
# #VersusVirus Hackathon

# Solution Proposal to Challenge #80:

# One platform for communication for teachers

# Table of Contents

# Overview

**Why:** With schools closed due to the spread of the COVID-19 virus, teaching becomes challenging. Teachers and educators are using video conferencing and e-learning tools to reach out to the students, but many of these platforms are not optimal to fulfill all the functions needed or are over-complicated. Functions are not intuitive, therefore there is the need for a simple single platform for students and teachers to communicate and interact in a practical and inclusive manner.

**What:** Creating a simple, intuitive app-based platform where the interactions between students, teachers, and parents can be performed at the same place for the facilitation of education. Video classes with a chat feature as well as homework are provided on the platform.

Data safety and privacy are of high priority for the platform. In this sense, the vision is that the data produced by services such as the video class or file-sharing will not be owned or used by any third party outside of Switzerland or a big corporation.

**Who:** The end-users of this platform will be teachers, students, and parents, especially for the younger students. With a good educational platform the community at large benefits from it as it promotes information flow and provides better teaching. The key drivers needed to carry the project through are teachers for valuable inputs of the problems, needs and usability, students as end-user input, software developers (back-end and front-end), graphic designers for user interfaces, cybersecurity experts and lawyers, marketing and business experts. Key stakeholders for this project would be the government and most of all cantons. Cantons in Switzerland benefit from the

freedom to choose how best to organize their education system. With the financial support of cantons the platform would be proposed to school institutions where teachers and students could interact in an innovative way.

**How:** With first-hand input from end-users a prototype/wireframe of how the platform would look like and the functions it will have can be made, which then can be demonstrated to end-users for reiteration and optimization. Synchronously, software developers and legal/safety experts need to be consulted on what is possible and reasonable to do. With joint efforts the educational platform can come into realization. For funding of the project it will rely on governmental/canton and educational system support, with possible private support from either individuals or funds interested in supporting a project beneficial for the community. For future funding, the platform could also be sold or introduced outside Switzerland, and implement the system of yearly subscriptions to existing users (first year free). The platform will need to be maintained for the best user experience by a technical team.

# Team

We are a team of 9 highly motivated people, coming from different backgrounds, hoping our skills and ideas may bring solutions to some of the arising challenges that are occurring due to the current pandemic situation of the COVID-19 virus. The Team consists of:

- **Eyrún Halla Haraldsdóttir** – Biomedical Engineer - Allrounder
- **Thomas Ziegler** – Corporate Communications - coordination and graphics
- **Roman Bättig** – Application Engineer- Wireframes
- **Renato Kälin** – Elementary School Teacher - Use cases and first-hand input
- **Luciano Rod** – Idea inputs
- **David Caspar** – Back End Developer/Software Architect - Architecture design
- **John L Diaz** – Front End Developer - Front End development
- **Andrea Ferrazzo** – Economics and Management Student - Allrounder
- **Ali Nasserzadeh** - Lead Engineer, Full Stack development - Initial idea support and front end development

# Project description

In order to prevent and diminish the spread of the current pandemic, governments and officials have implemented various measures to prevent the assembly of larger groups of people, one of them being locking down schools. As education is one of the fundamentals of society, it is of great importance that it can be continued in a remote manner. With that, new challenges emanate, how can teaching still continue in a reasonable way, especially for younger children? Luckily with all the technology that we have access to these days, remote communication is very accessible. Nevertheless, teachers and students/parents alike struggle with adapting to the new format and the existing tools and platform for remote teaching are not able to meet some of the needs in a practical manner. First of all, communication out of one single platform is not feasible, calls, notes, and tasks are disconnected. Secondly, many platforms are overcomplicated, there are different tools for the same problem with the same stakeholder, which causes confusion and important notifications/ information to get lost or miscommunicated. Thirdly, giving out/ receiving homework gets overcomplicated, having to download the document - print it out- scan it in– upload it again, causing substantially more time for the task and is unfeasible for students that don't have access to all the resources needed. Finally, the student/ teacher interaction during live classes is challenging, as the normal raise hand and engaging in educative and creative discussions is difficult over live video feeds, students might be discouraged and unmotivated to participate and even in these sessions the parent's involvement might be excessive, causing the child to rely too much on them and not entering the discussion on the basis of its own integrity.

Noticeable, there is a dire need for a more practical solution, one simple and user-friendly platform for communication with peers, students, and parents. Students should be able to resolve a problem, upload their work and teacher should be able to correct mistake directly on this upload

# The way to the solution

## Existing solutions

When looking at some of the already existing platforms for video conferencing, e-learning and document sharing i.e., there are many good solutions that can be used as inspiration or as an add-on for the solution needed. Following is a list of some of the more favored solution with a short description and notes on pros and cons for that solution as an educational platform:

| Platform | Description | Pros | Cons |
|---|---|---|---|
| **Zoom** | Easy to use video conferencing platform. It can integrate with other educational platforms. | + Simple interface<br>+ Screen Sharing is easy<br>+ Raise hand option<br>+ Good audio and video quality | - Not optimal for document or content sharing<br>- Limit to 40 min calls with a free plan |
| **Microsoft Teams** | Advanced video conferencing tool with many team collaboration tools. | + Easy to create subgroups<br>+ Document and content sharing is easy<br>+ Platform that holds together all documents | - Many add ons, not very intuitive, longer learning curve.<br>- Need to have an MS account<br>- Notification control needs improvement |

| | | | |
|---|---|---|---|
| **Slack** | Team collaboration platform for simplifying communications. | + Separate channels<br>+ Easy to use<br>+ Easy to share files | - Not a video conference tool<br>- Expensive paid plan |
| **Second life** | Virtual education solution using 3-D virtual world as an platform | + Interactive<br>+ Gives the student a sense of presence | - Major learning curve<br>- High speed internet needed<br>- Privacy issues |
| **Khan Academy** | A non-profit organization, offering free online content for students. | + Free and no need to download an app<br>+ High quality and well-explained content | - Not an interactive tool<br>- Limited by language |
| **Brilliant.org** | El-earning platform focused on math, science, and engineering. | + Interactive and visual examples.<br>+ Alternative approach to solve complex problems. | - Not an interactive tool<br>- Limited by language<br>- Not free |
| **Padlet.com** | An application that is easy to use to create an online bulletin board to display information on any topic | + Easy to use<br>+ Interactive<br>+ Great for sharing content | - Limited Padlets for the free version, which is small.<br>- Not for video conferencing |
| **Google Hangouts** | Communication platform with video call capabilities and messaging. | + Easy to use<br>+ For free<br>+ Good for basic calls used with google docs | - Need a google account<br>- Not optimal for document sharing |

| | | | |
|---|---|---|---|
| **Moodle** | An open-source Learning Management system. Used widely by educational institutions. | + Good for providing course content<br>+ Useful for uploading assignment<br>+ Customizable | - No live interaction or video conferencing<br>- Major learning curve when using for the first time<br>- Outdated interface |
| **Kahoot** | A game-based learning platform. Focused on making it easy to create, share and play learning games. | + Fun and interactive<br>+ User friendly and intuitive<br>+ Scoring based on how fast you answer | - No video conferencing tool<br>- Online reports limited in free version<br>- No uploading function<br>- Limited types of questions |

## Use cases

In order to meet the problems that teachers/ students are experiencing with the current solutions with a systematic manner, a list of use cases based on first –hand experience was created:

| **Use Case** | **User Story** | **Use Case 1** | **Use Case 2** | **Use Case 3** | **Use Case 4** | **Use Case 5** |
|---|---|---|---|---|---|---|
| **LP01** | Login | Login mit Email (Beispiel) | Passwort vergessen | Info | | |
| **LP02** | SuS/LP erfassen | Manuell | Excel / Schul-Export | Info | | |
| **LP03** | Klasse verknüpfen | Bulk (im Verband) | Einzelne Mutation | Info | | |
| **LP04** | siehe LP03 | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **LP05** | Individ. Links | Generieren | Mail Individ-PDFs | Save individ. PDFs | Save Class-PDF | Info |
| **LP06** | **see LP05) | | | | | |
| **LP07** | Schulbeginn | Signal "Starts soon" | Start Live-Stream | | | |
| **LP08** | **See LP07 | | | | | |
| ****LP019**** | Document Push | Send files to class | Send files to Individ. | | | |
| **LP09** | SuS see LP | Videostream of LP | Screenshare | View of specific SuS w/ Voice | | |
| **LP10** | Supervising | Chat-Roulette to see SuS | View of spec. SuS | Unmute SuS (Default Mute) | | |
| **LP11** | see LP10 | Unmute | | | | |
| **LP12** | Peer-View | Unblock Peer-View | Breakout room for Groups | | | |
| **LP13** | React to Hand raise | Alert "Question" | List of all Hand raise | | | |
| **LP14** | React to Question | Share SuS-Vid on Screen | Private Question (Mini-Session - all Mute for other SuS) | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **LP15** | Poll Unterstanding | Start Poll (Understand 1-6) | See Summary Responds* | Drill-Down to individual | *Share screen to show Class | |
| **LP16** | Poll Binary (y/n) | Start Poll (y/n) | See Summary Responds* | Drill-Down to individual | *Share screen to show Class | |
| **LP17** | Lesson End | Signal "Lesson End" | | | | |
| **LP18** | After Lesson | Timestamps f/ Highlights | Export Highlights (.mp4) | Export Highlights Doc-Template | Export other Meta Data | |
| **LP19** | **see above** | | | | | |
| **LP20** | Doc-Cloud | TBD | TBD | | | |
| | | | | | | |
| **SUS01** | individ. Link | Erhalt per Email | Login | BOOKMARK-TUTORIAL? | | |
| **SUS02** | 1st SUS Login | WALKTHROUGH *ANIMATED-MASCOT* | | | | |
| **SUS03** | Out-of-Class | See Timetable (St. Soon/end) | | | | |
| **SUS04** | In-Class | Raise Hand | Download Resources | | | |

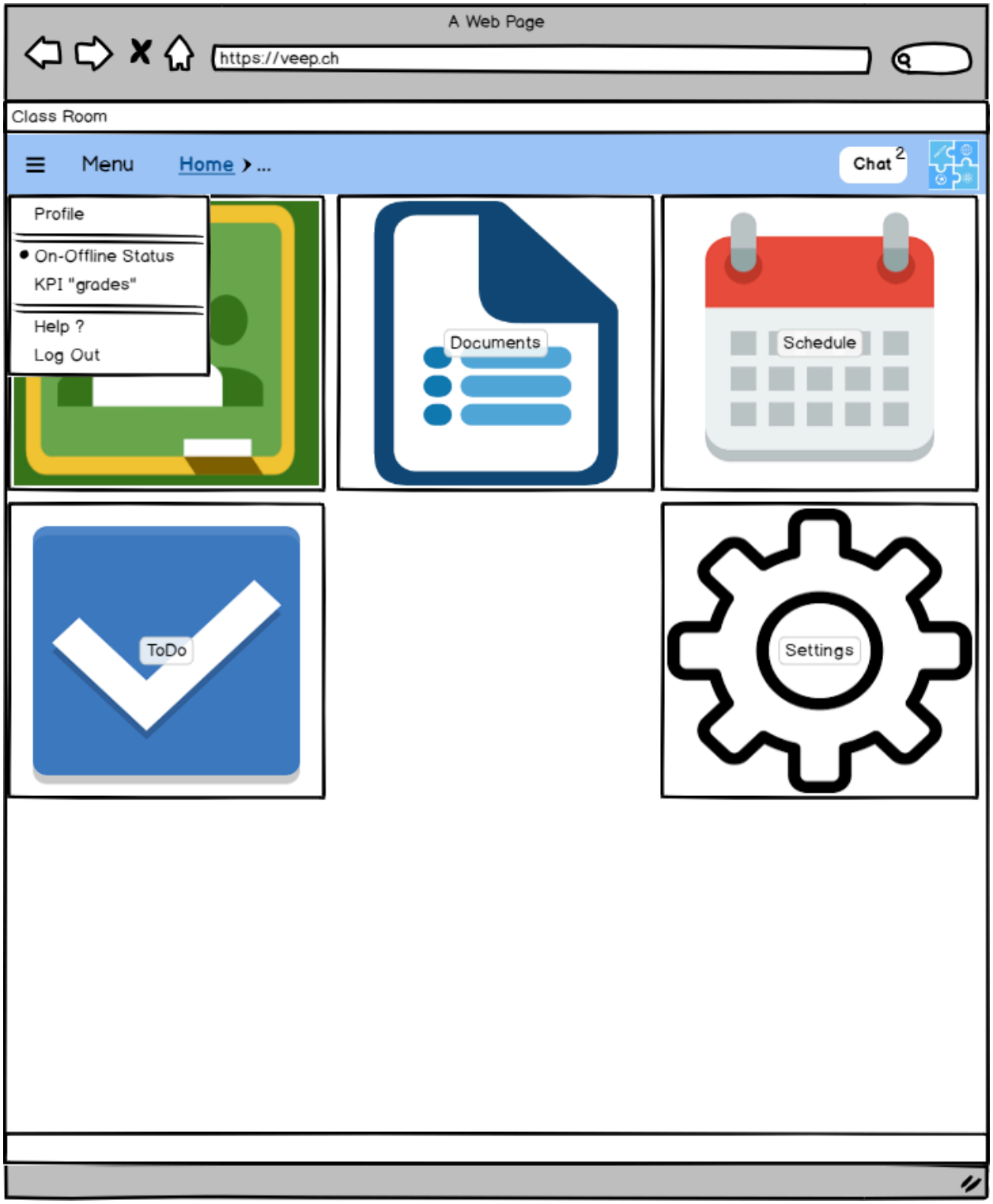
# Our solution: Veep

## General Application Design

The following image shows the main screen of our application as seen by teachers. For students, the screen looks similar, but the number and kind of displayed components/tiles might vary. From this screen, users can access all the various application parts that are bundled into Veep.

As indicated by this image, we foresee that Veep will at least feature the following components:

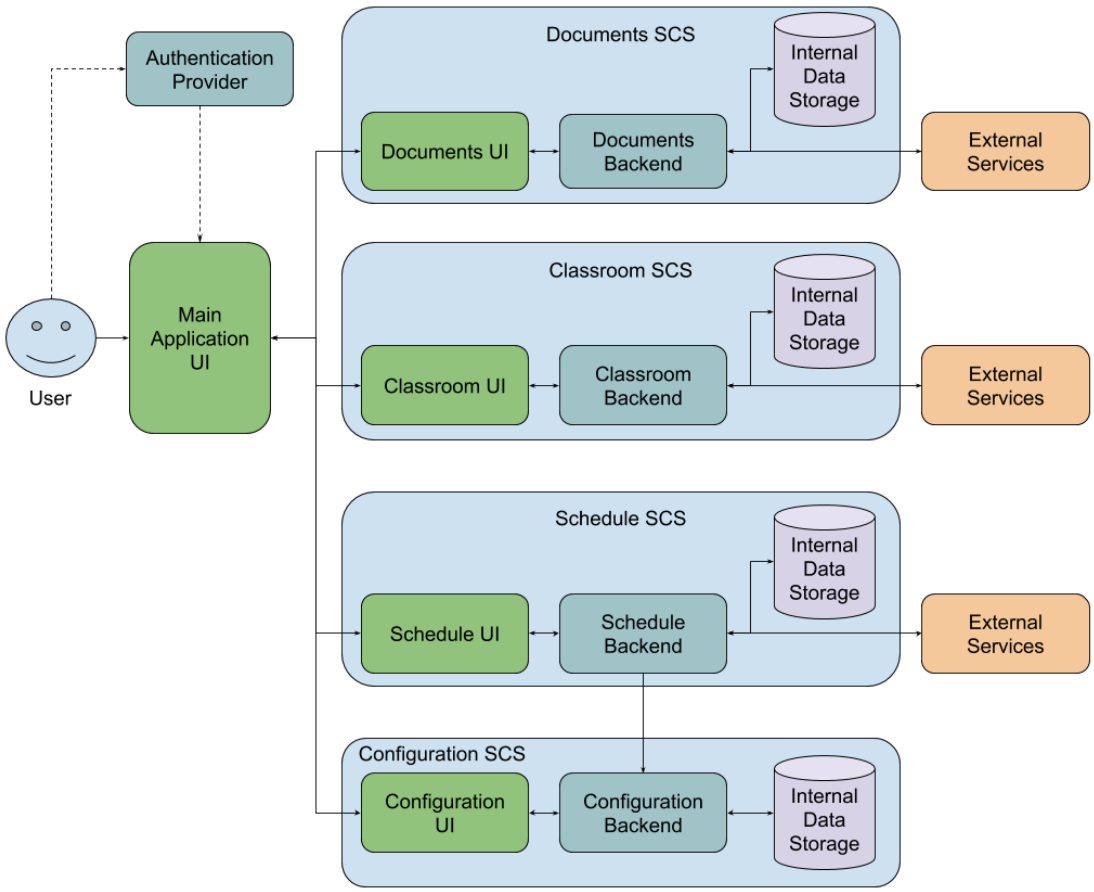
- a schedule showing for teachers and students when they attend which class;
- a classroom mini-application allowing students and teachers to interact with each other by video conference or by text messages;
- a document store allowing teachers and students not only to share learning material but also to hand in and return assignments;
- a todo list showing current tasks; and
- for teachers a configuration section where they can set up classes and learning materials, using an easy and intuitive drag and drop system with predefined modules such as video podcasts, discussion panels, quizzes, etc. The user interface should have the look and feel of a movie editor.


A Web Page
https://veep.ch
Class Room
Menu
Home › ...
Chat 2
Profile
On-Offline Status
KPI "grades"
Help ?
Log Out
Documents
Schedule
ToDo
Settings

## Veep Application Architecture

The general architectural style of Veep is the so-called [self-contained systems (SCS)] style. While discussing the system's functionality, we noticed that most application components like the video chat or the document store are very independent of each other and actually form complete web applications on their own. For instance, the component to manage documents is valuable to students and teachers without using any of the other components like the video chat or the schedule. The same can be said of all the other main components.

Thus, we arrive at an overall system architecture that looks as follows:



Each application component forms one self-contained system, providing the user interface as well as the related application logic, data storage and integrations with external services.

In addition to these self-contained systems, there are things like security, user authentication or regular backups that concern all SCS. For these matters, dedicated services should be used as indicated in the above diagram by the Authentication Provider service.

The whole application is held together by the main UI component, whose responsibilities are to make sure that

- each user is authenticated via an authentication provider (although the UI does not know the details of the authentication mechanism and should be involved as little as possible into this matter), that
- each user (students, teachers) sees the main components of the application and that
- each reference pointing to another component/SCS is resolved correctly and transparently (but again, the UI does not know the technical details and should be involved as little as possible)

## Specific Application Parts

### User Interfaces

As the previous section describes, all SCS feature their own user interfaces, which can thus be highly optimized, especially in terms of user-friendliness. Each SCS is free in terms of front-end technologies it uses given that the resulting UI fits into the overall user experience. The main application UI will then combine these different components into a user-friendly whole.

### Backend Systems and Data Storage

As indicated by the architecture diagram above, each SCS has its dedicated backend and data storage. Each backend exposes an API suitable for its front-end, and the data storage is tailored towards the needs of its SCS. If an SCS needs to communicate with another one, it happens by calling an API designed specifically for this purpose. No SCS is allowed to call other services than this API or access the data store of another SCS directly.

### External Services

If feasible, SCS may use external services, e.g. for data storage, data processing and the like. It is up to the SCS to decide which external services it uses as long as these external services fulfill general requirements imposed by the whole platform.

## Cross-SCS Concerns

So far, we identified the following cross-SCS concerns, and there are likely others which we would need to identify at a later stage.

### Intra-SCS References/Links

One crucial aspect of Veep is to be able to reference SCS-specific artifacts like documents, messages, etc. from other SCS. For instance, it should be possible to reference documents containing homework assignments from the schedule. In order to achieve this, there needs to be an overall mechanism to interpret and react to references/"links", and each SCS must be able to understand and handle users clicking on links.
One way to achieve this is to have a common convention about how links/references look like, e.g. a specific URN format. In addition, it might be necessary for the main user interface to detect when a link has been clicked and to update certain application components as their interface state has changed.

In an application like Veep, it is crucial to protect user data like assignment grades and the like. As indicated by the architecture diagram, there should be a strong, reliable mechanism to authenticate students and teachers. How such a mechanism could look lies beyond the scope of this overview, but it is clear right from the start that such a mechanism is necessary to fully unleash the potential of the Veep platform.

# Vision

With a fully developed, easy to use educational platform, remote education will become more accessible. It can be used for various stages of education and even after a lock-down has been alleviated to facilitate remote teaching. Furthermore it will not only be restricted to Switzerland but as a worldwide used application.

## Challenges

We have developed a strong concept supported by application architecture and wireframes. There is potential for a future application. Data security and legal conditions have to be analyzed further. Develop integrated video chat and file storage independent from the APIs of big corporations. Develop further the front-end in order to provide the best user experience tailored for the needs of children (fun and active learning) and teachers (intuitive and quick to use) while keeping in mind the simplicity of the platform. Finally, the project has to be presented to the Swiss Government and Cantons in order to receive the necessary funds to continue to develop the project. Crowdfunding could also be a viable solution.