MSc thesis

Master's Programme in Computer Science

Data privacy in software design – case dLearn.Helsinki

Heikki Ahonen

November 5, 2020

Faculty of Science

University of Helsinki

Supervisor(s)

Prof. T. Männistö, Dr. P. Dexter

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Heikki Ahonen

Työn nimi — Arbetets titel — Title

Data privacy in software design – case dLearn.Helsinki

Ohjaajat — Handledare — Supervisors

Prof. T. Männistö, Dr. P. Dexter

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
MSc thesis	November 5, 2020	35 pages, 10 appendice pages

Tiivistelmä — Referat — Abstract

The research group dLearn.Helsinki has created a software for defining the work life competence skills of a person, working as a part of a group. The software is a research tool for developing the mentioned skills of users, and users can be of any age, from school children to employees in a company. As the users can be of different age groups, the data privacy of different groups has to be taken into consideration from different aspects.

Children are more vulnerable than adults, and may not understand all the risks imposed towards them. Thus in the European Union the General Data Protection Regulation (GDPR) determines the privacy and data of children are more protected, and this has to be taken into account when designing software which uses said data. For dLearn.Helsinki this caused changes not only in the data handling of children, but also other users.

To tackle this problem, existing and future use cases needed to be planned and possibly implemented. Another solution was to implement different versions of the software, where the organizations would be separate. One option would be determining organizational differences in the existing SaaS solution. The other option would be creating on-premise versions, where organizations would be locked in accordance to the customer type.

This thesis introduces said use cases, as well as installation options for both SaaS and on-premise. With these, broader views of data privacy and the different approaches are investigated, and it can be concluded that no matter the approach, the data privacy of children will always prove a challenge.

ACM Computing Classification System (CCS)
Computer Science → System Architecture → Case study
Legislation → GDPR → Data privacy

Avainsanat — Nyckelord — Keywords

data privacy, computer science, gdpr

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software Systems study track

Contents

1 Introduction

Working life competence skills are needed in modern working place, and the need for those skills is increasing rapidly [12]. Skills such as analysing complex issues, understanding diversity and ability to act intentionally in changing contexts are included in working life skills. Learning these skills requires knowledge of the current level of the learner, as well as ability to follow the change in the level of skills [5]. The research group dLearn.Helsinki aimed to solve this problem.

The research was a collaboration between the Faculty of Educational Sciences and the Department of Computer Science. The former created the research tool, or the questions, surveys and result patterns, whereas the latter created the software for the tool to be used. The software was aimed for schools, universities and companies, and the research subjects were the pupils and employees, and teachers and employers were considered as team leaders in the research perspective.

Originally the software was completely anonymous, and users were given usernames, which were built from group specific prefix text and indexing numbers. However, while testing the software with actual users, also known as a pilot phase, the feedback from users and team leaders alike led to usernames being based on users actual names. This meant changing the software from anonymous to pseudonymous [1]. This was especially essential in schools, where young pupils had difficulty remembering their usernames [10], and teachers spent much time solving this issue.

Furthermore, the software was only provided as a Software as a Service (SaaS) solution, meaning it was hosted by the research group, and only one instance in the world existed [11]. Requirement for an on-premise version was also brought up during the pilots, as schools and workplaces would want to have the software in their own private servers and networks. A solution for this kind of installation was thus also examined.

From this premise I started my design science case. As the usernames were transformed to real identifiable names, the question of data privacy became apparent. In European Union, the General Data Protection Regulation (GDPR) regulates the usage of personal data, including a person's name [6, 14, 1]. Also in the regulation, the safety and security of children's data is more protected than adults, due to their age and possibility to understand the consequences of their data usage [14].

In this thesis, I present the results of my design science case, especially the artefacts created during the process. The artefacts include the use case scenarios and their development during my work, and a suggestion for future use cases. Another artefact is an installation tool, which aims to provide a solution for customer based installation.

The rest of the thesis is organized as follows. In Section 2 I introduce the research methods and take a closer look into the research questions on which the design science case is based. In Section 3 I present the background information, such as the privacy legislation. Section 4 handles the different user groups and their impact on the software design. In section 5 I present the design science artefacts. Sections 6 and 7 contain discussion and conclusion of the thesis.

2 Methodology

In this section I introduce the research methods used in this thesis. The research questions with the research hypotheses are also discussed in this chapter.

2.1 Research methods

For my thesis I used two different methods of research. For the background information I gathered information from literature. The more laborious part of this thesis was design science, with which I developed artefacts for development.

2.1.1 Background literature

I used literature for my background information and to find out if there are any previous cases of similar problems. The GDPR was a substantial part of the literature, as it was also a major source for most if not all current publications.

My original question was *("data privacy" OR "GDPR") AND ("software architecture")* but this proved to be too wide of a question and didn't answer the problem setting I was hoping for. However I had to create another question to find more relevant articles and came up with a new question: *("GDPR" AND "child").* This too was too wide as a search criteria, so I had to do manual search within the search results.

With these questions I could find articles quite relevant to the issue, even though the questions remained on broad level. With narrower questions, however, I could not find topical research, as the time window in which the GDPR has been in force is narrow. Because of that, I narrowed my search down furthermore with date restriction so all the articles were recent enough that the GDPR had either been published or even already in force. The first level of narrowing the amount of articles was done manually by searching the topics and the abstracts of the articles for their relevance. From this the articles were narrowed down even more by actually reading the articles themselves and finding the relevant articles.

From the articles gathered with these search criteria, I widened the search to find more

background. With this snowballing I found some articles that were relevant to the issue, but might be from before GDPR.

2.1.2 Design Science

I did my design science case working as a member of the research group dLearn.Helsinki. A design science case aims to create a design artifact for the software [4, 15]. This type of research is more widely used in engineering fields, but less in empirical research [15]. This is due to the structure of design science, where the aim is to create something more concrete [4, 15], rather than mostly theoretical approach of understanding reality [15]. Design science is still a valid method of research, where the research can be defined in six steps [15]:

1. Problem identification and motivation.
2. Define the objectives for a solution.
3. Design and development.
4. Demonstration.
5. Evaluation.
6. Communication.

The steps might have different names depending on the source [4, 15], but the basic concept is the same in different variations. First the problem has to be identified, or define the starting point of the research. Next step is to define the goals for the artefacts and solutions. Third comes the design and development, or the creation of the artefacts. Demonstration is to prove the use of an solution, does it actually solve the problem. Evaluation is for comparing the achieved results to the objectives set earlier, or does it actually reach the objectives set. Communication is where the case comes to close, and the results are communicated [4, 15], in this case via this thesis.

2.2 Research questions

The software dLearn.Helsinki is used on both children and adults, and the data privacy laws and permissions differ from one another [13, 14, 20, 10]. In the original state of

the software there was no difference between the two groups, but due to changes in the user recognition level, some differentiation is needed. In this section I introduce the research questions based on this premise, as well as the underlying assumption for each question.

Research question 1: How does the data privacy of children differ from that of adults, according to General Data Privacy Regulation (GDPR)?

The assumption is that the data privacy permissions for children's data are more restrictive than that of the adults. We cannot consider adults as one group, either, but we have to take into consideration several roles for them, as well. For example, the data of an employee can be used more freely by their employer, as usually some of the data is required by working contract [17]. In the GDPR and its recitals, there are multiple restrictions on how the children's data could be handled [13, 14]. On the other hand, there are notations on how the data of an employee could be handled more freely by the employer, due to employment contract [17]. The major difference between the most restricted and the most allowed groups causes an issue in the software, as at the beginning of the design science case they were treated equal, and the difference is handled manually and the data is processed on paper.

Research question 2: How does the difference affect the use cases in the software under inspection?

The assumption is that the permission to use data of under aged children has to be granted by the guardians of the children [13, 14]. As such a feature is not implemented in the software, this is one part of the design science case and requires study and design of new use cases as well.

This also requires handling and setting security levels for both children and adults, as in the original state of the software there were no different security levels depending on the user's age.

Furthermore, in the software there should be an ability to ask permissions for data handling. In the original state, no permission is asked in the software but rather on paper for all the test subjects, which can be seen as a "privacy-by-policy" situation [5]. In children's case the contract is signed by their parents where as with the adults the contracts are signed by the research subjects themselves. The creation of a digital solution for permission handling can also prove difficult, as verification of age or parental consent is not easy to verify by digital means [13].

There should be different security levels and use cases for teachers, parents and pupils, as well as for employees and employers. There is also a third use case scenario concerning universities, where there is a mix of both of these use cases [18]. Even though university students are users comparable to pupils and the university is a school as an organization, students are still handled as adult research subjects. The universities also have other staff than teachers, which leads the universities being a combination of both schools and companies, in the sense of the research software [18].

Research question 3: How can we implement different types of privacy protection to the software to reflect the differences?

The assumption is that some level of adaptivity, higher level of user recognition [19] or different installation options are needed for the software. As the original software is a SaaS solution, there is the requirement of user recognition during runtime [11]. Another solution would be changing the software to be customer deployable solution (on-premise), where run time recognition is not so widely required, but the user types would be implemented during installation [11].

If the software is kept as SaaS, the protection level could be implemented as per client organization, as well as per user [19]. If the software is delivered as on-premise installation for customer, the privacy protection could be implemented for each installation separately, as there would only be one type of organization per installation.

3 Background

In this chapter the background for this thesis is presented. First the legislation perspective is handled, as it is the most driving force for the requirements of privacy. Alongside the GDPR, previous work regarding data privacy are inspected. Finally, the more concrete requirements are discussed, such as customer requirements and their impact on the software.

3.1 Legislation and previous work

Privacy is not just a matter of legislation, but also a human right. Thus it is not only under the General Data Protection Regulation (GDPR), but has been stated in the Article 8 of the Convention for the Protection of Human Rights and Fundamental Freedoms by the European Court of Human Rights [3, 6] and in the Article 12 of the Universal Declaration of Human Rights [2]. Therefore a breach in privacy is not a simple matter, and should be taken into consideration with the gravity it deserves.

> The protection of natural persons in relation to the processing of personal data is a fundamental right. Article 8(1) of the Charter of Fundamental Rights of the European Union (the 'Charter') and Article 16(1) of the Treaty on the Functioning of the European Union (TFEU) provide that everyone has the right to the protection of personal data concerning him or her. [14]

As the GDPR set into action in 2018, the question of privacy arose to surface not only in EU countries, but world wide. That is because the legislation concerns the rights of EU citizens, no matter where they are or where the data is being handled [14]. The sanctions for breaking the privacy regulation are severe [14], and thus also taken seriously by the software industry.

> Infringements of the [...] provisions shall, in accordance with paragraph 2, be subject to administrative fines up to 20 000 000 EUR, or in the case of

> an undertaking, up to 4 % of the total worldwide annual turnover of the preceding financial year, whichever is higher. [14]

One of the key points for GDPR is that if personal data is being processed, the data subject, or the person who owns the data, should be notified about the processing in advance [6, 14]. Even then, the processing is allowed only if the data subject gives their consent for the processing [6, 14, 17].

> 'consent' of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her; [14]

Notable aspect of personal data is that for data to be personal, the person has to be identified and singled from it [10, 14]. As the software originally did not gather any identifiable information, this was not an issue. As the need to store research subjects' names became relevant, so did the need to think about personal data.

> The principles of data protection should apply to any information concerning an identified or identifiable natural person. Personal data which have undergone pseudonymisation, which could be attributed to a natural person by the use of additional information should be considered to be information on an identifiable natural person [14]

In the GDPR, the concept of pseudonymous data handling is noted as a possible custom to handle personal data [14, 1]. With this the personal data can be used to form a collection of a data subject, but said data cannot be tied to the actual person [14]. This can be done by removing any identification from the collected data, which in the case of dLearn.Helsinki means the removal of the users' actual names from the actual research data.

> 'pseudonymisation' means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person[.] [14]

The GDPR is a new regulation as it came to force in 2018, and at the time of this thesis there were no juridical verdicts on the interpretations of the regulation. There are not many studies on the regulation, either, for the same timing reason. Therefore the literature for this subject was mostly from other countries of from before the regulation had become in force, where the effects are mostly speculations. The European Union had nominated a group called Article 29 Data Protection Working Party (29WP), whose function was to give preliminary interpretations for the regulation [6, 14]. However, the Opinions of the 29WP are not legally binding like the regulation itself, but as the name suggest, merely opinions or suggestions, and are not enforceable as such [6]. Alongside the effects of the privacy regulation, I focused on finding preceding work for changing a SaaS solution to an on-premise version, as well as the possible security risks on doing so [11]. These articles were used to support the privacy concentrated angle of the thesis, as well as to create the design science artefacts.

From the literature, a clean consensus arose: The privacy regulation affects the data processing of individuals, and the status of the children is more secured than that of the adults [6, 13, 14, 22].

> Children merit specific protection with regard to their personal data, as they may be less aware of the risks, consequences and safeguards concerned and their rights in relation to the processing of personal data. [14]

Furthermore, children are not as knowledgeable of their rights or the threats they are under [18, 22]. The legislation differs not only between children and adults, but also between different groups among adults [16]. Where children can be considered to have the most protected personal data, amongst adults the rights of an employee can be handled by their employer quite freely [16, 6, 14].

> Member State law or collective agreements, including 'works agreements', may provide for specific rules on the processing of employees' personal data in the employment context, in particular for the conditions under which personal data in the employment context may be processed[.] [14]

It could be argued the rest of the adults form a third group, whose rights set somewhere in between these two extremities [10, 6, 17]. They are of course also protected by the GDPR, being part of the grand majority, rather than exceptions.

3.2 Customer requirements for change

The necessity for this design science case arose from the dLearn.Helsinki research group. The testing or piloting of the software had begun and the users testing the service had requests and suggestions, which as such required more planning and research than only minor software changes would have. Some of the changes needed affected the user handling inside the software, and some dealt with the installation and service provision type.

The first main issue was the necessity of changes from the user base in the piloting phase. When teachers wanted to know who is who, or which user is which, and what the usernames represent, it became apparent that the prefix and numbers weren't an adequate username scenario. Furthermore children could not remember their codes as usernames and it was impractical as the youngest research subjects were hardly able to read and write and it was even harder for them to remember this arbitrary code for them [10].

With these changes the program was not anonymous anymore since we were now handling the usernames as persons' real names. For this we had to pseudonymise the data so that the real names of the research subjects were not available for the research team but only inside the teams themselves [7, 14]. In a research perspective it didn't matter if the data was pseudonymous or anonymous, since all the research team needed was to be able to recognize or single out one person's data from another.

Another change required was for the use cases themselves, which also brings us our first artefact. As children cannot give consent to their data procession but the permission has to be given by their parents it came apparent that we would have to implement this in the software in multiple different ways.

> Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child. [14]

At the time of this thesis there was actually no consent question at all in the software itself but consent was handled with paper forms. Even though this could have been handled digitally especially for the adult users, it was handled manually since the difference of consent was so critical on different user types.

Third major case change was the growing user base and the need for different kinds of organizations inside the program. In the beginning there were only 3 user levels: pupil, teacher and admin. As user levels these were equivalent for user, team leader and admin A. A need arose for additional level of users, such as an organization leader, situated between the team leader and admin on privileges B. With the three levels of users, the teams were handled almost like organizations of their own, and the admin became the organization leader, even though having to handle all the organizations in the system. The admin being actually the system admin, this was not a sustainable solution, but organizations needed to be able to handle their internal accesses by themselves D.

During the pilots, a frequent request was the possibility to have the software installed on-premise on customers' own servers, rather than being reliant on the cloud service. The request mainly came from schools, with a feel of storing the children's data felt safer on their own server rather than in a server over the internet.

4 User groups

In this section the different user groups are introduced. They are the base for use case scenarios, which will be presented later. Some of the user groups differ from one another vastly, and therefore differentiating them is crucial. As such, the focus is on groups which are exceptional or different from others in the form of privacy, when comparing within or with other organizations.

4.1 Companies

Companies are organizations which include team leaders, employees, and admins, in respect to the dLearn.Helsinki software. All the users in a company are adults, and hence the privacy settings in the software have to be considered with this in mind. Even though all are adults, they still do have different kinds of security settings, for example different access levels depending on their status in the organization.

4.1.1 Employees

Employees are adults rather than children and this is a major difference when comparing the data privacy settings of these two groups as data subjects. Employees can give their personal consent for their data handling and it could be asked directly in the software [16, 14, 6]. Also, employers may direct employees to answer, as there might be some restrictions or forcing events in their working contracts. Just like with pupils, pseudonym data is enough since there is still no need for personal data for research setting.

4.1.2 Team leaders

Team leaders are at similar status to teachers in schools, but have different permissions when it comes to granting data permission. As the research subjects are adults, team leaders cannot give consent on their behalf. It is notable, that team leaders are also employees, and can thus have a double role in the organization in respect to the

research. In this aspect they either require two accounts to the software, or their use cases in the software are more complex than are considered in this thesis.

4.2 Schools

The schools are organizations with teachers, pupils, rectors or for example school councilors. In school environment pupils are the research subject teachers can be seen as and team leaders and rectors could be system or organization admins.

4.2.1 Pupils

Pupils are most protected by the law since they are usually underage [1, 10, 14, 22]. In GDPR there are several recitals and parts of law that differentiate children from the consent of adults. Children cannot give consent to their data, nor can they possibly understand everything where their data is being handled [1, 9, 14, 22]. This is why in the GDPR there is a clause for their permission to be handled by their legal guardians, as well as in the software where their data should be handled the content should be asked in such a way that the children would have a possibility to understand why their data is handled [9, 14].

However, as in the case software there is no real need to know the personal data of children. The personal information of the children comes from the necessity of easier usernames. For the sake of the research, anonymous data would be preferable, so the handling of personal data is not as large an issue as it could be [1]. The only need for personal data is the usernames and that came from the requirements of the teachers and team leaders rather than from the necessity in the software. For research, the data is pseudonymized, and the personal data is kept within the school.

The software gathers the statistical data of a person, and if they have answered. From the research point of view, this data can be pseudonymic. This means that the research group does not need the names for the children, and the personal data is kept within the school, as in given to the teacher. As the original plan was with anonymous data, this does not compromise the research integrity, as the tracebility of an individual is the same, but the identifier given in the software is created differently.

4.2.2 Teachers

Unlike team leaders in companies, teachers have the right to change the data permissions of research subjects, the pupils. As pupils cannot give the consent on their own but they require parental consent, handling the consent via teacher helps in authenticating the source of consent. With this extra step the pupils cannot grant the consent themselves. Teachers, alike team leaders, could also be research subjects, and could require more detailed access levels to be able answer surveys in the software.

4.3 Universities

Universities have similarities between schools and companies as well. Even though at an organizational level they could be very similar to schools where there are students, teachers and rectors, the research subjects are still adults and should be handled like the employees in a company. In the university there are also other employees than teachers so there could be also another layer of user cases in the organization [14, 10].

4.3.1 Students

University students are adults and can give their personal content and the university may instruct the students to answer for example in return for credits or part of a university-wide survey [10]. Compared to children, they can give their own consent, even though in organizational level they are the same [10, 14].

4.3.2 Staff

University staff are adults and can give their personal consent and, as with other employees, the employer may direct them to answer the questions [10]. Like in other companies, the contract of the employer might require them to complete certain tasks, for example, filling in a form [6]. Thus the consent for privacy has to be taken into consideration as equal to staff members in any organization.

5 Design science artefacts

In this chapter I introduce the artefacts produced in the design science case. Each artefact is introduced alongside the six design science steps used to produce them [15].

5.1 Use cases

In this part I introduce the use cases and how they affect this design. In the beginning of the design science case, the program was aimed for small but growing demo target audience and the use cases were similar to that situation. As the demo cases grew, so did the need for larger organizations in the program, as well as the difference between different kinds of organizations, such as schools and companies became more apparent.

The software started as anonymous, but the requirement for user information inside the organization from the test users pushed to move towards storing personal information in the software. As team leaders and teachers wanted names to be held inside the software but the researchers did not want or need the information, some solution was required.

Use cases for new organization levels were needed to bring in the organization admins, as well as take burden off the system admin who is actually in the research group. Also use cases for different types of organizations were needed since the original use cases were only in one type which was restricted to be schools.

1. Problem identification and motivation: No use cases were documented when I started this project. Furthermore, new use cases have to be implemented and designed before they can be taken into use.
2. Define the objectives for a solution: The objective is to have clear mapping of use case scenarios. This includes both the old and the new use cases.
3. Design and development: The designing of the use cases was done by researching the software for use types and finding out what kind of use cases existed. During the development of the software new use cases were developed in accordance to the requirements from the customers and from the software development as such.

USE CASE	Pupil	Teacher	Admin
Create User		x	x
See User information	x	x	x
Edit User information		x	x
Delete User		x	x

Table 5.1: Original use cases for user CRUD, extracted from appendix A

4. Demonstration: Documented use case scenarios as well as the developed user type requirements. The demonstration has not been fulfilled in the fullest form of design science, as all the use cases are not in use, but for the purpose of this thesis they are adequate.
5. Evaluation: The evaluation step can be seen in accordance with step 6 for communication with the thesis provides the required documentation and further suggestions for software development. With these, we can develop the software and create the new use cases in accordance to the requirements.
6. Communication: As mentioned in previous step, the communication is done as part of this thesis.

5.1.1 Original use cases

The original use cases consisted of three user levels; pupils, teachers and admins. There were some lacking features in that state of the software but there was a necessity to find out what we are actually having and what we needed for the software to be more complete. This first design science artefact in appendix A shows the use cases of that situation when there are aforementioned three levels of users and only a very selective amount of use cases. These are not applicable for companies, as they are all set to the school environment. Furthermore, the teacher and the admin are very close to each other as use cases, even though the admin comes from the research group itself.

5.1.2 Developed use cases

When the program grew and the research went forward another level of users was added called the organization admin. The organization admin originally resembled the software admin very much, as all their use cases were identical to one another. Also the

USE CASE	User	Team Leader	Organization admin	Admin
Create User		x	x	x
See User data	x	x	x	x
Edit User data	x	x	x	x
Delete User		x	x	x
Create Survey		x	x	x
Update Survey		x	x	x
Delete Survey			x	x

Table 5.2: Developed use cases for user CRUD and surveys, extracted from appendix B

amount of use cases grew as there became abilities to create questionnaires on different levels than just the organization admin.

In this edition in appendix B we now have four levels of difference. We have added an organization admin between the team leader and an admin from the system. Furthermore, we have acknowledged that the user levels are not only constricted to the school, but have been changed to be user, team leader, organization admin, and system admin.

Now we can disambiguate the users, so that it can be used in both schools and organizations. At this stage, still, the organization admin and the system admin are very close to each other, and there is no real distinction between them. The admins still have equal privileges in the system. However this is not a viable solution, since there were now two levels of the users with exact same levels of permission, and there would actually be need to differentiate the organization admin from the system admin.

5.1.3 Suggested use cases

In the future the restrictions of system admin should be even more since there is no need for the sysadmin to be accessing the user or survey data rather than only be able to update it from the research side and also extract the data. Furthermore, there are new use cases such as granting and deleting the permission, meaning the data handling or the permission handling of a data would be implemented in the software itself rather than being manually on paper, as shown in appendix C.

At this point there's also a difference between schools and companies but no clear use case for University have been planned. it is most likely that the use case for University is that off the company but on the employee level there are both students and employees of the University.

USE CASE	Pupil	Teacher	School Secretary	Admin
Create User		x	x	x
See User data	x	x	x	x
Edit User data	x	x	x	x
Delete User		x	x	x

Table 5.3: Suggested use cases for school CRUD, extracted from appendix C

Grant permission		x		
Delete permission	x	x	x	

Table 5.4: Suggested use cases for school permissions, extracted from appendix C

In these suggested diagrams, we have separated the different organization types. Not only do we have the four levels of users, like we had in our previous chart, but we have now also separated the school from the company, as can be seen in appendixes C and D. The school has the pupil, teacher, and school secretary, as well as the school admin as seen in appendix C, whereas the companies have employees, team leaders, organization admins and system admins, as seen in appendix D. Furthermore, there have been developed more use cases, and they are especially relevant for the permission granting of the privacy settings, seen in appendixes C and D.

In the school environment, the pupil can only delete the permission from him or herself, or his teachers can grant and delete the permissions. A school secretary can delete the permission and the system admin cannot touch the privacy settings at all. This is due to the privacy of an under aged child in a school. Only the teacher can verify the consent from the parents, and thus verify the permission, and the students are themselves allowed to delete the permission as well. The teachers and school secretaries are also allowed to delete the permission in accordance if there is the pupil or the parents require so. In companies and universities the granting and deleting permission is left only for the employee, as they have the full control and ownership of their data. Team leaders, organization admins and system admins do not have to have the privilege to change the settings of an adult user.

From case study perspective, use case diagrams are clear artefacts. The development of the diagrams is noticeable, and the progression of different aspects can be seen clearly. Thus the evaluation can be done by examining the diagrams, as the development is tangible in different versions of the diagrams. The communication to the customer, in

USE CASE	Employee	Team Leader	Organization admin	Admin
Create User		x	x	x
See User data	x	x	x	x
Edit User data	x	x	x	x
Delete User		x	x	x

Table 5.5: Suggested use cases for company CRUD, extracted from appendix D

Grant permission	x			
Delete permission	x			

Table 5.6: Suggested use cases for company permissions, extracted from appendix D

this case the research group, can be done via documentation, such as this thesis.

5.2 Installation options

At the moment the software is produced as a service which means there's only one installation that is provided by the research company. This was challenged by some of the customers during the pilot phase, when they desired to have an on-premise version of the software

5.2.1 Software as a Service

Software as a Service (SaaS) is easier for the research group to maintain as the only installation is held by one group. This also means that no installation requirements are demanded from the customers but everyone can access the software, as long as they have a computer with an internet connection. The research data is easy to access for the research group as the data is being handled on their own server.

However, the SaaS also gives some restrictions. There are no custom logins for users such as different schools can not use Wilma login nor can companies use their own OAUTH login either [8]. There's no possibility to integrate the software to other systems nor can it be restricted to only certain networks.

In the future when the software is marketed the software can be billed by organization size and the billing would be most likely equal to all organizations.

1. Problem identification and motivation: At the time of this thesis, the software

is only provided as SaaS, and the customers require an on-premise variety. The problem is that the software is only in the cloud service whereas customers would want to have it on their own servers.

2. Define the objectives for a solution: The objective for solution is to have a have an on-premise version of the software available alongside the SaaS product, so that the customers could provide the software from their own server, and thus have the data and access managed by their own organization.

3. Design and development: To tackle the problem at hand, an installation script was created, as well as other design choices considering both the SaaS and the on-premise version. The designing started from having the other option available and then deciding which of the services would be most reasonable to implement in the long run.

4. Demonstration: We have an installation script that works, and also at the same time, we have the SaaS version available.

5. Evaluation: The evaluation of this problem is hard to implement, as the SaaS is this the only option, and comparison between SaaS and on-premise cannot be done at this point. However, the SaaS works as intended, and it has been developed into further production. Thus the evaluation could be done on the SaaS production, and leaving the comparison to a later stage.

6. Communication: The SaaS is the only version available for customers, at the moment, so the communication towards the customers can only be done through with that.

The SaaS solution has certain advantages compared to the on-premise version, which will be introduced next. The access to research data by the research group is one of the major advantages, as it does not only make research easier, but also restricts the amount of people who have access to the data. The SaaS being the status quo also requires less work and development, as the software is served as a single instance, and no updates for customer installations need to be provided. On the other hand, all the maintenance can be a burden, and take away from the main purpose of the software.

5.2.2 Installation tool

In the pilot phase there became apparent need for an on-premise version of the software. The aim for this is for users to be able to install the software on their own servers and give them an option of local installation. The maintenance of such an installation is on the customer's responsibility as the research group would have no access to their servers. Furthermore, if the research data is on the customer server, then the access to the research data would rely solely on the customer co-operation. This brings difficulties since the research integrity must be held together with contracts. This is tied to the previous chapter very closely, as now we are talking about the optional choice of the SaaS production.

1. Problem identification and motivation: As previously, the problem is that the on-premise version of the software was required by the customers, but also another problem arises with that. The data access for the researchers becomes more tedious, as there's no direct access to the server, but the access has to be asked for from the customer side.
2. Define the objectives for a solution: The objectives for this are the installation script, as mentioned earlier, to having an option of an on-premise installation. Also the other problems have been tackled, such as the data access, and this should be considered with contracts with the customers. This, on the other hand, is outside the scope of this thesis and will not be handled.
3. Design and development: The design and development of this problem has been quite straightforward. My task was to design and implement the script and the required tools for this script to be used. With these tools the customer could have installed their own on-premise version of the software, as long as they met all the requirements set by the installation script.
4. Demonstration: The demonstration for this part is that the script actually works, and we can use it to install an on-premise version. This has not been tested in customers' use as the on-premise version is not available yet, but it has been proven to work in a test environment.
5. Evaluation: As this has not been truly proven to work yet in a production environment, the evaluation and communication has only been done inside the research team and can not be validated or communicated with the customers for

now. On the test level, the solution works and can be managed as such for the customers.

6. Communication: The installation tool was trialed in 2019, but was never taken into production. It could still be considered as completed, even if it requires some finalization.

This would bring some customization for the customers such as implementing their own logins or they could be integrated with some other services.

A billing plan for such an implementation would be different than that of the SaaS since there would be no possibility to see the size of the organization using the software. Also as the software is installed on a customer server all the code is accessible by the customers themselves.

In the next example from appendix E, we can see the first part of the installation script. It begins by creating the configuration files for the script itself, and the configuration files are retrieved from the dLearn.Helsinki server, if required. Thescript creates environment variables and checks if they previously exist. It will also check the existence of Docker, as Docker is required for the software to be available. If all the requirements are met, the script will download the Docker images from the Docker vault created in the previous chapter.

To set the environment variables for customer use, the script asks two questions from the user. The user is also informed about the settings being changed. The first question is used to determine if the institution using the software is a school, and sets the variable `DLEARN_SETTING` accordingly. As the setting is used to determine if the organization has under-aged participants or not, only two options are necessary.

The second question is used for setting the default language of the software. In the SaaS solution, the default language is Finnish and the option for English is available after login. For the installable version, the default language could be set during the installation, and other options could be chosen by users.

With these kinds of settings in the installation tool, we have created the option for privacy and restricted the different kinds of privacy settings and need for privacy settings already in the installation stage. If a on-premise software is installed, we have the need only for either adult settings or the settings with that with the ability to have minors as users in the software.

Evaluating the success of the SaaS and on-premise artefacts is difficult. As the SaaS version of the software was already implemented before the case study, the role of the case was merely to document the existence of the SaaS, and provide the possible alternative. Evaluating the on-premise, on the other hand, could not be done in production environment, as it was not implemented during the case. Communicating the success of these steps for these artefacts is thus also left for future development.

```
28 # Create the configuration files for Docker
29 dlearn-create-config-files() {
30     echo "Checking if necessary files to run the system are present.
31 If not, files are created! \n"
32 
33     if [[ ! -f env_vars ]]; then
34         echo "Creating env_vars"
35         echo "SECRET_KEY=DELETED
36 POSTGRES_PASSWORD=DELETED" >> env_vars
37     else
38         echo "File env_vars already present!
39 Skipping..."
40     fi
41 
42     if [[ ! -f docker-compose.yml ]]; then
43         echo "Creating docker-compose.yml"
44         echo "version: '3'
45 
46 services:
```

```
...
```

```
89     else
90         echo "docker-compose already present!
91 Skipping..."
92     fi
93 }
94 # Download Docker images from the registry
95 dlearn-download-images() {
96     if [[ -z "${DLEARN_USER}" ]]; then
```

Figure 5.1: Code for first configuration, extracted from appendix E

```
152 # Set production environment, school or office, eng or fin
153 dlearn-set-environment() {
154     echo "
155 This is to set the environment.
156 Do you use this in school? [Y]es / no"
157     read answer
158         if [[ $answer = [yY][eE][sS] || $answer = [yY] || $answer = ""
                ]]; then
159             export DLEARN_SETTING="SCHOOL"
160             echo "Set DLEARN_SETTING as SCHOOL"
161         else
162             export DLEARN_SETTING="OFFICE"
163             echo "Set DLEARN_SETTING as OFFICE"
164         fi
165     echo "Done.
166
167 This is to set the language.
168 Which language do you prefer? [E]nglish / Finnish"
169     read answer
170         if [[ $answer = [eE][nN][gG][lL][iI][sS][hH] || $answer = [eE]
                || $answer = "" ]]; then
171             export DLEARN_LANG="ENG"
172             echo "Set DLEARN_LANG as ENG"
173         else
174             export DLEARN_LANG="FIN"
175             echo "Set DLEARN_LANG as FIN"
176         fi
177     echo "Done."
178 }
```

Figure 5.2: Code for variable creation, extracted from appendix E

6 Discussion

I found that children are very protected compared to adults. In the software, however, the data used was very minimal, consisting merely of the full name. Furthermore, this data was not required for the research purposes, but rather for the teachers to identify the students. This could be considered fair use of students' information for teaching purposes in teaching environment. Considering the findings, the research questions could be answered as follows.

Research question 1: How does the data privacy of children differ from that of adults, according to General Data Privacy Regulation (GDPR)? Privacy differs even between different groups of adults, but especially for children in comparison to adults. Children and their data are more protected, as assumed. The major difference is that children cannot give their consent for data handling, parental consent is required, even though children still own the right to their data.

Research question 2: How does the difference affect the use cases in the software under inspection? For the software at hand, the parental consent and its implementation caused the most changes or requirements for them, as the difference consent requires a completely separate use case for children and affects other user types as well. The most noticeable change is the confirmation of consent by teacher for the child, compared to an adult subject being able to consent for themselves. Determining who can authorize the consent or its removal in the software has great effect on the final product.

Research question 3: How can we implement different types of privacy protection to the software to reflect the differences? This question was approached from the viewpoint of use cases as well as installation tools. The difference could be implemented with either on-premise installation options or with different types of organizational and user access levels in a SaaS implementation.

Considering the six steps of design science, steps five and six of evaluation and communication deserve a closer look. For all the artefacts created during the design science, these steps were considered individually per artefact, while in fact the case should be examined as a whole. The evaluation of these artefacts prove difficult, as the time between the case and the thesis became elongated. Also the theoretical approach to

the design science case did not prove beneficial for the case, but a more hands on implementation would have been better.

The total evaluation criteria of the case would have benefited from a clearer definition to begin with. The current method of defining criteria for each artefact separately works, but the consistency would have been better if the design case steps were defined for the whole case, rather than different pieces. As such even though the individual artefacts reached their goal, the evaluation and communication of them was not as successful. This could have been improved by a more timely submission of the case study. As most of the artefacts were developed but not taken into production, the discussion step for their success comes down to possible future work.

In the recent light of Vastaamo.fi psychotherapy center's data breach [21], data security is a timely subject, and the handling of personal data should be handled with utmost care. In the case of dLearn.Helsinki the risk of leaking personal data is smaller compared to Vastaamo.fi, as the only personal data is the user's real names as login information. A risk nonetheless exists, and has to be considered when developing the software.

7 Conclusion

In the case of dLearn.Helsinki, the effect of data privacy is noticeable. The difference in the handling of under-aged subjects' data causes not only changes to the process, but also differences in the use cases and user levels within the software. A possibility of a different installation system was also considered as a solution. If an on-premise installation is delivered to the customer, the user levels and accesses are locked into certain levels, which removes the necessity of different types of organizations in the installed version of the software. With the SaaS there is a possibility of error in user type definitions, but no one outside the research group has to go through the installation. The access to research data is also restricted to the research group, making the security of the data more explicit.

The changes suggested in this thesis were not implemented at the time the case was studied. Since then, the software has evolved, and some of the suggestions have already been taken into consideration, such as the different organisation types in the software. For future work, update for the status of the suggestions would be in order. Furthermore, developing the different installation options was still in progress while the case was studied. If the options are still viable, development of the installation tool and an on-premise solution would continue this work.

The consent for the research should be implemented in the software itself. Especially for the parental consent, having the information within the same system as the research data is crucial.

The different installation options bring possibilities but also different kinds of problems. While both solutions have their pros and cons, I am inclined to favor SaaS, due to the simplicity and data security. With the SaaS solution, we can have the data access restricted only for the researchers, if the security settings are done properly. Furthermore, all the settings are restricted by the system admin, which is part of the research group, rather than having these admins in all the organizations as would be on the on-premise versions. This limits the amount of people who have access to the data.

Taking data privacy into consideration in a software is a difficult task on its own. As the amount of different user types grows, the possible differences in their privacy has

an increasing effect on the software under development. For dLearn.Helsinki, having under-aged research subjects and changing software from anonymous to pseudonymous created a situation where data privacy had to be considered from multiple aspects, rather than handing all data subjects equal. Independent of the future implementation of the software in the future, the data should be handled safely, no matter the age of the subjects.

Bibliography

[1] S.-D. Axinte, G. Petrică, and I. Bacivarov. "GDPR Impact on Company Management and Processed Data: Acces la Success Acces la Success". In: *Calitatea* 19.165 (Aug. 2018), pp. 150–153.

[2] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Métayer, R. Tirtea, and S. Schiffner. "Privacy and Data Protection by Design - from policy to engineering". In: *CoRR* abs/1501.03726 (2015).

[3] C. of Europe. "Convention for the Protection of Human Rights and Fundamental Freedoms as amended by Protocols No. 11 and No. 14 *". In: (1950).

[4] A. Hevner, A. R, S. March, S. T, Park, J. Park, Ram, and Sudha. "Design Science in Information Systems Research". In: *Management Information Systems Quarterly* 28 (Mar. 2004), pp. 75–.

[5] T. Hoel, D. Griffiths, and W. Chen. "The Influence of Data Protection and Privacy Frameworks on the Design of Learning Analytics Systems". In: *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*. LAK '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 243–252.

[6] E. Keane. "The GDPR and Employee's Privacy: Much Ado but Nothing New". In: *King's Law Journal* 29.3 (2018), pp. 354–363.

[7] A. Kobsa. *A component architecture for dynamically managing privacy constraints in personalized web-based systems*. Vol. 2760. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2003, pp. 177–188.

[8] P. C. Kumar, M. Chetty, T. L. Clegg, and J. Vitak. "Privacy and Security Considerations For Digital Technology Use in Elementary Schools". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–13.

[9] E. Lievens and V. Verdoodt. "Looking for needles in a haystack: Key issues affecting children's rights in the General Data Protection Regulation". In: *Computer Law & Security Review* 34.2 (2018), pp. 269–278.

[10] M. G. Marković, S. Debeljak, and N. Kadoić. “Preparing students for the era of the General Data Protection Regulation (GDPR)”. In: *TEM Journal* 8.1 (2019), pp. 150–156.

[11] R. McTague and P. Fagan. “Architectural Refactoring as a Strategic Tool in the Evolution of a Web-Based SaaS Product”. In: *Proceedings of the 2nd International Workshop on Refactoring*. IWoR 2018. Montpellier, France: Association for Computing Machinery, 2018, pp. 11–13.

[12] OECD. *World Economic Forum*. Deloitte, 2017.

[13] S. Papadimitriou, E. Mougiakou, and M. Virvou. “Smart educational games and Consent under the scope of General Data Protection Regulation”. In: *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 2019, pp. 1–8.

[14] E. Parliament. “REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)”. In: *Official Journal of the European Union* (2016).

[15] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. “A Design Science Research Methodology for Information Systems Research”. In: *Journal of Management Information Systems* 24.3 (2007), pp. 45–77.

[16] S. Polst, P. Kelbert, and D. Feth. *Company Privacy Dashboards: Employee Needs and Requirements*. Vol. 11594 LNCS. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2019, pp. 429–440.

[17] D. F. Povse. “It’s All Fun and Games, and Some Legalese: Data Protection Implications for Increasing Cyber-Skills of Employees through Games”. In: *Proceedings of the Central European Cybersecurity Conference 2018*. CECC 2018. Ljubljana, Slovenia: Association for Computing Machinery, 2018.

[18] L. Sion, P. Dewitte, D. Van Landuyt, K. Wuyts, I. Emanuilov, P. Valcke, and W. Joosen. “An architectural view for data protection by design”. In: *Proceedings - 2019 IEEE International Conference on Software Architecture, ICSA 2019*. 2019, pp. 11–20.

[19] B. Spasic, A. Rath, P. Thiran, and N. Boucart. “Security Pattern for Cloud SaaS: From system and data security to privacy”. In: 2018.

[20] A. Tsirtsis, N. Tsapatsoulis, M. Stamatelatos, K. Papadamou, and M. Sirivianos. “Cyber security risks for minors: A taxonomy and a software architecture”. In: *Proceedings - 11th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2016*. 2016, pp. 93–99.

[21] YLE. *Psychotherapy centre’s database hacked, patient info held ransom*. 2020. URL: https://yle.fi/uutiset/osasto/news/psychotherapy_centres_database_hacked_patient_info_held_ransom/11605460.

[22] J. Zhao, G. Wang, C. Dally, P. Slovak, J. Edbrooke-Childs, M. Van Kleek, and N. Shadbolt. “‘I Make up a Silly Name’: Understanding Children’s Perception of Privacy Risks Online”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–13.

Appendix A Original Use Cases

USE CASE	Pupil	Teacher	Admin
Create User		x	x
See User information	x	x	x
Edit User information		x	x
Delete User		x	x

Create Survey		x	x
Update Survey		x	x
Delete Survey			x

Answer Survey	x		
Test Survey		x	x

Answer Question	x		
Create Question		x	x
Update Question		x	x
Delete Question		x	x

See individual results		x	x
See group results	x	x	x
See personal results	x	x	x

Create Group		x	x
Update Group		x	x
Delete Group		x	x

Create Subgroup		x	x
Update Subgroup		x	x
Delete Subgroup		x	x

Appendix B Developed Use Cases

USE CASE	User	Team Leader	Organization admin	Admin
Create User		x	x	x
See User data	x	x	x	x
Edit User data	x	x	x	x
Delete User		x	x	x

Create Survey		x	x	x
Update Survey		x	x	x
Delete Survey			x	x

Answer Survey	x	x		
Test Survey			x	x

Answer Question	x			
Create Question		x	x	x
Update Question		x	x	x
Delete Question		x	x	x

See individual results		x	x	x
See group results	x	x	x	x
See personal results	x	x	x	x
See organization answers		x	x	x
See all answers		x	x	x

Create Group		x	x	x
Update Group		x	x	x
Delete Group		x	x	x

Create Subgroup		x	x	x
Update Subgroup		x	x	x
Delete Subgroup		x	x	x

Appendix C Suggested Use Cases for Schools

USE CASE	Pupil	Teacher	School Secretary	Admin
Create User		x	x	x
See User data	x	x	x	x
Edit User data	x	x	x	x
Delete User		x	x	x

Create Survey		x	x	
Update Survey		x	x	
Delete Survey			x	

Answer Survey	x			
Test Survey		x	x	

Answer Question	x			
Create Question		x	x	
Update Question		x	x	
Delete Question		x	x	

See individual results		x	x	
See group results	x	x	x	
See personal results	x	x	x	
See class answers		x	x	
See all answers			x	

Create Group		x	x	
Update Group		x	x	
Delete Group		x	x	

Create Subgroup		x	x	
Update Subgroup		x	x	
Delete Subgroup		x	x	

Grant permission		x		
Delete permission	x	x	x	

Appendix D Suggested Use Cases for Companies and Universities

USE CASE	Employee	Team Leader	Organization admin	Admin
Create User		x	x	x
See User data	x	x	x	x
Edit User data	x	x	x	x
Delete User		x	x	x
Create Survey		x	x	
Update Survey		x	x	
Delete Survey			x	
Answer Survey	x			
Test Survey		x	x	
Answer Question	x			
Create Question		x	x	
Update Question		x	x	
Delete Question		x	x	
See individual results		x	x	
See group results	x	x	x	
See personal results	x	x	x	
See organization answers		x	x	
See all answers		x	x	
Create Group		x	x	
Update Group		x	x	
Delete Group		x	x	
Create Subgroup		x	x	
Update Subgroup		x	x	
Delete Subgroup		x	x	
Grant permission	x			
Delete permission	x			

Appendix E Installation Tool Code

Word ”DELETED” replaces some parts of the code for security.

```
#! /usr/bin/env sh

# Basic usage information, printed at the very beginning
usage() {
    echo "---
This is the installer for dLearn.Helsinki project.
This software is developed by University of Helsinki.

To use this software you need environment variables DLEARN_USER and DLEARN_PASSWORD set.
If you do not have these credentials, please contact the dLearn.Helsinki support.

---
Following methods are supported:

dlearn-help: Prints this message.
dlearn-download-images: Downloads docker images from container registry.
dlearn-run-images: Download images and run 'docker-compose up' to run the application.
dlearn-input-test-data: Format the database and add default data.

There can be other functions as well, but are only supported in development environment.
---
For more information and support, contact dLearn.Helsinki support."
}
# Print usage for help
dlearn-help() {
    usage
}
# Create the configuration files for Docker
dlearn-create-config-files() {
    echo "Checking if necessary files to run the system are present. If not, files are created! \n"

    if [[ ! -f env_vars ]]; then
```

```
34          echo "Creating env_vars"
35          echo "SECRET_KEY=DELETED
36 POSTGRES_PASSWORD=DELETED" >> env_vars
37      else
38          echo "File env_vars already present!
39 Skipping..."
40      fi
41
42      if [[ ! -f docker-compose.yml ]]; then
43          echo "Creating docker-compose.yml"
44          echo "version: '3'
45
46 services:
47   db:
48     container_name: db
49     restart: always
50     image: registry.gitlab.com/DELETED
51     ports:
52       - "5432:5432"
53     volumes:
54     #  - $PWD/logs/postgres:/var/lib/postgresql/data/logs
55       - $PWD/data:/var/lib/postgresql/data/
56     env_file:
57       env_vars
58
59   dlearnpy:
60     container_name: dlearnpy
61     restart: always
62     image: registry.gitlab.com/DELETED
63     ports:
64       - "4000:4000"
65     depends_on:
66       - db
67     links:
68       - db
69     volumes:
70       - $PWD/logs/app:/git/logs
71     env_file:
72       env_vars
73     command: make prod-serve
74
75   nginx:
76     container_name: nginx
77     restart: always
```

```
78      image: registry.gitlab.com/heikkihei/dlearnpy:nginx
79      ports:
80        - "80:80"
81      volumes:
82        - $PWD/logs/nginx:/var/log/nginx
83      links:
84        - dlearnpy
85      depends_on:
86        - dlearnpy
87  " > docker-compose.yml
88  echo "Done."
89      else
90          echo "docker-compose already present!
91  Skipping..."
92      fi
93  }
94  # Download Docker images from the registry
95  dlearn-download-images() {
96      if [[ -z "${DLEARN_USER}" ]]; then
97          echo "You need to set DLEARN_USER"
98      elif [[ -z "${DLEARN_PASSWORD}" ]]; then
99          echo "You need to set DLEARN_PASSWORD"
100     elif ! [ -x "$(command -v docker)" ]; then
101         echo 'ERROR: docker is not installed.'
102         echo ''
103         usage
104     else
105         echo 'Logging into registry.gitlab.com'
106         docker login registry.gitlab.com --username=$DLEARN_USER --
              password=$DLEARN_PASSWORD
107         echo 'Pulling dlearnpy'
108         docker pull registry.gitlab.com/DELETED
109         echo 'Pulling postgresql'
110         docker pull registry.gitlab.com/DELETED
111         echo 'Pulling nginx'
112         docker pull registry.gitlab.com/DELETED
113     fi
114 }
115 # Run the Docker images, i.e. run the software
116 dlearn-run-images() {
117     if [[ -z "${DLEARN_USER}" ]]; then
118         echo "ERROR: You need to set DLEARN_USER"
119         usage
120     elif [[ -z "${DLEARN_PASSWORD}" ]]; then
```

```
121         echo "ERROR: You need to set DLEARN_PASSWORD"
122         usage
123     elif ! [ -x "$(command -v docker-compose)" ]; then
124         echo 'ERROR: docker-compose is not installed.'
125         echo ''
126         usage
127     else
128         echo 'Logging into registry.gitlab.com'
129         docker login registry.gitlab.com --username=$DLEARN_USER --
              password=$DLEARN_PASSWORD
130         echo 'Running three containers'
131         docker-compose up
132     fi
133 }
134 # Input test data to the database
135 dlearn-input-test-data() {
136     if ! [ -x "$(command -v docker)" ]; then
137         echo 'ERROR: docker is not installed.'
138         echo ''
139         usage
140     elif ! [[ $(docker inspect -f '{{.State.Running}}' "db") = "true"
          ]]; then
141         echo "ERROR: container 'db' not running."
142     else
143         echo "This will reset the database to default data. Are you
              sure? [Y]es / no"
144         read answer
145         if [[ $answer = [yY][eE][sS] || $answer = [yY] || $answer = ""
              ]]; then
146             docker exec -it $(docker ps -aqf "name=db") psql -U
                  postgres -c "DROP DATABASE IF EXISTS dlearn" -c "
                  CREATE DATABASE dlearn" -c "\c dlearn" -c "\i
                  default_data/default_data.sql"
147         else
148             echo "Not changing the database"
149         fi
150     fi
151 }
152 # Set production environment, school or office, eng or fin
153 dlearn-set-environment() {
154     echo "
155 This is to set the environment.
156 Do you use this in school? [Y]es / no"
157     read answer
```

```
158         if [[ $answer = [yY][eE][sS] || $answer = [yY] || $answer = ""
                ]]; then
159             export DLEARN_SETTING="SCHOOL"
160             echo "Set DLEARN_SETTING as SCHOOL"
161         else
162             export DLEARN_SETTING="OFFICE"
163             echo "Set DLEARN_SETTING as OFFICE"
164         fi
165     echo "Done.
166
167 This is to set the language.
168 Which language do you prefer? [E]nglish / Finnish"
169     read answer
170         if [[ $answer = [eE][nN][gG][lL][iI][sS][hH] || $answer = [eE]
                || $answer = "" ]]; then
171             export DLEARN_LANG="ENG"
172             echo "Set DLEARN_LANG as ENG"
173         else
174             export DLEARN_LANG="FIN"
175             echo "Set DLEARN_LANG as FIN"
176         fi
177     echo "Done."
178 }
179 # FOLLOWING FUNCTIONS ARE FOR DEVELOPMENT USE ONLY!
180 dlearn-remove-docker-containers() {
181     echo "This function is for development use only.
182 This will remove ALL Docker images and containers on the machine.
183 Are you REALLY SURE? [Y]es / no"
184     read answer
185         if [[ $answer = [yY][eE][sS] || $answer = [yY] || $answer = ""
                ]]; then
186             docker rm --force $(docker ps -a -q) && docker rmi --force
                    $(docker images -q)
187         else
188             echo "Not removing anything"
189         fi
190 }
191 # Set development variables
192 dlearn-set-variables() {
193     if [[ -z "${DLEARN_USER}" ]]; then
194         echo "Setting DLEARN_USER..."
195         export DLEARN_USER="DELETED"
196         if  ! grep  -q "$DLEARN_USER" env_vars; then
197             echo "Saving DLEARN_USER into env_vars"
```

```
198             echo "DLEARN_USER=DELETED" >> env_vars
199         fi
200         echo "Success"
201     else
202         echo "DLEARN_USER already set!"
203     fi
204     if [[ -z "${DLEARN_PASSWORD}" ]]; then
205         echo "Setting DLEARN_PASSWORD..."
206         export DLEARN_PASSWORD="DELETED"
207         if  ! grep  -q "$DLEARN_PASSWORD" env_vars; then
208             echo "saving DLEARN_PASSWORD into env_vars"
209             echo "DLEARN_PASSWORD=DELETED" >> env_vars
210         fi
211         echo "Success"
212     else
213         echo "DLEARN_PASSWORD already set!"
214     fi
215     echo "Done."
216 }
217 # Create a demo environment by adding necessary variables
218 dlearn-demo() {
219     echo ""
220     dlearn-set-environment
221     echo ""
222     dlearn-create-config-files
223     echo ""
224     dlearn-set-variables
225 }
226 # Remove all variables and env_vars -file
227 dlearn-unset-vars() {
228     unset DLEARN_SETTING
229     echo "DLEARN_SETTING unset"
230     unset DLEARN_PASSWORD
231     echo "DLEARN_PASSWORD unset"
232     unset DLEARN_USER
233     echo "DLEARN_USER unset"
234     unset DLEARN_LANG
235     echo "DLEARN_LANG unset"
236     rm env_vars
237     echo "File env_vars removed"
238 }
239 # Run usage when the script is sourced
240 echo ""
241 usage
```