

Information systems project report

Group 13:

Bae, Magnus

Bremnes, Jan Alexander Stormark

Krane, Magnus

Sande, Kristian Olafsen

Tørresen, Håvard

May 3, 2013

Contents

Introduction	4
Background	5
Case description	5
Old system	5
Complexity	6
Our proposed solution	7
What is wrong?	7
What do we do?	7
How we are going to do it	9
Central Datastore	9
Storage model	9
User interface	10
Business processes	12
Registering for a subject or course	12
Complaining on a grade	13
Security	18
Login	18
Authentication methods	18
Access control	18
Discussion	19
Conclusion	20

List of Figures

1	Datastore and APIs	9
2	Storage model	10
3	Layered architecture	11
4	Model of the current process of registering for a course.	14
5	How we envision our new process of registering for a course should be.	15
6	A simplified model of the current process of complaining on a grade.	16

7	How we envision the new process of registering a complaint should be.	17
---	--	----

Introduction

This report documents parts of “NTNUs Ultimate Digital Learning platform” (NUDL); a system we designed as part of an exercise in the subject TDT4175. The report covers a subset of features imagined, and the involved business cases. NUDL is a proposal for an IT-system for the Norwegian University of Science and Technology (NTNU). NUDL proposes to digitalize some of the current processes involved in studies at NTNU, such as delivering a complaint on a grade, to ease the process for all parties involved and cut an unnecessary load of paperwork. Beyond that it also proposes to streamline these processes, along with the processes that are already digitalized, under a standardized user interface and an updated security system.

The report will begin with an analysis of the current situation, highlighting the processes and problems we found - most in need of rework and digitalizing. Following, there will be an in-depth explanation of how we propose to solve these issues, and how NUDL will work and respond when put to use in situations that are managed by other systems today.

Background

In this chapter we will present the background for this project, and review the factors that have influenced our decisions and vision for this project.

Case description

NTNU wants to digitize its teaching processes as much as possible.

This includes the whole chain from preparation by the teacher to making the curriculum available, lecturing, giving exercises and grading exams/exercises/projects and students complaining about their grades.

Old system

Today's system is divided into several semi-independent systems. This is for instance StudentWeb, Innsida 2.0, It's Learning, NTNU's webpages, and so forth. StudentWeb is part of bigger system called "Felles Studentsystem" (FS) which NTNU is required to use. This system includes back-office clients and other systems, and changing this would require years of lobbying and participating in slow working committees to come up with changes. By the time that new system were to be finalized it would probably be plagued with the same problems we see today.

There's also another system which is available for students at the IME-faculty at NTNU, this is called "EksamensWeb" and allows students to electronically register complaints to their exams. None of these systems really talk to each other. Data from FS is pushed into It's Learning through batch jobs, and the end result is a set of several systems, which on their own are very capable, but that don't really work well together. This lack of interconnection means that a student has to go through several systems to do something simple. If one would like to register into a new course, one would have to deal with three different systems (Figure 4). If one would like to complain on a grade given on an exam, one must also interact with three systems (at best, most students have to send in a written complaint), some of which demands manual labour from the faculty staff (Figure 6).

Complexity

Since the total system is divided into several sub-system, these systems are unnecessary complex and slow. Some of these systems are also:

- Proprietary closed source.
- Based on a 15 year old code base.
- Architecturally locked in to Microsoft-technologies.
- Expensive to license.

In our view this means that these systems are very hard to build further upon. It also means that if we were to upgrade only parts of the current system, it will probably end up just as complex as the current system since we would have to consider connections to the parts that are left.

Our proposed solution

In this chapter we will discuss our proposed solution to the assignment and how we would like to solve the task.

What is wrong?

Having initially looked at today's work processes we see lots of areas where things can be improved. Due to the limitations of this assignment not everything is being pursued, but the idea behind our proposed solution is that it should be able to address all of the problems with the systems that are in use today.

Our biggest complaint about the current systems is that it's very fragmented and requires lots of labour-hours. There are, in our experience, discrepancies between what is stated on a webpage describing a subject, and what is the reality. Several of the group members have experienced being told "we'll update it so it's correct for next year". Staff have to keep their subjects updated in several places, there's course descriptions on NTNU webpages, there's It's Learning, and a lot of courses also have their own separate webpage for those occasions where It's Learning doesn't cut it, or where they want to make information freely available.

Another problem seems to be the inherent problems with the technologies used. It's Learning tends to not learn a lot, but bugs both the users and itself with un-intuitive behaviour, odd bugs and browser incompatibility. For the last months logging into It's Learning has been messed up because it ends up just redirecting itself back to Innsida where one has to click once more (one group member even gets routed through It's Learning for HiST). It's Learning is for the most part a standalone system in a continuously more interconnected web full of "open" standards. The way we see it, It's Learning is the rod that has been pushed between the spokes of the wheel. There's a lot wrong with the system, but it's not all bad. It's just not right. In 2013 It's Learning is sadly holding the rest back.

What do we do?

Based on our and others analysis¹ [3] we conclude that it's time to replace It's Learning. In the long term our plan is not only to replace It's Learning,

¹It's worth noting that Rambøll's conclusions were that It's Learning can be streamlined, however we think that it's too difficult to turn this lemon into lemonade

but to create a Content Management System (CMS) and Learning Management System (LMS) that envelopes the whole of NTNU. It does sound very ambitious, but the plan is not to create a huge blob of software, rather we propose a modularized tiered platform where data is stored in a centralized storage. The idea is to create a secure, robust, and manageable platform which is testable, modularized, and compartmentalized. We also want to implement one thing that we really feel is missing today - search.

Do you remember what was before Google? Yes, obviously you do, but do you remember how it really felt before Google? Maybe not so much. Search is good, but good search is better. According to Google Zeitgeist (search statistics) [2] people are too lazy to type in facebook.com, they just search for “facebook”. Granted, the average NTNU student isn’t your average person, but why should students have to manually navigate complex navigation trees when they can type a couple of words and get the result they want immediately? We think that by designing our system architecture and data storage with search in mind, we can bring value and help both students and staff to save time.

Today students have to interact with studentweb and It’s Learning. Data gets passed one way only. We want our system to bridge the gap between the two by using a service-facade-like pattern² to link our system and FS/StudentWeb. This way, if one changes it’s easy to adjust, and there is room for making our system better while the communication with FS is still bound by the same contract. We want to make our system interact with StudentWeb on part of the student. That means that the students interact with a system that is consistent. As a security measure our system will log error messages and calls, and in rare cases where propagation does not work as expected, alerts can be raised and personnel with the right clearances can take action and manually merge the discrepancy.

²The “Service-Facade” is an implementation of a contract as defined in an abstract interface and it’s sole purpose is to mediate information between systems and formats. [1]

How we are going to do it

This section will give a brief description of how we plan on implementing our solution.

Central Datastore

We propose creating a central datastore containing all public information about NTNU. We will provide APIs that will give easy access to all the data, and the APIs will be open to the public. NUDL will use this datastore, illustrated in Figure 1, to retrieve all necessary information about courses, rooms and timetables, and so it will need to be constantly updated in order to contain the most recent information. This means that third party developers will have equal access to non-restricted data, which will enable students and staff to write their own tools which can easily be integrated with NUDL. We aim to make NUDL a platform to which tools can be added and removed as needed, not a monolithic system where someone else is in control. The calendar generator at www.ntnu.1024.no is a great example of an extremely useful application created by a student. We know that IME offers some APIs today, and these could be extended and integrated into our system.

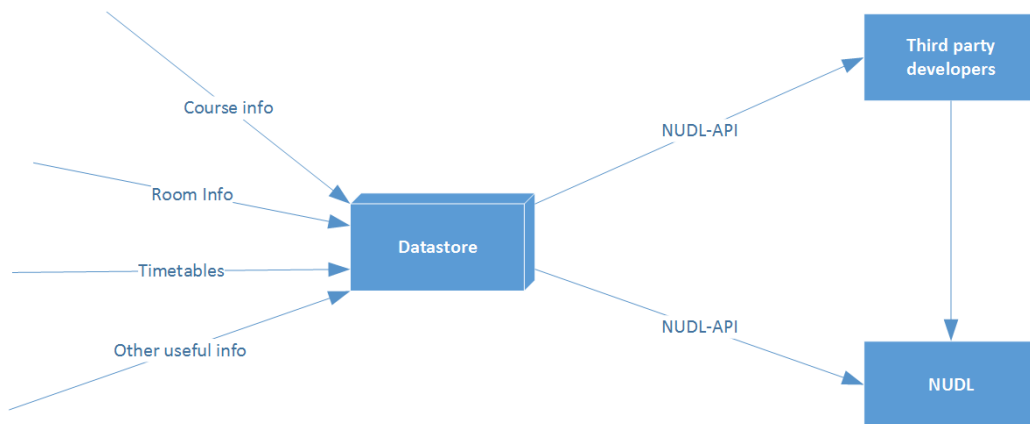


Figure 1: Datastore and APIs

Storage model

The datastore mentioned in the previous section will only contain publicly available information. In addition, we will have two other separate databases; one userstore containing information about staff and students,

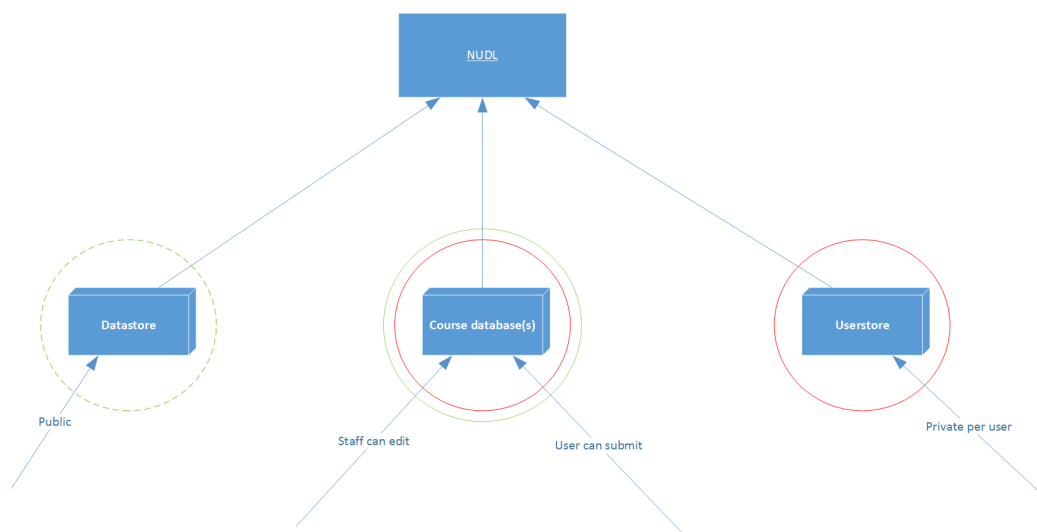


Figure 2: Storage model

and one database storing everything related to the individual courses. In Figure 2 the general datastore is shown on the left, course database in the middle and userstore on the right. The dotted green circle means that everyone is given read access, a full red circle means read and write access is restricted to a strict subset of users, and a full green circle means that there are two different levels of security. The userstore will be fully secured against unauthorized access, and only the users will have access to their individual information. The course database will be used to supply the same functionality as It's Learning does today. Students will for the most part have read-only access to the course database, but they will be given the possibility to deliver exercises and participate in forums and chat. Course staff will have read and write access to the course content pages, but will only be able to read the student submissions. Administrative staff will have accesses as needed. Course staff will be able to make course material available to non-students, by making it accessible through the public APIs.

User interface

We want NUDL to present users with a user-friendly and unified interface. It will not be like today, where you have different interfaces for It's Learning, Studweb and Eksamensweb. It's Learning and Eksamensweb will be replaced by NUDL, and Studweb will be hidden from view. Students can log in to Studweb from NUDL, via Feide, but they will be met by the NUDL interface, which will take care of the communication with Studweb, as shown in Figure

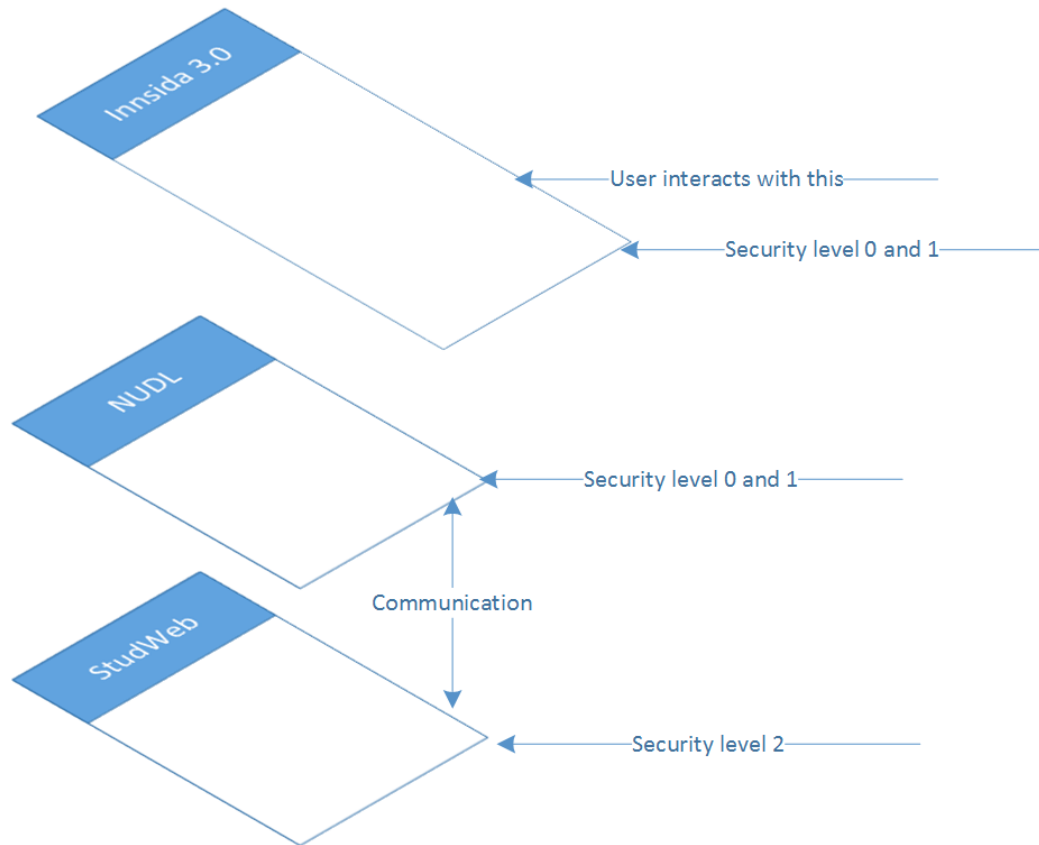


Figure 3: Layered architecture

3. Staff and students can do everything they need to do, from one central location. There will be no more need to log in to different services like today, where one first has to search for courses on ntnu.no, (UI no.1) log on to Studweb in order to register for the course (navigating a very cluttered and confusing UI no.2) and then log on to Innsida (UI no.3) and then finally continue to It's Learning (UI no.4). If they want to complain on a grade, they need to send in a paper form, or students at IDI can log on to Eksamensweb (UI no.5). NUDL combines all these different interfaces into one coherent whole.

Business processes

In this section we'll provide BPMN-models [4] of today's work processes, and show how these work processes will benefit from our system. There might be inaccuracies in the models, but to our knowledge there isn't any major flaws. Due to the fact that most of the users of today's system are students this is the perspective we've focused on in this project. It's worth nothing that all staff should benefit from the new system.

Registering for a subject or course

As shown in Figure 4, the current process of finding, selecting and registering for courses is very cumbersome for the students. The information about each course is spread around on different sites, and due to It's Learning being used to handle most of the information released during a semester, there is no simple way for students to see what a course actually contains.

Figure 5 shows our suggested changes to this process. With all information gathered at a single location, we can offer students a coherent experience and improved functionality. The students can follow the old approach, and manually look through courses, log in at the end, and register, or they can at any earlier point choose to log in and utilize the course suggestion feature to simplify selection. It's worth nothing that even though the work-flow can seem a bit similar in certain cases, the intention is that this information should be more accessible and more correct than today.

Course suggestion works by the system comparing the available subjects to the ones you've already taken, if there are subjects in which your results were good the system will suggest subjects that are close to, or in the same field as those subjects. It will also filter out any courses for which you might be missing prerequisites.

One of the complaints against It's Learning is that all courses are locked so that only students following the course can get access to the information. This has led to a fragmentation between courses using It's Learning to provide the students with information and resources, and the courses which provide their own website. Our system will allow staff to choose whether or not a subject should be publicly available or not. By using discretionary access controls, staff can also create certain pages, folders or files that aren't accessible for people not following (or not logged in). This granular access control would provide opportunity to streamline and provide a uniform experience across subjects and disciplines. The BPMN-model for choosing course does not reflect these abilities since it does not affect the process, only how

much information is available to the students.

Complaining on a grade

The current complaint-process, illustrated in Figure 6, is split across several systems, and forces most students to file complaints in writing, which are later digitized, slowing the process down unnecessarily. Our proposal is to completely digitize this process by allowing the complaint to be delivered electronically, as well as unifying the different sites included in the process as shown in Figure 7.

Behind the curtains, NUDL would be registering dates and times, making sure that everything happens at the right time (or that a student isn't allowed to complain after the deadline). One possible feature could be to collect anonymized data for statistical purposes and use this to look for areas of improvement.

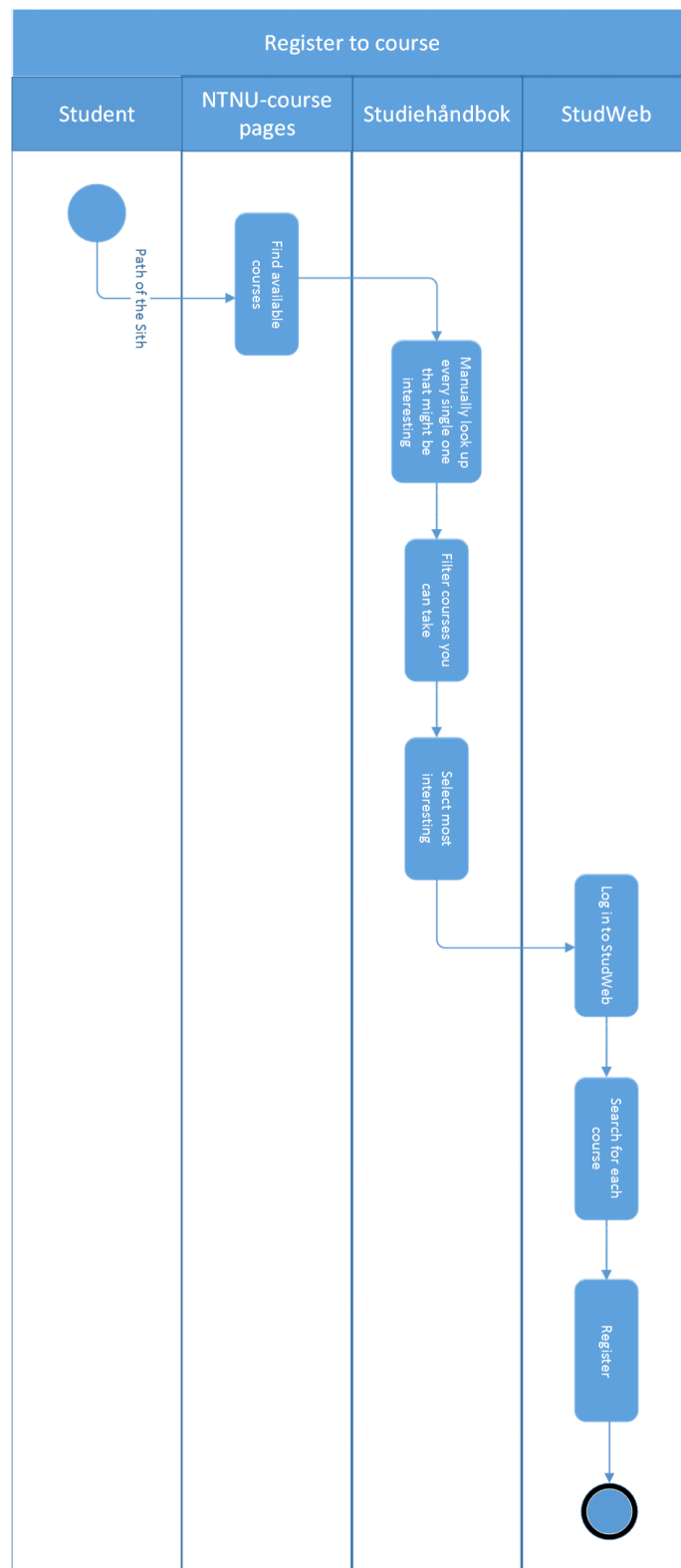


Figure 4: Model of the current process of registering for a course.

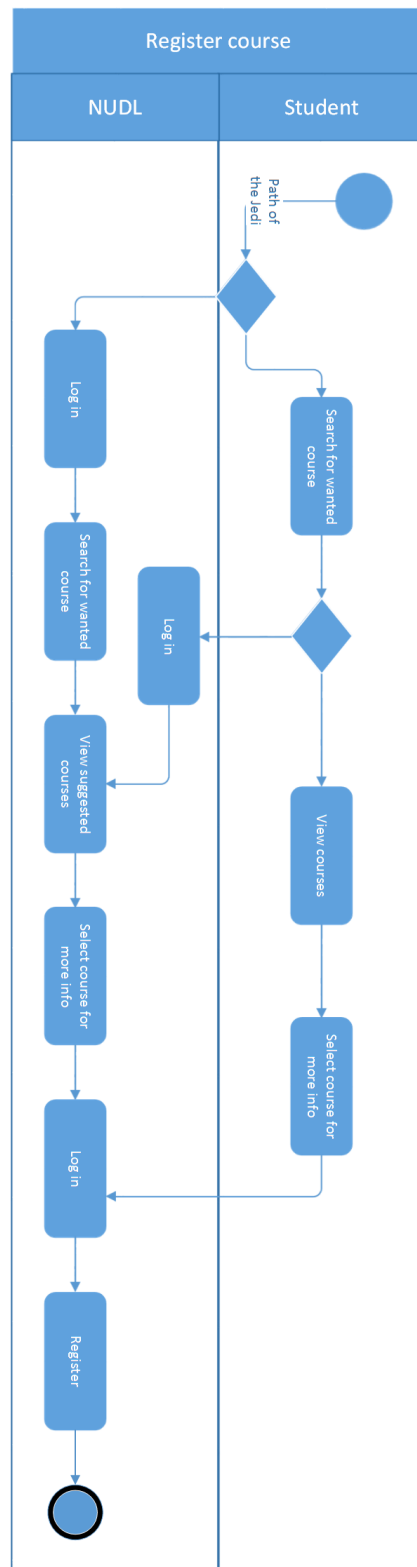


Figure 5: How we envision our new process of registering for a course should be.

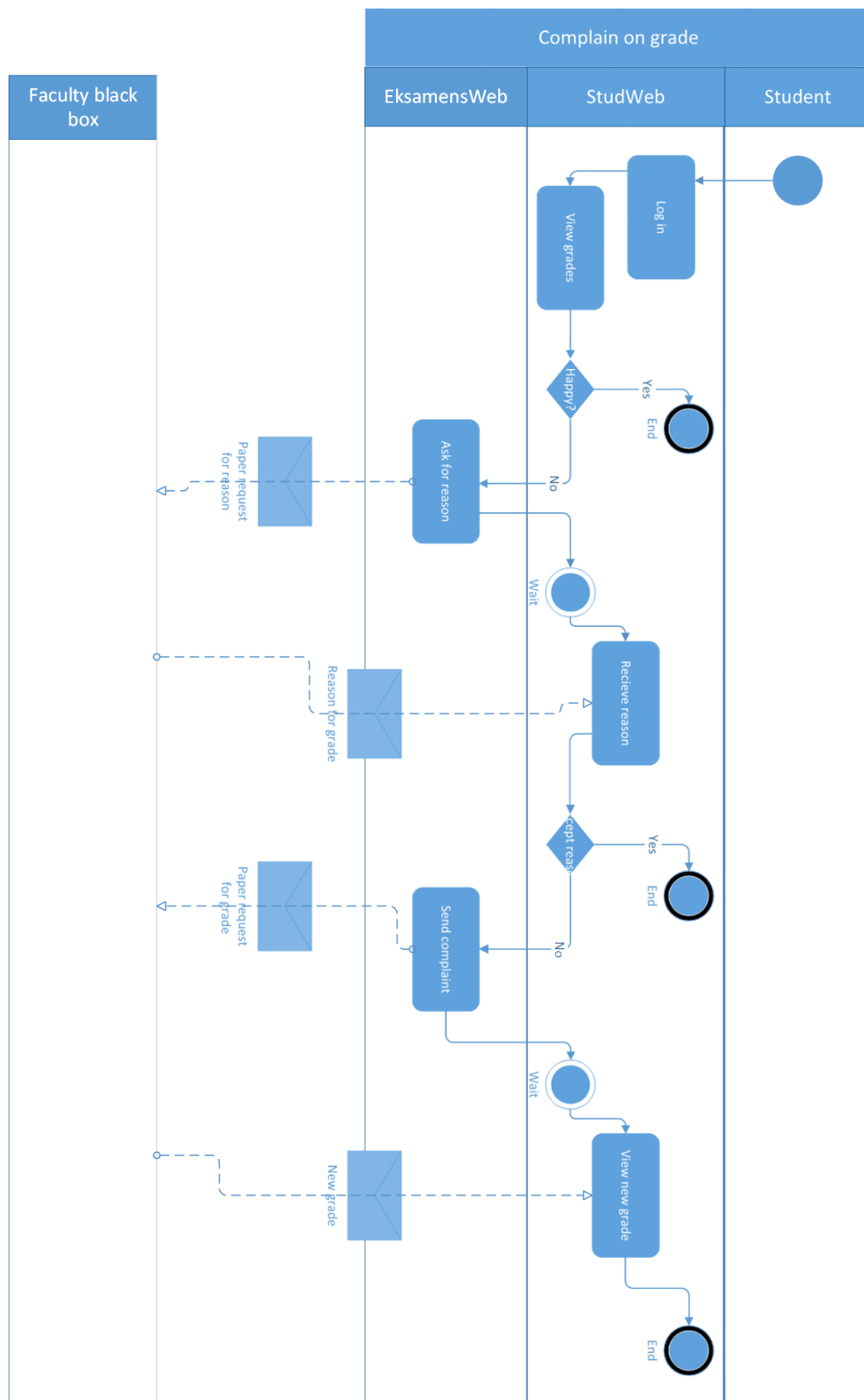


Figure 6: A simplified model of the current process of complaining on a grade.

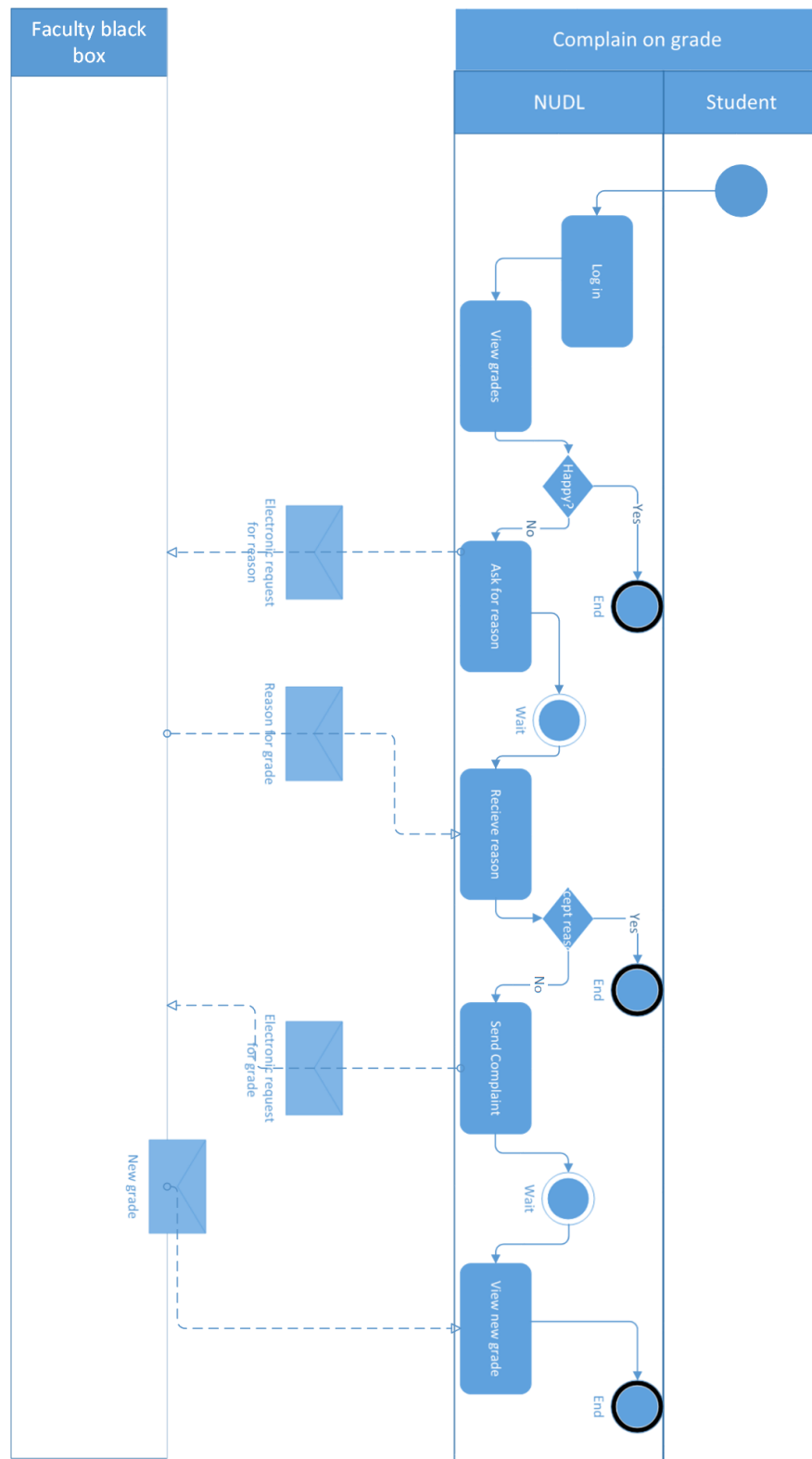


Figure 7: How we envision the new process of registering a complaint should be.

Security

Collecting all data in one system does raise more than a few questions about security. Our system will have to be designed with security in mind - from the ground and up. Although external breaches of the system security are ever-present issues today, there's also the possibility of internal misuse of the system and the data stored. Both of these sources of threats have to be considered and risks have to be identified to be able to provide proper mitigation. Although this is very interesting on its own, it is outside the scope of this project and will not be discussed further. Our focus on security for this course has been access and control.

Login

As mentioned earlier, login through FEIDE is an option and will most certainly be used for the low-level security. FEIDE provides the right mixture between security and practicality for everyday use. We do acknowledge that using one login for the whole system would seem practical, but it also means that it would be very easy to get access to other students private information. Therefore will our system include a second security level for which the user will have to provide a generated token to log in. Two-factor authentication increases the security of the systems by many factors, and is to our eyes crucial. Today's system uses a 4-digit pin-code for the user to get access to all grades. This is too weak, and we are actually quite astonished that this is still in use.

Authentication methods

To make the system flexible, we want to design login around pluggable authentication modules, allowing easy upgrades to the system if one technology becomes obsolete. To provide the two-factor authentication there are several possibilities, however - we do think that the best solution is the systems where the users receive an SMS with a one-time code.

Access control

Authentication is not everything, just because a user is logged in does not mean they are privileged for any action. To make sure that information is secure and is kept secure the system will be designed with redundant access control layers and separation of duty for staff to make sure that information is kept where it's supposed to - in the system.

Discussion

So far in this report, we've tried to show what we think are some of the big problems with today's solution, and how we would go about to fix it. We are aware that NTNU has invested a lot of money in It's Learning and that for many different, some of them not even "political" reasons NTNU will probably not even consider replacing It's Learning. One of the obvious reasons are that enough people actually seem to be content with what they get. As technology students, we have higher expectations than most as to what a computer system should be, and we do see a lot of shortcomings in the architecture of today's system.

A lot of the designs made in this project is obviously based on guess-work, and therefore some of our ideas might be more or less impossible to realize. As this has been a relatively small course we've just assumed that what we've designed will work with the existing systems.

We think that our vision of a new LMS for NTNU should be tailored for NTNU, by NTNU. We want NTNU to own the system as well as the data. It will obviously be a bit of work to maintain, but it could foster a nice number of Master thesis' and PhD's, as well as the savings provided by not having to license systems from other providers. The biggest payoff would however be that students would be more pleased, more productive and have much better experiences using LMS as a part of their education. If NUDL becomes a success, nothing stops NTNU from re-selling it to other institutions, and over time the system could even pay for itself.

If made properly, NUDL should be more secure than today's solutions, as well as provide the users with a consistent user-experience across different platforms and parts of the system. Having a system which behaves consistently and looks consistently, should be a security factor on it's own, by reducing the likelihood of successful phishing attacks or insecurity due to wrong usage.

Conclusion

In this report we've shown that today's multitude of systems is a confusing mess for students and staff, hindering them in working effectively and providing little help in day-to-day stuff, except digitalizing what would be done manually in older times. Our solution is to remove several of today's systems, and use the rest as subsystems for our new LMS-platform; NUDL.

NUDL is designed to be built as a modular system with low connectivity and high security. Providing API's allowing faculties to customize how they use it in their work, and students to develop tools to simplify their day. Much of the philosophy behind NUDL is shared with many pioneers in open source: We want to provide a rich, free to use, toolset for everyone to utilize. By opening up the system we democratize the digital learning process. By allowing everyone to participate and chime in, (almost) everybody wins.

Not only is NUDL intended to be an LMS, but we also want it to be a CMS for the whole of NTNU. This way one makes sure that the information that is made public is consistent with the information that isn't. Creating NUDL will require much more research and man-hours, but we think it's the only way out of the problems that are inherent with the existing solution.

References

- [1] BIEN, A. *Real World Java EE Patterns - Rethinking best Practices*, kindle ed. press.adam-bien.com, 06 2009, p. 840 (Kindle Location).
- [2] GOOGLE. Google Zeitgeist 2011 - Norway top searches. <http://www.googlezeitgeist.com/en/top-lists/no/top-searches>. [Online; accessed 02. May 2013].
- [3] RAMBØLL. Evalueringsrapport bruk av It's Learning ved NTNU. http://www.ntnu.no/c/document_library/get_file?uuid=cc8a29fa-84f4-44b3-9af4-36e6c486746c&groupId=524136. [Online; accessed 02. May 2013].
- [4] SILVER, B. *BMPN Method & Style*, 2nd ed. Cody-Cassidy Press, 2011.