Contents

1 Introduction

The examination process is often a **tedious task** for those who are in charge. Great amounts of time go into organizational problems and overhead. Digitizing exams would resolve many of these problems. A step towards electronic examination would make the process more flexible, scalable and resource-efficient. Meanwhile, leading to a more accurate depiction of a students' competence.

Digitizing exams is no new idea. Although, many concepts [cite papers that talk about byod or multi media room exams] and implementations focus on conducting e-exams in the same environment as *paper-based exam*. Resulting in exams that are either conduced on the universities hardware [@KlausurenAndDerUniMainz]. Alternative the exam is a so called **BYOD** (bring your own device) exam, where the exam taken student owned devices [@GLM2015_RobertPeregoodoff].

With the the current COVID-19 situation decentralizing exam would be of both economical and Technically the

It is important to notice, that *decentralized e-exams* differ from *paper-based exams* and even from *centralized e-exams* in some key points. Foremost, the examiner has less control over the environment the exam is taken under. This raises questions about exam integrity and fairness. These questions must be addressed through careful conceptualization of questions and intelligent software design.

Of course some *e-exam* are already conducted today. Large proportions are making use of a **proctoring system**. In such a system a supervisor can access the examinees' device, can monitor all their activity and will watch them through their webcam. This proctoring process is costly. It hardly scales and still easily can be fooled.

Further, test-taking applications are found in many *LMS' (Learn Management Systems)* such as [@Ilias], [@Moodle] or [@Blackboard]. Unfortunately, most often these applications focus on student self-assessment. They also majorly vary in quality and utility. As they are integrated in a complete LMS, changing to the *best* implementation is in many cases not an option.

Last, as exam data is highly confidential, there is a strong argument to be made against closed source solutions. It is crucial to know exactly how the used application works and how data is handled. Adding, open source projects are less prone to major security issues as the development can leverage the crowd sourcing capabilities that an open source system provides.

2 Requirements for E-Exams

We find e-exams to be advantageous in a variety/numerous of ways. Still, e-exams are only valid if they can meet the same requirements that are asked for in paper-based exams. [@Handke2012] provides suitable requirements. These are limited to topics the e-exam software can directly influence, issues concerning exam content are not being considered. These requirements can be divided into three broad categories:

The fist requirement defines the desired outcome of an exam:

- **General Validity.** Exams should aim to provide an accurate depiction of an examinees competence level.

Requirements that mainly influence interactions of examinees and examiners with the examination system:

- **Protection against contestation.** No formal, or technical deficiencies should occur, that would question the validity of the exam.
- **Equal Treatment.** Individual examinees must be treated equally.
- **Protection against cheating.** Exams must be protected against manipulation of the exam outcome by examinees.
- **Transparency.** The examination process and results must be understandable and verifiable.

Requirements that mainly influence the technical implementation of how the examination system handles data:

- **Protection of Data.** Data of examinees is personal data, as such it must be protected from misuse.
- **Integrity.** Exam data must maintain consistency, accuracy and trustworthiness throughout its entire lifetime.
- **Attributability.** One taken exam must uniquely map to one examinee and vice versa.

These categories set a general framework of how to design any examination system. In case of this thesis, e-exams are of interest. They use specific design principles to match the previous requirements. These design principles will be introduced in the following.

2.1 General Validity

As the [@ETH] states in **their** paper, examinations should support the purpose of universities to produce highly capable individuals. The measure of degree to which students succeed in that aspect is largely based on their performance in exams. Subsequently, students are highly incentivized to focus their studies on a exam format and specific question types. This interdependency between knowledge acquisition and examination shows the importance of exam design and poses the question of what and how to test. Different question types are particularly well suited to test specific aspects of learning. These questions types can be defined as follows [@ETH]:

- **(Semi) Closed questions**, which mainly revolve around the demonstration of *factual knowledge*.

Solutions are not disputable, there are only right and wrong answers. Typical formats include multiple-choice answers or simple text input. *E.g. "What does* BYOD *stand for?"*

- **Competence questions**, which are suited to test for a certain *practical skill*. Solutions are given in form of an implementation of the specific task at hand. *E.g. "Using the provided software, implement an e-exam about e-learning."*
- **Essay-type questions**, which are suited for assessing *transfer knowledge* and *understanding*. Solutions are given by free text input. *E.g. "Explain why subjects in computer engineering are especially well suited for e-exams."*

Further, different degrees of allowed aid can be identified: In open book exams, students are allowed to solve the question at hand using any resource they feel they need. These open book exams rely mostly on both competence and essay-type questions. It could be argued that these types of questions resemble a real world scenario in which access to information is rarely limited. Meanwhile, in such open book exam situations, closed question are rendered insignificant as simple factual knowledge is easily accessible. In order to ask closed questions it is therefore necessary to restrict access to any aid.

Classic paper-based exams do not provide a feasible way of combining degrees of allowed aid. Therefore, some question groups tend to be neglected. This constraints possibilities to create an accurate depiction of an examinees actual competence. With e-exams on the other hand, we can implement such a varying degree of usable aid, creating a *partial* open book exam. This can be achieved by letting students generally use any resource they need in order to answer the question. Additionally, we introduce per question time constraints. These time constraints can be adjusted according to the question and question type. Leaving closed questions with a strict time constraint and creating an *either-you-know-it-or-you-don't* situation, where the student has no time to look up any solution. Essay-type questions just as competence questions can employ more generous time frames, giving the examinees freedom to make use of their tools.

Ultimately, examination software in general has no direct impact of what exact questions the examiner asks. The content of a question obviously predefines how well this question can predict an examinees' capabilities. Still, the use of a partial open book exams mode allows for a diverse set of question. This mode allows to test factual, transfer and practical knowledge to an equally valid degree.

ADD DESIGN PRINCIPLE

2.2 Protection against contestation

Generally, contestation of entire paper-based exams is not a common problem. This is mainly due to the controlled environment paper-based exams are taken in. Adding, the medium that is used to test examinees (i.e. paper) is fail-safe. E-exams, especially decentralized ones, introduce the possibility of failure of the exam medium. They rely on software, on the operation of the exam device and of course on internet connection.

The most crucial point is the reliability of the e-exam software. As with any software, high reliability can only be achieved by rigorous testing and continuous improvements. In the end, it is more valuable to have a working system that lacks "quality of life" features than software full of bugs.

Another important point is device operability. Decentralized e-exams are taken on the examinees device.It largely lies within in the responsibility of the device owner to assure it is working as intended. Still, software can assist by advising examinees to keep their devices updated, plugged into power and not to take the exam if they fear device failure.

Software-wise these directives can be supported by making the exam accessible and workable if the internet connection is lost for short periods of time. Additionally, exam answers should continuously be sent to a server to minimize the risk of data loss. In case of both a device crash and internet failure, the exam should persist on the local storage of the device. The device then can be rebooted and the exam can be continued. These measures combined with the fact that modern devices generally show low failure rates assure that the exam environment is fairly save and contestation therefore unlikely.

2.3 Equal Treatment

Equal treatment of examinees should be ensured throughout the entire examination process, reaching from taking the exam to its correction.

Possible inequality arises in some key areas. In BYOD exams student devices are largely heterogeneous– they run different operating systems and consist of different hardware. This fact should not lead to different exam-taking experiences. The choice of hardware should be largely irrelevant. Consequently, it makes little sense to develop proprietary software for each operating system. Modern web technologies provide a common language among different systems. Web applications do not lack speed or functionality and can be adopted cross-platform. The software is hosted at a central entity where it can be maintained and improved. The software artifact is then delivered via a modern browser.

The process of correcting exams is also an area where possible inequalities can be found.

> **Especially, in paper-based exams checking an exam for correctness is one of the most time-consuming processes in conducting an exam.**

Resulting in fatigue and thus sometimes in answers checking mistakes. Besides accidental mistakes, [@James 1927] has found negative bias in connection with bad handwriting. He found students with bad handwriting get categorically worse grades than students with better handwriting.

By using e-exams these inequalities can be eliminated. First, some question types, such as multiple-choice questions can be checked automatically. This is an immediate improvement over correcting these questions by hand. Second, as exam answers are available in digital text, reading and checking answers is easier. Answers must not be deciphered, correction of exams can be done faster. Meanwhile, e-exams can also eliminate biases against certain students.

2.4 Protection against cheating

When thinking about any assessment the consideration and handling of academic dishonesty (e.g. cheating in an exam) is one of the most important parts. Moving from paper to e-exams poses the question what parts – if any – must be adjusted to accommodate for changed circumstances and environments.

In his paper [@McGabe] poses seven fields of possible cheating in exams which he then evaluates by occurrence and perceived severeness. Six of which are relevant for this thesis' purpose ˆ(The seventh would be *"Using false excuse to delay test taking"*). The fields can be described as follows:

Student cooperation:

- **Knowing the questions.** Learning about the exam content from someone who has already taken it.

- **Cooperation with outsiders.** Reseaving help from someone cheat on an exam.
- **Cooperation with fellow examinees.** Copying from another student during an exam with their knowledge.

Use of disallowed aid:

- **Exploit environmental circumstances.** Copying from another student during an exam without their knowledge.
- **Use of unauthorized notes.** Bringing prepared cheat notes to use in the exam.
- **Use of electronic, unauthorized aid.** Using search engines or the lecture material to solve questions.

Before thinking about how to obviate these cheating scenarios an important statement must be made: Cheating cannot completely be eliminated. There are always means for students to engage in cheating. E-exams cannot change that, but compared to paper-based exams some measures against cheating may be more effective.

Knowing a question. The generation of questions is a time-consuming process, thus a strategy may be to keep questions as secret as possible and reuse them throughout multiple exams. This is a rather ineffective strategy as platforms such as [@studydrive] often provide comprehensive protocols from memory of examinees who have engaged in a given exam. E-exams can choose a different approach. Instead of having few questions and keeping them secret, e-exams can leverage large question pools. At a certain point it becomes unfeasible for students to prepare for every available question as question pools grow larger. The digital nature of these questions makes them easily shareable, allowing question pools to grow faster.

Cooperation with other examinees. For closed questions this cooperation can be prevented by using strict time restrictions. As already stated above, these questions fall in the category *either-you-know-the-answer-or-you-don't* there is no need for a lengthy reflection period. As time constraints are tight, there is no way of communicating with others and solving the question. For more open question types the time limitation is not as tight. At the same time, answers require more in-depth considerations. To ensure that students write down their own ideas and cannot share their thoughts, the “copy” and “paste” functionality can be disabled. Further, e-exams can easily be randomized, thus preventing students from sharing solutions.

Cooperation with outsiders. As decentralized e-exams are not conducted in a controlled environment, cooperation with outsiders becomes a severe problem. Examinees could try to take the exam in the presence of an expert. Some try to solve this problem by using proctored e-exams. These exams use live surveillance through webcam and microphone that is evaluated by a person watching in real time. This approach hardly scales as for every 4-5 students a supervising proctor is needed. Programs like [@ETC_Toefl] can make use of such a system, as their high test fees leave room for additional expenses.

Although, live surveillance of students is not a valid option the psychological effects of being monitored can be leveraged. A measure might be to make use of integrated webcams and microphones of the devices at hand. This video and sound data can be reviewed if needed. More importantly, it creates a mental barrier. If examinees really commit to engage in academic fraud they will most certainly find a way to do so. The goal is simply to prevent those from cheating that would only cheat if they would feel no threat of being caught. The sole existence of any measures makes students behave more honest. This can be compared to, surveillance makes crime less common at public places [@Welsh2004].

Exploit environmental circumstances. (e.g. Peeking at answers) Again randomization can solve this problem. As questions appear in a different order for each student, even multiple-choice questions cannot simply be copied.

Use of unauthorized cheat notes or electronic aid. Following the argument made about partial open book exams we find that besides time constraints no additional measures must be enforced. Cheat notes are redundant if there is no time to use them.

We find these cheating scenarios to be largely managed by e-exams. Still, as specific software is in use, the degree of cheating must constantly be assessed. Further software bugs must be fixed, while security flaws must be identified and resolved.

2.5 Transparency

The examination process should be transparent for examinees. Students must be able to understand their mistakes and shortcomings. This implies that the exam software provides ways to give feedback. Further, as examiners are not free of mistakes, corrections can sometimes be faulty. Well implemented transparency allows students to review the examiners correction and contest against individual corrections. Important to mention is, that every student should get the chance to review their exam. The digital nature of e-exams makes this degree of transparency easy to realize. Sharing a corrected digital copy of an exam, allows examinees review their answers and understand their gaps in knowledge. Contestation against certain questions could also be processed within the exam software.

2.6 Attributability, Protection of Data and Integrity

Exam data is highly sensitive and demands high levels of information security. As with any information system, basic information security principles apply. The following points prove to be of special importance.

Exam data must be uniquely traceable to examinees. This can be easily realized by having examinees log into an user account before they can perform any action. Examinees either get an unique identifier in-software or an unique identifier that is provided by the testing authority. Any of their actions is then linked to their user id.

To assure solid data protection, strong user rights management must be enacted. This guarantees that only authorized groups can view or correct exams. In this way data is largely protected from missuses.

This measure ties into the integrity of exam data. As access is restricted exam data cannot be changed. To provide even more security, answered questions can be sent to a central server instance as soon as students continue to the next question. Further, frequent database backups of the exam data should be standard procedure.

Another consideration to take into account is the availability of the source code. Processes should be completely transparent and comprehensible. Exam authorities should be able to host exams . This can be achieved by providing the exam software in open source format. Further open source programs can leverage crowd participation to render software bug-free and to eliminate existing security flaws.

3 Designing the software artefact

In the previous chapter we found no exam software fully matching all requirements. In the following an overview of how a software artefact could implement all these design principle from a technical point of view will be made. Further, as a proof of concept two design principles are implemented in a working software prototype. This prototype helps determine software architecture as well as used technologies. First, general thoughts about data structure will be made, followed by an short introduction into used technologies. All these statements will be made with the exam requirements in mind.

4 Data structure and architecture

The artefact is classic client-server application. With a frontend (client), responsible for interaction with both examinees and examiners and a backend (server), responsible for data handling. The client and server side communicate over a simple REST API. The advantage of a client-server application lays in the separation of the data and the end-user. Users do not have direct access to data but any read or write action must be made over and API. Here permissions can be checked and possible misuse inhibited. Second, as the user has no direct influence on the servers actions the sever can act as a source of truth. Data that was once committed to the server is now immutable. For example, the foraging of exam answers after submission becomes impossible. The API can also provide different kinds of endpoints to interact with for different kinds of users. Examiners for example can use one API endpoint to create exam questions, whereas students will get an error code if they try to interact with it.

One of the most important design considerations is the data model that is used to store and access exam and user data. There are four central instances: First, there is the question instance. This is probably the most important instance. A question consists of a title, the question type, the question text, the questions points and the questions time limit. Further each question has a question body, the shape of the question-body depends on the question type. For example the multiple-choice question-body consists of a reference to the question it belongs to and a selection of possible answers. For free-text questions currently no additional body is need, still to be consistent in data structure and to allow for later additions, free-text questions also have a body. The second most important

First, there is the user, representing either the examinee or the examiner. Users are uniquely identified by a id. This id is provided by the application but could also be an identifier provided by the testing authority. The second instance is

Both the server and the client side are written in a code language called JavaScript. It is the most popular language on [@Github https://octoverse.github.com/#top-languages]. JavaScript allows programmers to realize the complete technology stack with one language, making it a compelling language to write an application in. Besides many modern and popular libraries for web development are written in JavaScript. Some of these libraries also find use in this artefact, the most crucial being [@React] and [@Express].

[@React] is “a JavaScript library for building user interfaces”. It uses structures that are devidable in reusable components. React makes it easy to create complex application instead of only simple websites. It was orignally created by [@Facebook] and finds it use in the tech-stacks of [@Uber], [@Airbnb], [@Netflix] and many more.

[@Express] is a common library to create backend services with. It is lightweight and allows for the creation of both simple and complex APIs. The express server also handles data storage, for this purpose a database is connected. The artefact uses a noSQL database called [@MongoDB]. [@MongoDB] does not store data in tables but in JSON like documents. As the JSON format is inspired by JavaScripts objects, the data structure used in the frontend part of the application thus directly translates to the data structure that is used to store the given data.

General Validity: Per question time constraints **Protection against contestation:** Offline Capable **Equal Treatment:** Digital answers and device indifference **Protection against cheating:** per question time constraints. video and sound surveillance. use of large question pools **Transparency.** a way to give feedback. availability. **Protection of Data.** **Integrity.** **Attributability.** User Permission Management. Authorship of Actions. Open source ness