypen allerdings, ist nur ein Server implementiert, der JSON an den Client schickt, allerdings nicht permanent die Fortschritte der Schüler\*innen aktualisiert.

Nicht implementiert sind mindestens zwei der Aufgabentypen. Die Aufgabe in der Location Oma's Haus ist nur sehr grob implementiert mit einer Menge Platzhaltern. Allerdings war dieses Vorgehen im Rahmen des Prototypen auch angemessen.

### 4.2 Mögliche zukünftige Features

Hier eine theoretische Überlegung, welche Features im Prototypen noch ergänzt oder erweitert werden könnten.

Zum einem war der Rahmen des Story Modus ursprünglich viel größer als im Prototypen. In zukünftigen Versionen könnten weitere Welten ergänzt werden, und eine größere Storyline die alle Level verbindet.

Auf einem technischen Level könnte der Datenbank Aspekt definitiv verbessert werden, sodass keine Datendopplungen mehr nötig sind. Momentan werden außerdem alle Datensätze zu Beginn der App einmalig geladen, aus Gründen die ich in 2.2.3 weiter erläutert habe. In zukünftigen Versionen wäre es sinnvoller, beispielsweise Skripte nur dann zu laden, wenn ein\*e Schüler\*in auf die Level zugreifen möchte.

Außerdem sollten Schülerprofile jedes Mal aktualisiert werden, wenn ein Lehrer auf den Klassen Fortschritt zugreift. Im besten Fall geschieht dies sogar Live.

Momentan werden im Rahmen der App keine Daten persistent gespeichert. In Zukunft soll das natürlich der Fall sein, mit allen ethischen Überlegungen zu der Datenspeicherung. Im Rahmen des Systems soll dabei ausschließlich der Name, die Mail und die Schulen der Schüler\*innen gespeichert werden, um diese in der Datenbank zu unterscheiden und zuzuordnen. Sonstige Daten wären dann ausschließlich der jeweilige Fortschritt, der nur im Rahmen des Systems überhaupt relevant ist, und so nicht für andere Zwecke missbraucht werden kann. Es sollte zudem eine Option für Lehrer\*innen geben, alle Klassendaten am Ende des Schuljahres wieder zu löschen. So wird zum einem sichergestellt, dass die Speicherung der Daten nur eine begrenzte „Lebenszeit“ haben, da diese nach der Verwendung des Systems im Schuljahr sowieso irrelevant werden, und zum Anderem stellt es sicher dass Lehrer\*innen wirklich nur die Klassen gezeigt bekommen, die sie im Moment auch unterrichten. Beispielsweise macht es Sinn, eine Klasse 5c am Ende eines Schuljahres zu löschen, wenn der\*die Lehrer\*in dann im neuen Jahr auch eine neue Klasse 5c bekommt.

Im Prototypen zudem nicht enthalten ist die Auswahl von Schüler\*innen bei der Registrierung aus vorhandenen Schulen und Klassen, sowie Verifizierung der Lehrer durch etwas einen einmaligen Token.

### 4.3 Individuelles Fazit (Anouk Martinez)

Ich bin persönlich zufrieden mit dem Stand des Prototypen, und bin der Meinung, dass wir den Rahmen dessen was geplant war sehr gut umgesetzt haben.

Gut gelungen finde ich wie gut die ursprüngliche Planung in die Entwicklung der Features eingeflossen ist. Alles hat sich wirklich kohärent angefühlt, und nicht wie zwei separate Teile des Projektes. Besonders mit Rückblick auf das, was zu Beginn geplant war, haben wir nicht

nur die Wireframes recht gut umgesetzt, sondern sogar zum Teil mehr Features implementiert, zum Beispiel den Investigation Modus.

Zwar könnten einige Features noch verbessert oder erweitert werden, besonders in Hinsicht auf den Social Computing Aspekt (Hier als gutes Beispiel die Wahl eines\*einer Protagonisten\*in, oder einen geschlechtsneutralen Hauptcharakters). Allerdings müssten diese dann außerhalb des Projektes weiterentwickelt werden.

#### **4.4. Individuelles Fazit (Jacqueline Ganz)**

Mein Eindruck von diesem Projekt ist sehr positiv.