Version 2.1

13-09-2013, 12:30
Last changed by: Soren Lauesen
slauesen@itu.dk

Requirements and Proposal for
Grant Management and CMS
(below called the Grant System)

Customer
The Y-Foundation

Supplier
Supplier C

The supplier's reply is in red
Contact: slauesen@itu.dk

The delivery comprises
Software, operation and maintenance of the Grant System

Contents

See other project documents, e.g. supplier selection and RFC vs. bugs, on:
http://www.itu.dk/people/slauesen/SorenReqs.html#Y-Foundation

This document is based on *Requirements Template SL-07*. The template and the Grant System requirements (© Soren Lauesen, 2017) may be freely used on the condition that this copyright clause is stated in the document.

Change log

Version 1.0: 20-02-2013. First draft of the future flow.
Version 1.1: 28-02-2013. First draft of Chapter B to E.
Version 2.0: 10-03-2013. Major changes of Chapter A to E. First draft of Chapter F and G.
Version 2.1: 13-03-2013. Minor changes of Chapter A to G. First draft of Chapter H to L.

Version 2.2: 23-03-2013. Issues from stakeholders and the related requirements changes:
1. Auditor: Protect account numbers, log changes, etc. C14-4, H5-5, D0-1 to 3.

Secretary:
2. When handling of the case has started, the applicant cannot change the application or its appendices. C10-4, C41-4.

Domain expert for engineering:
3. Possibility for “my own space” for notes for each application. C20-6, D5-9 and 10.
4. Possibility for printing overview of the applications with own notes. C20-8.
5. Possibility for transferring them to Excel for own analyses. C20-9.
6. Possibility for categorizing applications, make statistics, etc. C20-1, 2, 9 and E2-2 to 4.

Web-editor:
7. Need for application statistics for use on the public web site. C20-1, 2, 9 and E2-2 to 4.
8. Possibility for newsletter. C30-10.
9. Possibility for seeing applications suitable for publishing. C31.
10. Possibility for a "share-button" for sharing with Facebook, etc. (To be discussed further).
11. Possibility for larger web-pictures and access from smart phones and tablets. (This is hardly a requirement to the supplier, but to the Foundation’s web editor).

Version 2.3: 28-03-2013. Issues from stakeholders and the related requirements changes:
Auditor:
1. No need for individual creditor accounts. C10-34, D2-6 and the diagram.
2. The account number belongs to the applicant. D3-4, D6-18 and the diagram.

Focus group 04-03-2013:
3. Want to see the evaluation criteria. C40-1.
4. Want to see the arguments for rejections. C41-12, C10-11 and 21, C13-11.
5. No need for seeing the state of the application during the process. Deleted in the introduction to C41.

Domain expert for medicine:
6. Classify as medicine/engineering and retrieve it in overviews. D2-7, C20-1 and 7.
7. On the application overview, see recommendations, accumulated amounts and suitability for publication. C20-3 and 4.
8. You may want a discussion at the meeting even when you have recommended an amount. (Can simply be done by not indicating ”grant”. We can of course also add one more application state, but let us see what the suppliers suggest).
9. It must be possible for more than one person to look at the application at the same time. Added in C20-users, C21-users and L1-Nominal load.
10. The medical area should not be called ”doctors”. (Changed in various places).
11. Backup-requirements missing. Probably covered by H3-2 and L2.

Version 2.4: 10-04-2013. Issues from board members, etc.:
1. Specify the relation between proposal and contract. A2.
2. Icons to support recognition of applications, traffic lights for states. C20 solution note.
3. Possibility for indicating that you are positive about an application, but still want to discuss it at the board meeting. D5-3.
4. Available amounts per grant area per round. C20-4, D1-3 and 4.
5. Downplay the web supplier and emphasize that others must be able to operate and maintain the system. Work area 3, introduction, G1 and J6.

Version 2.5: 06-06-2013. Changes after first meeting with suppliers.
1. Drop the requirement that Economic is to be used for accounting. Payment can go through the accounting system or other ways. A1, F7.
2. Need for bulk mail with individual additions. C13-12p.
3. One common creditor account suffices. C14-7.
4. Need to see other board member’s recommendation live during the meeting. C21-1.
5. Maintenance of the web site in Danish as well as English. C30-60.
6. Wish that iPhone etc. could be used to look at applications during the board meetings. G1-3.

Contract version 1.0: 28-06-2013. First draft of requirements and solutions.
Contract version 2.0: 12-09-2013. Including all changes agreed during contract negotiation.
Contract version 2.1: 13-09-2013. Solution note for C30: Changes relating to modifiability of the web site for upload, and to handling of cases where the web site is operational while the upload part is down.

A. Background and overall solution

A1. Background and vision

Twice a year the Y-Foundation grants applications. There are two grant areas: Engineering and Medical. In total the Foundation receives around 300 applications at each round of application. The Foundation has two full-time employees.

The board of the Foundation has four members - two business members and two domain experts, one in engineering and one in medicine.

The board decides which applications to grant. Next, it is a clerical task to send accepts or rejects to the applicants, pay grants and receive final reports. The entire flow is described in section B1. The detailed requirements are in Chapters C and D.

At present the customer uses the C5 product for accounting, but is willing to change to eConomic, Nav or another system.

The customer uses Microsoft Office and Outlook, and is satisfied with this. The existing site is Globalsite. Bysted has developed it and the customer maintains the contents with a part-time web-editor. Today applications and correspondence are filed in paper form in traditional file folders. Addresses of contact persons, etc. are in Excel and are merged from here to letters and lists in Word. Chapter M shows examples of applications and summary lists used at board meetings.

The customer imagines the new system to be an integration of a case-management system (or CRM-system), a CMS system (the web-part) and an accounting system. As far as possible the system should consist of one or more existing systems configured for the customer's work. The present situation and a potential future situation are shown in Figure A1 and A2.

Figure A1: Existing system

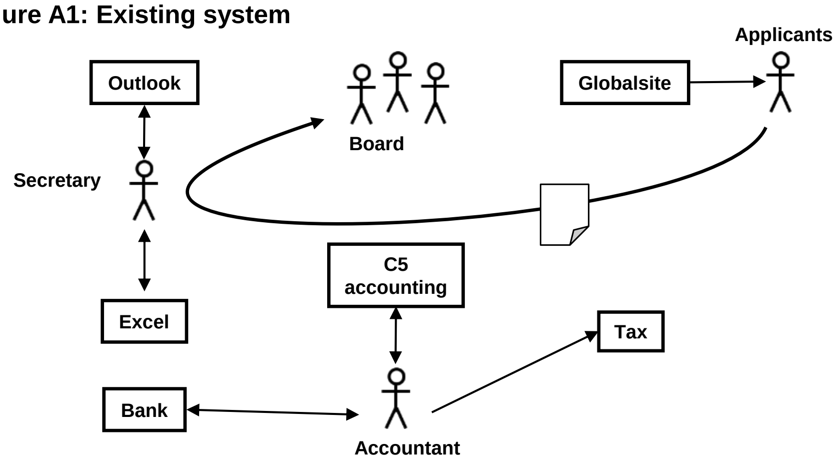
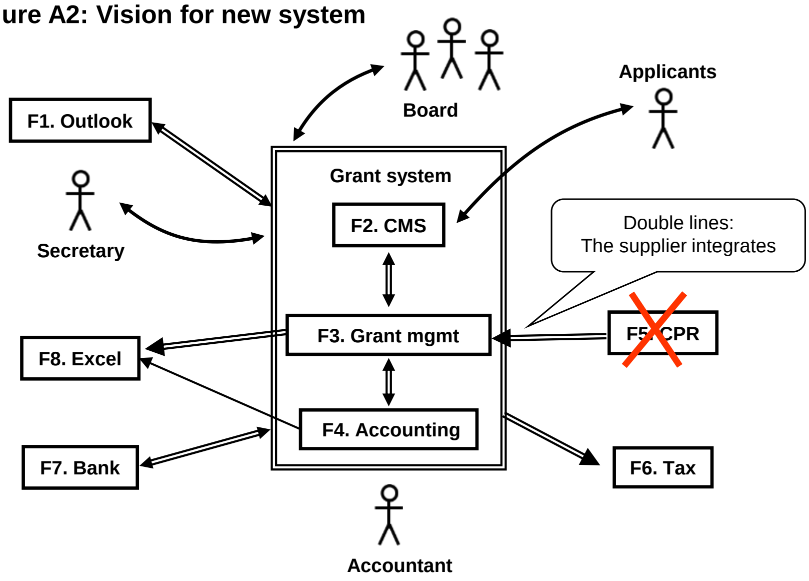


Figure A2: Vision for new system



A2. Supplier guide

This section explains the requirements format.

All requirements are written in tables:

- Column 1 is the requirement (the customer's demand - what he wants the system to support).
- Column 2 may contain the customer's solution example. In the supplier's reply, column 2 is a short description of the proposed solution. It must be in red.
- Column 3 may be the customer's rating of the proposed solution, test references, etc.

The requirements are organized in chapters according to their kind, e.g. Chapter C about user tasks to be supported, Chapter H about security. Within each chapter, the requirements are written in tables, e.g. a table with requirements relating to a specific task.

The customer's solution examples are only for inspiration. The supplier is welcome to suggest completely different solutions. They become legal requirements when both parties have accepted them. However, in case the accepted solution doesn't meet the demands stated in column 1 in a reasonable manner, column 1 has priority.

Initially, the customer wants an informal meeting with the supplier where the parties walk through the tasks and other requirements. The customer expects that the supplier shows how his system can support the various tasks and meet the other requirements. It might be done by showing a similar application.

The customer doesn't expect that the supplier has attempted to fill in column 2 at this stage.

Text outside tables

Text outside the tables can serve several purposes:

A. **Assumptions** behind the requirements, for instance that the task must be supported for this kind of users, this frequency of use, etc.

B. **Requirement notes** that elaborate column 1 in the table. In principle they should be inside the table, but they don't fit well. One example is a list of access rights to the system.

C. **Solution notes** that elaborate column 2 in the tables. They are not requirements but example solutions. One example is various ways a user can look up a code in a table.

D. **Examples** and other information to help the reader understand the requirements.

A3. Overall solution

The supplier proposes an integrated solution based on:

1. Workflow System C, which is based on Microsoft SharePoint
   a. Tight integration with Microsoft Outlook in such a way that emails easily can be attached to the grant applications.
   b. The solution is based on supplier C's workflow system, configured to the Y-Foundation's unique requirements. This simplifies support of the system because there are no customer specific changes to be maintained.
2. Microsoft Dynamics NAV (Navision). Among other things, the accounting system will handle payment of grants through the bank and reporting to the Tax authorities.
3. Wordpress as CMS. The supplier configures it as a web-site with basic application functionality.

The system is operated by three specialized operators: One for Workflow system C, one for Navision and one for Wordpress. The latter might be operated by the hosting supplier that also operates the SharePoint part. The reason for this split is to use the optimal hosting model.

Hosting of Navision is offered through a supplier who already hosts many Navision installations and offers 99% availability. The Wordpress-based CMS is hosted at a supplier who offers 99%+ availability on the web-site, because their focus is on availability. Finally Supplier C hosts Workflow System C with an availability of 98%. The systems are loosely connected so that downtime of one system doesn't imply downtime of the other systems. In this way the perceived availability is higher.

The supplier assists the customer with the deployment and provides support and maintenance as needed.

Apart from Workflow System C, there are no special programs, and anyone with knowledge of the Microsoft products will be able to support the solution.

The supplier offers the following alternatives:

Alt A. Applicants upload not only their application, but also later documents, e.g. proof of enrollment at a foreign university, progress report and annual account statement. In this way the Foundation secretary avoids handling a large amount of e-mails and documents. Communication with the applicants can be done through the Foundation's web-site, where the applicant also can follow what happens to the application. The applicant can subscribe to application updates and get mail notification in case of changes. In this way the Y-Foundation can provide solutions that fit the individual applicant. The Y-Foundation has rejected this alternative, and requirements related to it are not valid.

Alt B. A shared database of contacts is provided so that all persons automatically are recorded here. In this way, the Foundation can easily find persons who have applied earlier. Unique keys are civil registration numbers (CPR) and email address. To get the full benefit, the shared contact database must be updated on a regular basis. The Y-Foundation has accepted this alternative. As a result, the system will contain a shared contact database.

B. High-level demands

This chapter explains how the customer's work flow and business goals are met, how to mitigate high-risk requirements, and how to compare proposals.

B1. Visions about the future work flow

The Foundation's vision is to handle applications according to the flow below in the future. Only the normal flow is shown. The related tasks and subtasks are shown for each step. Chapter C shows the details.

Steps in the flow	Tasks and subtasks
1. The applicant investigates what can be granted and formalities about the application.	Entire C40.
2. The applicant fills in the application form on-line, attaches various documents and sends the application.	C41-1 to 5.
3. The Foundation secretary makes a first assessment of the application with two possible outcomes:	
A. The application is passed on to the proper domain expert (applies to around 250 applications at each round of applications).	C10-13.
B. The application is rejected immediately (applies to around 50 per round).	C10-11.
4. Every now and then, the secretary informs the domain experts that applications are waiting to be assessed.	C10-13.
5. The domain expert looks at the application and the history trail of the applicant, records his assessment and a suggested grant amount.	C20-1, 2, 4 to 6.
6. The other board members look at the application and record their assessment in the same way. When everybody agree, the application is passed on to the secretary.	C20-3 to 5.
7. At the biannual board meeting, the board decides the other applications. The board members can see the application on-line, including the other member's assessment. Each of them may print overview lists in advance.	C21-1, 2 and 3. C20-8.
8. The secretary records the decisions and sends them to the board members for approval.	C12-1 and 2. C13-1, 2 and 3. Entire C22.
9. The secretary sends letters of acceptance or rejection to the applicants.	C13-11 and 12.
10. The applicant sends supplementary documents, e.g. proof of admittance to the foreign university applied for, and confirmation that grants haven't been received from somewhere else too.	C41-11.
11. The secretary approves the documentation.	C10-32.
12. The grant system reports personal grants to the tax authority system. This also serves to validate the applicant's civil registration number (CPR).	C14-5. C10-12.
13. The grant system creates a file for money transfer through on-line banking. The external accountant and the secretary approve amounts as well as account numbers. Next, the transfer takes place.	C14-1 to 4 and 6 to 8.
14. The secretary informs the applicants that the money has been transferred.	C14-9.
15. During the grant period (the project period), other amounts may be transferred. It takes place as in steps 10 to 14.	C10-41 to 45. Entire C14. C41-14.
16. When the project terminates, the applicant sends a final report.	C41-14.
17. The secretary sends a reminder if the final report doesn't arrive.	C10-42 and 45.
18. The secretary approves the report and closes the grant case.	C10-41 and 46.

B2. Business goals

The customer's reason to acquire the system is to reach some business goals. The customer expects that the system contributes to the goals as stated below. The supplier can rarely reach the goals alone. Customer contribution is needed too. This means that the goals are **not requirements** to the supplier. They are shown in a table only to provide overview.

All goals are important and the sooner they can be met, the better. Some goals are crucial to meet at a specific date, for instance for business or legal reasons. There are no such deadlines in this project.
.

Goals for the new system	Solution vision	Related requirements
1. Better support of the grant management, particularly close to application deadlines.	a. File all data on an application electronically and distribute it electronically. b. Simplify accounting and grant payment. c. Automatic reminders.	C10, C12, C20, C21. C14. C10-37 og 45, C12-2, C13-12.
2. Less stress on the board close to the board meeting.	a. Applications are assessed continuously, particularly study travels. b. The board can see each other's assessments.	C20, C21.
3. Better communication with the audience through the web-site.	a. Better information about who reads the applications and when a reply can be expected. b. Automatic confirmation of the application.	C40. C41-5.

B3. Early proof of concept

Some requirements are high-risk and the supplier may not be able to deliver what he promised in his proposal. If this is detected late in the project, the customer may terminate the contract, but this is a disaster to both parties. Usually the customer chooses to accept the inadequate system, possibly with compensation from the supplier. To reduce the risk, the customer requires an early proof of concept for the high-risk requirements.

Both parties may cancel the contract if the early proof fails.

The following requirements are considered high-risk. Deficiencies here can hardly be repaired late in the project. In his reply, the supplier must state how he will carry out the proof of concept and when. The date must be stated as the number of workdays after signing the contract. The customer expects 40 workdays or less.

Areas where an early proof is required:	Example of proof:	Code:
1. Usability, including efficient task support.	A prototype of the system or a functional version is tested as described in section I. Possible within _10 workdays.	N/A
2. Integration with the accounting system (see F4).	Payment transfers are carried out.	N/A

B4. Minimum requirements

Proposals are considered even if many of the requirements aren't met. However, there must still be a clear business advantage to the customer.

B5. Selection criteria

The customer will compare proposals based on how well the business goals are met and on the cost of system and operation.

Together with the supplier, the customer will look at all requirements and take notes on how they are met.

C. Tasks to support

The system must support all user tasks in this chapter, including all subtasks and variants, and mitigate the problems. Column 1 of the tables describe what user and system will do together. Who does what depends on the chosen solution.

A task is carried out from start to end without essential interruptions. If necessary, the case must be parked and resumed later.

Although subtasks are numbered, they don't have to be carried out in this sequence, and many of them are optional. The user decides what to do and in which sequence. A subtask may also be repeated during the same task.

Some subtasks may be performed in alternative ways. It is shown with a, b, etc. As an example, a request about a grant application usually starts with finding the application (subtask 2), but the secretary may instead create the application (subtask 2a).

Letters p, q, etc. indicate something that today is a problem with this subtask.

Work area 1: Grant management

At present grant management is carried out by one secretary.

User profile: Experienced MS Office user with deep knowledge of the work area.

C10. Handle a request about a grant application

This task describes what the secretary can do when a grant application or a request about an application arrives.

Users: Secretaries.
Start: The task can start in many ways: An applicant has created an on-line application, an e-mail about an application, an ordinary letter, by phone, an automatic reminder, an automatic message saying that the board has agreed on an application, or that the secretary wants to see which application is most important to deal with now.
End: When data about the application has been updated and we cannot do more about the application right now.
Frequency: Total: Around 300 requests in the weeks close to the application deadline. Less in other periods.

Subtasks and variants:	Proposed solution:	Code:
1. Maybe see an overview of the application states and select one of the applications.	The system shows a list with status, amount, etc. Sorting and filtering. It is the same kind of list as C20 (see the solution node). Alt. A: The secretary primarily uses this application list and Outlook less, because the applicants reply to questions and add documents directly from the application web-site..	
2. Find a specific application.	Search the application by application ID, part of the applicant name, email, etc. In addition automatic retrieval of the application based on an email with the application ID or the applicant's email address.	
2a. No application has been created. Assess whether to create one and maybe create it. Usually applications have been created on-line, but there may be exceptions.	When the application form on the web-site is used, the application is created automatically. When the application isn't created on-line, the secretary creates it manually, for instance based on an email.	

3.	See the application status and maybe update it.	The application status is visible at all times and is retrieved by the secretary. When the request arrives by email, it can easily be attached to the application as described in subtask 2 Next, the system assists with opening the related application and the secretary can update it according to the email contents. Alt. A: The applicant uploads data and the system updates the application status accordingly. The customer can configure the system so that status is changed automatically. Further a notification may be sent by email.	
4.	Maybe lock the application for further updates by the applicant.	In the basic version there is no need for a lock. The applicant can upload once only. Alt. A: The secretary can lock and open a specific application, but may also lock and open all applications for the current round at once.	
5.	File the request.	Requests received and replied by email are filed by attaching the mail to the grant application in Outlook. The user can use drag-and-drop or a track-button. Alt. A: Requests that relate to a specific grant application are filed on the application automatically.	
6.	Create organizations and persons with relation to the application or relate the application to existing organizations and persons.	The applicant is automatically created as a contact in the application. Alt. B: The system maintains a common database of contacts and their roles in the applications. Contacts are identified by civil registration number and email address.	
7.	Maybe change their data (see section D).	Manually through email in the basic version in such a way that changed account numbers based on emails are considered changed by the Y-Foundation. As a result there will always be a warning on them. Alt. A: The applicant changes the data on-line himself.	

Especially early in the flow:		Proposed solution:	Code:
11.	Reject the application if it obviously doesn't qualify. Explain the reason.	"Reason" might be a template that the secretary fills in. The secretary handles it application by application. The solution offers email templates that allow the replies to be standardized.	

12. Maybe validate the civil registration number where it occurs. Might be postponed to the time of payment, but it is nice to check it early.	Modulus-11 check (which nearly always must be correct). The system warns if the check fails, but allows it anyway.	
13. Assess whether a domain expert should be asked to look at the applications, e.g. because there are several unhandled or because the deadline is near.	Sent manually as email. As above, email templates are offered.	

When the application has been assessed by all board members:	Proposed solution:	Code:
21. Send rejection if everybody agree. Explain the reason.	As in C13-11, but doesn't have to be mass mails. The reason is typed in the main application record and later merged automatically into a mail template.	
22. Assign it for handling at the board meeting if wanted or if there is disagreement.	Automatically based on the application's state. When the secretary has screened the application, the board can in principle assess it.	

For accepted applications:	Proposed solution:	Code:
31. Send letter of acceptance and maybe ask for supplementary documentation, e.g. letter of acceptance at the foreign university applied for.	As in C13-12, but doesn't have to be mass mails.	
32. Approve documentation or ask for more.	Through manual change of status or email to the applicant.	
33. Maybe create an instalment plan for the payments.	The system has space for a plan, but the secretary fills it in manually. Initially the payment is created automatically as a single instalment to save time for the secretary.	
34. Create a creditor account for the application.	Navision automatically connects the account number with the application ID.	
35. Maybe adjust the payment plan.	Done manually.	
36. Order an amount to be paid later.	The secretary orders it in the Grant system, which later transfers it to the account system.	
37. Start time monitor for supplementary documentation, final report or instalment report.	Automatically at change of state, also without Alt. A.	

For accepted applications (where the first payment has been made):	Proposed solution:	Code:
41. Approve final report or instalment report.	Manual change of state.	
42. Maybe send a reminder.	The secretary can configure it to be done automatically. Alt A: In this version, the reminder can be attached to the applicant uploading the report on-line. The secretary must still check the report quality, of course.	
43. Maybe adjust the payment plan.	Done manually as needed.	
44. Order an amount to be paid later.	As C10-36.	
45. Maybe start time monitor for final report or next instalment report.	Automatically at state change.	
46. Maybe close the case.	Manual change of state.	

C11. Prepare board meeting

The applications to be handled at the board meeting are those in state *for_board_meeting.* The board just needs the usual agenda and can on their own see the applications in the system and print them as needed.

C12. During the board meeting

This task describes what the secretary does with the applications during and right after the board meeting.

Users:	Secretaries.
Start:	The board meeting starts.
End:	When the results have been sent to the board for confirmation.
Frequency:	Twice a year.

Subtasks and variants:	Proposed solution:	Code:
1. Note the result of each application, including changed amounts and whether the application is publish worthy.	The secretary does it directly in the system.	
2. After the meeting: Ask the board to confirm that everything is correctly recorded. Start time monitoring.	Through ordinary emails. Each board member might sign electronically: However, we believe that an email is simpler and sufficient.	

C13. Carry out the decisions

This task describes what the secretary does when the entire round of applications is approved..

Users: Secretaries.
Start: When approval is received, or changes are needed, or at a reminder that pproval hasn't arrived in due time.
End: When nothing more can be done right now.
Frequency: A few times after each board meeting.

Subtasks and variants:	Proposed solution:	Code:
1. Maybe correct errors and send for new approval.	Errors are preferably corrected in the application records. Next a new meeting summary is generated and sent to the board members for approval.	
2. Maybe send reminders to board members.	Reminders can be sent automatically based on a deadline.	
3. Record that the round of applications has been approved.	Final approval is made by the customer based on the approval from the board members. Electronic signatures are possible if desired.	

When everybody has approved the round:	Proposed solution:	Code:
11. Send rejection letters with explanations.	Explanation as a template that the secretary can check off. The system creates all mails based on a mail template, but doesn't send them yet. The secretary can change them individually before sending them. Automatic sending to all applicants is also possible if the individual explanation is limited to a single field on the application record.	
12. Send grant letter and maybe a request for supplementary documentation. Start time monitoring.	Template where the secretary can choose documents that are required. Alt A: With this solution, certain documents will be mandatory on the form that the applicant has to fill in before e.g. payments can be completed.	
12p. **Problem**. Many letters are almost identical and it is cumbersome to send them one-by-one.	The system creates all mails based on a mail template, but doesn't send them automatically. The secretary can change them individually before sending them. Standard letters can be sent as a batch while deviations are handled manually.	
13. Park the work.	As long as there are uncompleted applications in the round, it will be open for further action.	

The rest of the grant management takes place in C10 and C14.

C14. Pay grants

This task describes what secretary and accountant do when amounts are to be paid.

Users:	Secretaries and accountants.
Start:	A bunch of grant payments have to be made.
End:	When they are made.
Frequency:	A few times after each board meeting.

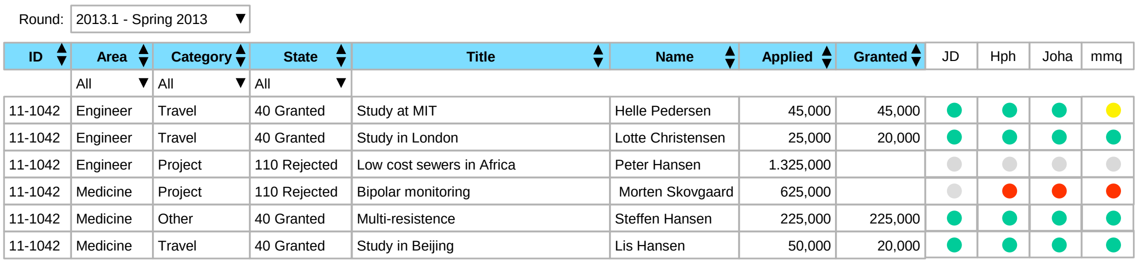
Subtasks and variants:	Proposed solution:	Code:
1. Make a list of payments with the applicant's name, bank account and amount.	The system creates the list, typically once a month. This can with advantage be done automatically based on the recorded instalments. The secretary is automatically advised that a round of payments are ready for approval.	
2. The secretary checks and approves the list.	The secretary approves the list and can review each payment. Payments to bank account numbers that have been changed after receipt of the application will be flagged clearly in the list.	
3. The accountant checks and approves the list.	Yes. The system transfers the list to Navision, which later carries out the payment. Clear flagging as above.	
4. Check that account numbers are what the applicant specified.	The system shows whether the account number originates in the on-line application. If not, both secretary and accountant must check against application-related documents. If the account number has been changed by the Y-Foundation, this will be clearly flagged on the payments in relation to the approval.	
5. Report personal grants to the tax authority and validate the civil registration number.	Takes place automatically through the accounting system. Validation of civil registration number, apart from modulus 11 check, is abandoned.	
6. Transfer the list to the customer's bank to carry out the money transfers.	Takes place automatically through the accounting system, but the secretary must approve with own digital signature.	
7. Create a creditor account for the applicant if this hasn't been done already.	Takes place automatically when the grant application is created in Navision, if the customer wants it.	
8. Transfer the list to the accounting system for posting on the creditor accounts.	Has been done already in subtasks 1-3.	
9. Send a message to the applicants about the payment.	The system sends an email to the applicant when the list has been transferred to Navision. Alt A: With this solution, the applicant is informed through the application web-site. When the applicant subscribes to changes by email, the applicant will receive a mail about changes relating to the application.	

Work area 2: The board

The area comprises assessment of applications.

Solution note

Board members access the system through a web-browser. This will work from a traditional PC as well as an iPAD via VPN. The solution is outlined below and presented to the Y-Foundation.



Round: 2013.1 - Spring 2013 ▼

ID ▲▼	Area ▲▼	Category ▲▼	State ▲▼	Title ▲▼	Name ▲▼	Applied ▲▼	Granted ▲▼	JD	Hph	Joha	mmq
	All ▼	All ▼	All ▼								
11-1042	Engineer	Travel	40 Granted	Study at MIT	Helle Pedersen	45,000	45,000	🟢	🟢	🟢	🟡
11-1042	Engineer	Travel	40 Granted	Study in London	Lotte Christensen	25,000	20,000	🟢	🟢	🟢	🟢
11-1042	Engineer	Project	110 Rejected	Low cost sewers in Africa	Peter Hansen	1.325,000		⚪	⚪	⚪	⚪
11-1042	Medicine	Project	110 Rejected	Bipolar monitoring	Morten Skovgaard	625,000		⚪	🔴	🔴	🔴
11-1042	Medicine	Other	40 Granted	Multi-resistence	Steffen Hansen	225,000	225,000	🟢	🟢	🟢	🟢
11-1042	Medicine	Travel	40 Granted	Study in Beijing	Lis Hansen	50,000	20,000	🟢	🟢	🟢	🟢

This screen is the key for all grant management in the system and is used continuously by secretary as well as board members. It is possible to filter on the various columns, including the "traffic lights" for the individual board members. In this way it is easy to see whether something is new. Board members can directly from this screen change state and write public as well as private comments for each application, and through the link access the application to see details. Depending on the specific configuration it will also be possible to access for instance the application document directly from the list.

C20. Assess applications before the board meeting

This task describes what a board member does before the board meeting.

Users: Board members. The four board members and the secretary need to see the applications at the same time and note their own comments directly in the system. See also access rights in H1.

Start: When an advice arrives from the secretary or a board member about looking at grant applications, or when it is time to look at them.

End: When all of them have been assessed or there is no more time available right now.

Frequency: Daily when the application deadline gets near.

Subtasks and variants:	Proposed solution:	Code:
1. Look at the applications you have to assess, i.e. those you haven't assessed and are for you as a domain expert, or are for the entire board.	The system shows a list of applications with status, amount, etc. See the solution note below. The user can sort and filter on the data.	
2. Maybe look at the applicant's history trail, i.e. other applications he has been involved in and granted amounts.	Visible on each application when it is possible to match on either civil registration number or email address. Further, the secretary may manually connect old applications in case civil registration number or email doesn't match.	
3. Look at other board member's assessment and whether it has been assessed as publish worthy.	Visible in the list. The field "suited for publication" isn't shown above, but can be shown too.	
4. Look at accumulated, granted amounts for each grant area, at the total amount applied for, and at the available amount (see D1).	Visible in the list. At the top the round is shown, including amounts set off for the different categories (not shown above).	
5. Note your own conclusion, comments and possibly changed amounts. Other board members will be able to see them.	Noted directly in the list.	
6. Note your private comments that are not intended for others.	Noted directly in the list.	

7.	When you have assessed applications as a domain expert: Advice other board members to look at these applications.	By changing the state from "not seen" to "seen", "accepted" or "rejected", the application will automatically be visible to other board members. The board member may also send an email.	
8.	Maybe print a list of applications and maybe their attachments. To be used in the preliminary assessment and later at the board meeting.	Data, sort and filter as in C20-1. Attachments of specific types (see D8) are included. It all ends up as one long document, e.g. a pdf-file, to make it easy to print. It is recommended that printing is kept at a minimum and replaced by sending mails with links to applications and key documents in order to (1) save printing and paper, (2) make it easier for board members to use the online solution.	
9.	Maybe transfer the list of applications with state, name, amount, category, etc. to own spreadsheet.	The user can easily do it. The user filters the data, next chooses to see the data in Excel for further analysis.	

Solution note

The following is proposed:

1. Each application on the list is identified by ID (a number) and name (short title), preferably also an icon (a small picture) that the applicant has uploaded.
2. Further this is shown: Application state, amount, grant area, category, etc. Preferably everything on one line.
3. State may be shown as a grey-yellow-green-red dot for each of the four board members.
4. A click can show details of the application.
5. When an application is changed, the lists on other screens are updated automatically as outlined above in the "traffic lights".

C21. During the board meeting

This task describes what the board members do with the applications during the meeting.

Users: Board members. The four board members and the secretary look at the applications at the same time and note their own comments directly in the system. See also access right in H1.
Start: When discussion of the applications start.
End: When all applications have been discussed for now.
Frequency: Twice a year.

Subtasks and variants:	Proposed solution:	Code:
1. Look at each application. See what the other board members mean, preferably live as soon as they have indicated something. Look at the full application and attached documents.	As C20. The system updates the list of applications without the board members having to click a "refresh".	
2. Enter your own comments that are not intended for others.	As C20.	
3. Maybe record the joint conclusion.	As C20.	

C22. After the board meeting

This task consists of checking the decisions and approving them.

Users: Board members.
Start: When a message arrives from the secretary about approving the minutes from the board meeting.
End: When the minutes have been approved, or rejected with comments.
Frequency: Twice a year.

Subtasks and variants:	Proposed solution:	Code:
1. Look at each application and check the decision against own notes.	The system checks the decision against the board member's notes during the meeting (D5-6 and 8). We suggest that the system automatically generates a summary of all the applications and their decision in which board member's public comments and joint decisions are shown. In this way an automatic summary can be made. After this, the applications are locked and cannot be changed.	
2. Maybe reject the minutes with comments.	Through ordinary email.	
3. Approve the minutes.	Through ordinary email or electronic signature of the board members.	
4. Maybe transfer the list of applications with state, name, amount, category, etc. to your own spreadsheet.	The user can easily do it.	

Work area 3: Web editor

The area comprises creation and maintenance of the customer's web-site for several audiences: Potential applicants, applicants during creation of the application, the press and the public. The customer expects that the supplier delivers a CMS with basic application functionality and contents. The customer maintains the contents with his own web editor.

It is important that the web site uses a widely used technology in order that many suppliers can maintain it and that it reasonably well supports the subtasks listed below. If this can be done with the existing Globalsite (from Bysted), it is an advantage, of course, but not decisive.

Solution note

In the future the customer will use Wordpress for his web site. The supplier delivers various templates that the customer can use to create the site, and will instruct the customer on how to do it.

The web-page where the applicant uploads his grant application is created by the supplier, as far as possible based on the existing web-page. It is on a different web-server than Wordpress. If the customer later wants to change it, it is a maintenance task.

In cooperation with the customer, the upload site is usability-tested before the system test in order to reduce the maintenance need.

The supplier handles the situation where the customer's web site is operational, while the upload server isn't. The solution might be that the applicant sees a message that the upload temporarily is out of order and that you as the last resort may contact NN.

C30. Edit the customer's web-site

Users: Web-editor or an employee that has got this role.
Start: When time allows.
End: When there isn't more time right now, or when the new version is sent for approval or published.
Frequency: Occasionally.

Subtasks and variants:	Proposed solution:	Code:
1. Add or remove a web page.	Part of Wordpress.	
2. Add, modify or remove templates and text.	Part of Wordpress.	
3. Add, modify or remove meta-data, e.g. keywords, time of change, validity period.	Part of Wordpress.	
4. Add, modify or remove links to media file, e.g. picture or video.	Part of Wordpress.	
5. Add, modify or remove links to other pages on the customer's site or the internet in general.	Part of Wordpress.	
6. Add, modify or remove parts with animation, e.g. automatic change of picture.	html5 support. Part of Wordpress.	
7. Add, modify or remove parts for upload of applications (see solution note above).	Later changes are a maintenance task and not web-editing.	
8. Add, modify or remove pages that require login, e.g. pages that show the state of an application.	See solution note above. Later changes are a maintenance task and not web-editing. Functionality for upload/download/filling in the application and display of application states, etc. are not part of the solution. In case of requests for change, contact the supplier.	
9. Add, modify or remove parts that show dynamic data, e.g. the state of an application or a list of applications.	Is a maintenance task and not web-editing. See subtask 8.	

10. Add, modify or remove newsletters.	Part of Wordpress and maybe mailchimp or a similar system for sending newsletters. This is not a core part of Wordpress. We recommend e.g. http://mailchimp.com which has integration with Wordpress. The supplier can assist with setting up mailchimp. The customer must pay for using mailchimp.	
11. See how the web pages will appear when they are published, but without publishing them.	Editing is done in another context, e.g. a staging server. Part of Wordpress. It is possible to create content as a private draft and in this way see what the web pages look like. There is no staging server in Wordpress.	
12. Test upload parts and parts that require login without publishing the pages.	The upload page and in Alt A the user site, are not part of the web-site, but are located on another server.	
13. Undo changes.	Part of Wordpress.	
14. Send the new version for approval, e.g. by an employee or the board.	Part of Wordpress.	
15. Publish the new version.	Part of Wordpress.	

Particularly about templates:	Proposed solution:	Code:
20. See which templates are available and where they are used.	Part of Wordpress.	
20p. Problem: Hard to see what a template does just based on its name.	Show what it looks like when filled out?	
21. Add, modify or remove templates.	Part of Wordpress.	

Particularly about media files:	Proposed solution:	Code:
30. See which media files exist and where they are used..	Part of Wordpress.	
30p. Problem: Hard to see what the file shows just based on its name.	Show a preview in the overview. Part of Wordpress.	
31. Add, modify or remove media files.	Part of Wordpress.	

Particularly about links:	Proposed solution:	Code:
40. Check all links and mark those that are blind.	Part of Wordpress.	

Particularly about upload parts for applicants:	Proposed solution:	Code:
50. Add or modify login parts or user subscription.	Is a maintenance task and not web editing.	
51. Indicate what to upload always and what is optional. Specify validation rules.	Is a maintenance task and not web editing.	
52. Specify what happens when the user has uploaded something, e.g. which data in the data model to add and who to inform.	Is a maintenance task and not web editing.	

Particularly about English pages:	Proposed solution:	Code:
60. Maintain pages in an English as well as a Danish version. It is hard to do manually.	Not resolved. We recommend to create a Danish web site and an English one to ensure better search engine optimization. The pages must then have meaningful names in each language. Wordpress doesn't support the possibility to notify the web-master when a change is done in one of the languages, but not the other.	

C31. Publish selected projects

Users: Web-editor or an employee with this role.
Start: After a board meeting where new applications have been granted.
End: When there is nothing more to do right now or the new version has been sent for approval or published.
Frequency: A few times a year.

Subtasks and variants:	Proposed solution:	Code:
1. See applications marked as publish worthy.	The Web-editor gets access to the grant system.	
2. Contact the applicant about pictures, etc.	Manually. Alt A: Might be done through the application web site, but currently not part of the proposal.	
3. Edit the web page based on the application, pictures, etc..	Manually. Alt A: Might be done through the application web site, but currently not part of the proposal.	

Work area 4: Applicants and the public

The area comprises potential applicants, applicants with an application in progress, the press and the public.

C40. Visit the Y-foundation's web-site

Users:	Applicants and the public.
Start:	When the user wants to know something about the Y-Foundation.
End:	When there isn't more of interest right now.
Frequency:	In total: Not known.

Subtasks and variants:	Proposed solution:	Code:
1. See what the Y-Foundation supports, how to apply, and how the application is assessed.	The customer's own task.	
2. See examples of granted projects.	The customer's own task. It is possible to expose the application archive or parts of it on the web-site as a search corresponding to the filtering the customer will apply internally. This is not part of the proposal, but might be offered as a few days of additional work.	

C41. Apply for a grant

Handling of a grant application may take a long time. The applicant wants to know when a decision can be expected.

Users:	Potential applicants.
Start:	When the user wants to apply for a grant or gets a mail from the foundation about the application.
End:	When nothing more has to be done right now.
Frequency:	In total: 300 applications twice a year. Close to the deadline probably 50 per hour.

Subtasks and variants:	Proposed solution:	Code:
1. Provide or edit applicant data and see any error or omission.	See solution note to work area 3.	
2. Upload the necessary documents.	See solution note to work area 3.	
3. The applicant lacks documents and wants to park and resume the application.	Possible with Alt. A, i.e. not possible with the chosen solution.	
4. Send or resend the application. Can be done as long as handling of the application hasn't started.	Resend only possible with Alt. A, i.e. not possible with the chosen solution.	
5. Receive confirmation that the application has been received and when a decision can be expected.	Sent automatically by email.	

Particularly when handling of the application has started:	Proposed solution:	Code:
11. Send supplementary or corrective information or documentation, e.g. proof that the applicant has been admitted at a foreign university.	Send with ordinary e-mail. The secretary assesses it and updates the application (see C10-32). Possible on-line with Alt. A.	
12. Receive the decision and understand the explanation.	Ordinary e-mail, maybe with an attached document. Alt. A: Done on-line.	
13. Receive message about payments.	Ordinary e-mail. Alt. A: Done on-line.	
14. Send final report or maybe instalment report.	Ordinary e-mail. Alt. A: Done on-line.	

D. Data to record

The system must record the data described in this chapter. The user can create, view, and change the data through the tasks described in Chapter B4.

Figure D is a data model (an Entity/Relationship diagram, E/R) that gives an overview of the data. Each box is a class of data. Imagine a pile of file cards behind the box. The box symbolizes one of the cards. As an example, D2 is a pile of cards that holds a card for each grant application. Next to the box is a list of the fields on the card.

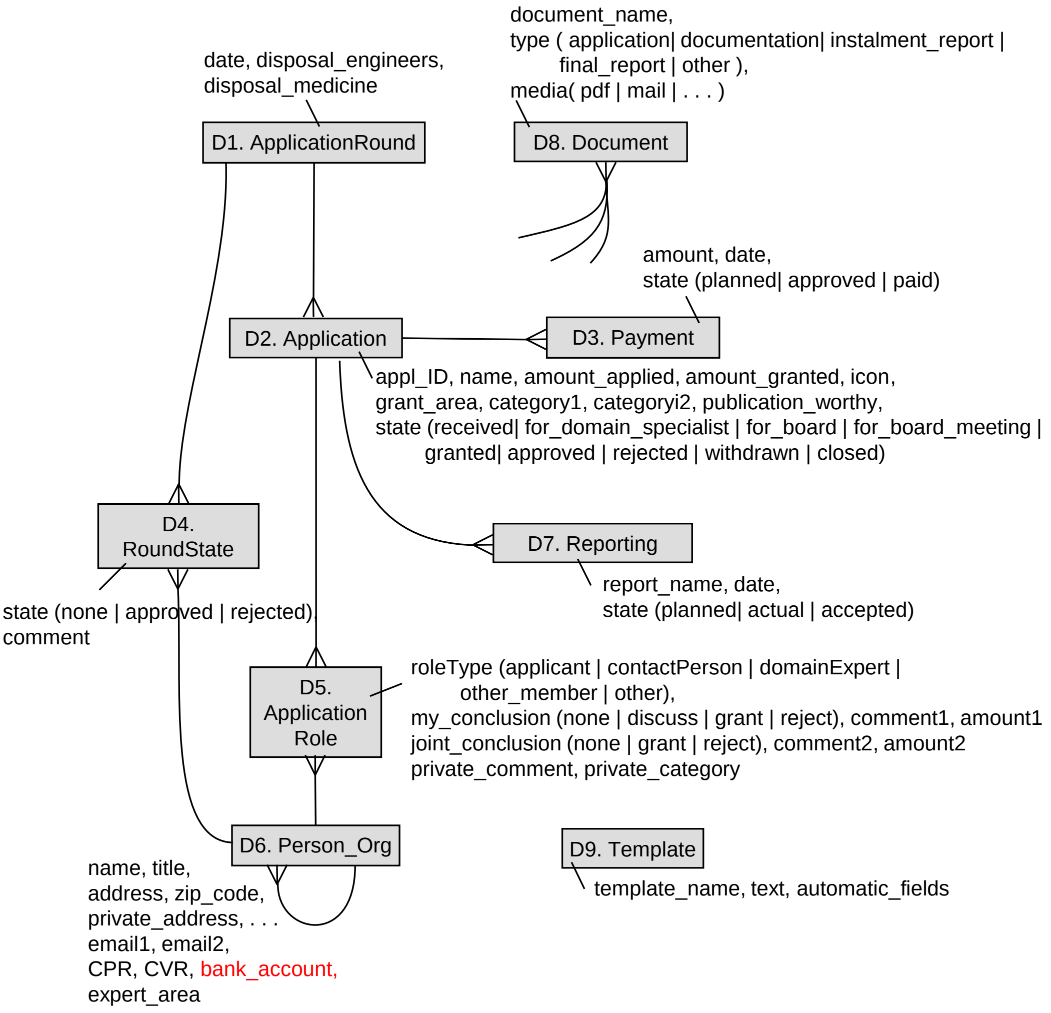
There are relationships between the boxes, shown as crow's feet. A crow's foot shows that a card relates to one or many cards in another pile. As an example an application may have several payments, but a payment may relate to only one application. Data need not be structured this way in the system, but it must be handled in some way.

Solution note

All data below are offered unless something else is stated. Minor changes, e.g. new states or fields, are agreed on during development. We have already observed a need for more dates in D1-ApplicationRound (deadline, meeting date, reply date) and for more creditors to the same application. In order to support more creditors, the data model below must be changed. It is only possible to receive one applicant with account number from the web form, so the customer must manually add receivers with their account number by email.

The following pages describe the boxes in detail.

Figure D. Data model for the Grant System


document_name,
type (application| documentation| instalment_report |
final_report | other),
media(pdf | mail | . . .)
date, disposal_engineers,
disposal_medicine
D1. ApplicationRound
D8. Document
amount, date,
state (planned| approved | paid)
D2. Application
D3. Payment
appl_ID, name, amount_applied, amount_granted, icon,
grant_area, category1, categoryi2, publication_worthy,
state (received| for_domain_specialist | for_board | for_board_meeting |
granted| approved | rejected | withdrawn | closed)
D4. RoundState
D7. Reporting
report_name, date,
state (planned| actual | accepted)
state (none | approved | rejected),
comment
D5. Application Role
roleType (applicant | contactPerson | domainExpert |
other_member | other),
my_conclusion (none | discuss | grant | reject), comment1, amount1
joint_conclusion (none | grant | reject), comment2, amount2
private_comment, private_category
D6. Person_Org
D9. Template
template_name, text, automatic_fields
name, title,
address, zip_code,
private_address, . . .
email1, email2,
CPR, CVR, bank_account,
expert_area

D0. Common fields

Each data class records history, i.e. each change creates a new version of the "file card" and preserves the old one. Each "file card" may also relate to documents about the file card.

Fields and relationships:	Proposed solution:	Code:
1. Change Time: Date and time when the "file card" was created or changed.	Flowsystem C contains standard fields that store this information.	
2. Source: The person who created or changed the "file card".	Flowsystem C contains standard fields that store this information.	
3. History: Relation to earlier versions of the "file card" (not shown in the diagram).	Agree later what needs to be logged.	
4. Documents: Relation to zero or more documents that belong to the "file card". Any data class may have related documents. For some data classes this is elaborated below.	Documents relate to a grant application and are placed in a folder for that application. Some documents may have a special status and be visible on the application form.	

D1. ApplicationRound

An application round is the anchor point of the applications to be handled in the round.

Examples: The application round with deadline 12-04-2014.
Data source: New rounds of application are created by the secretary.
Data use:

Data volume:	Proposed solution:	Code:
1. Two rounds are created per year.	Rounds may be created automatically biannually as work flows.	

Fields and relationships:	Proposed solution:	Code:
2. date: The application deadline.	Field on the record for a round.	
3. disposal_engineers	The amount available in the engineering area in this round.	
4. disposal_medicine	The amount available in the medical area in this round.	

D2. Application

Examples: An application about study at a foreign university. An application about low cost sewers in Africa.

Data source: An application may be created by a secretary or by an applicant through the web site. Application data can be changed by a secretary and some fields by board members.

Data use: The application is used during the grant process.

Data volume:	Proposed solution:	Code:
1. Around 600 applications are created a year.	Created automatically based on applications from the web-site plus the secretary when applications are received by email.	

Fields and relationships:	Proposed solution:	Code:
2. appl_ID.	Assigned automatically by the system.	
3. name: A name the applicant specifies. May be changed by the secretary.	Yes.	
4. amount_applied: The total amount applied for.	Yes.	
5. amount_granted: The total amount granted.	Yes.	
6. creditor_account: (The account number where payments to the applicant is recorded). Isn't used.	(See bank account in D6.) There should be an account number somewhere (IBAN)	
7. grant_area (engineering | medicine).	Yes	
8. category1: e.g. travels, publications . . .	The customer maintains this category.	
9. category2: e.g. a priority.	The customer maintains this category.	
10. publication_worthy: yes/no. Indicates to the web-editor that the application should be shown on the web site.	Yes.	
11. state: See the requirements note below.	The states outlined below are supported.	
12. applicationRound: Relation to D1.	Yes.	
13. payments: Relation to one or more payments.	When an application is granted, a payment matching the granted amount is created automatically.	
14. applikationRoles: Relation to D5 that links to the persons and organizations that have something to do with the application.	Alt B: If a common list of contacts is used, applications from the various contacts can be retrieved quickly.	
15. reports: Relation to one or more reports during the grant period.	Reports are extracts of relevant data, e.g. applications filtered by round, category, etc. The reports here are documents submitted by the applicant during the grant period.	
16. documents: Relation to the documents that belong to the application.	Documents can be attached to the application and may be special documents, e.g. a status report.	
17. icon: A small picture that the applicant may upload. Will show in the overview lists of applications.	If the applicant uploads a picture, it will be shown.	

Requirement note:
The state of the application can be one of these:

1. received.
2. for_domain_specialist: The secretary has looked at it and passed it on to the board's domain specialist.
3. for_board: The domain specialist has looked at it and passed it on to the other board members.
4. for_board_meeting: The board members haven't agreed before the board meeting.
5. granted: The board has before or on the board meeting decided to grant the application on the condition that the necessary documentation will be sent.

6. approved: The application is granted by the board, and the secretary has approved the documentation, e.g. proof of admittance at the foreign university.
7. rejected: The application is rejected already when received or later by the board.
8. withdrawn: The applicant has withdrawn the application.
9. closed. The case is closed.

D3. Payment

Each application has one or more planned payments. They are carried out as the project proceeds or is adjusted.

Examples: An annual or quarterly payment to a project.
Data source: Created by a secretary or created automatically at an on-line application. A secretary can change the payment data.
Data use: Used to forecast the Foundation's total payments and to carry out the actual payments.

Data volume:	Proposed solution:	Code:
1. Around 400 payments are made a year.		

Fields and relationships:	Proposed solution:	Code:
2. amount.		
3. date: The planned or actual payment date.	Both fields are supported. The planned date is entered by the secretary. The actual date is set through the integration with Navision when the payment has been made..	
4. bank account: Not for each payment. Only for the applicant (D6).	Bank account is retrieved from the application.	
5. state: planned, approved, paid.	Yes	
6. application: Relation to D2.	Yes	

D4. RoundState

For each application round, every board member must approve the decisions. RoundState keeps track of this. This point is cancelled. The secretary updates the state of the round as part of the workflow when the board has approved the grants of the round. Mails from the board can with advantage be filed on the round as documentation of this.

Examples: The engineering specialist's approval of grant round 12-04-2014.
Data source: The individual board member.
Data use: Shows whether the secretary can carry out the decisions.

Data volume:	Proposed solution:	Code:
1. There are four for each grant round.		

Fields and relationships:	Proposed solution:	Code:
2. state: None, approved, rejected.	Cf. above, it is planned that the secretary sends minutes to the board members by email and collects the replies. It is also possible that each board member signs electronically.	
3. application round: Relation to D1.	Yes	
4. person_Org: Relation to D6.	Yes	

D5. ApplicationRole

An application role indicates that a grant application relates to a specific person or organization.

Examples: The application relates to the company that applies, a contact person in the company, a board member and the remaining board members.

Data source: The secretary creates application roles, partly automatically. The board members change state for their own application role during the process (voting). An application may change applicant, contact person, etc. during the process.

Data use: The roles keep track of whom to inform on what, and who votes what.

Data volume:	Proposed solution:	Code:
1. Around 6 roles are created per application.		

Fields and relationships:	Proposed solution:	Code:
2. roleType: The type of role in question. See the requirement note below.	The customer can maintain the role types he wants. The roles for domain expert and board member can be renamed, but the functionality related to for instance the "traffic lights" can not be adjusted by the customer.	
3. my_conclusion: None, discuss, grant, reject: This field is used only by board members. Can be seen by the other board members.	Fully supported - "traffic lights"	
4. comment1: The board member's opinion. Visible to other board members.	Yes	
5. amount1: The board member's suggestion for granted amount. Visible to other board members.	Not shown in the demo earlier. Separate field or just written in the comment field?	
6. joint_conclusion: For recording the decision during the board meeting.	The joint conclusion is probably in the application record and not in the individual roles. But the field exists.	
7. comment2: For notes on the decision during the board meeting.	This field should probably be in the application record. But the field exists.	
8. amount2: For recording the actually granted amount during the board meeting.	This field should probably be in the application record. But the field exists.	
9. private_comment: The board member's private comments. Not visible to others.	Yes	
10. private_category: The board member's. Not visible to others.	Is this a text field or a category?	
11. application: Relation to D2.	Yes	
12. person_Org: Relation to D6.	Yes	

Requirement note, RoleType

Includes these role types:

1. applicant: Either a person or an organization.
2. contactPerson.
3. domainExpert: The board member who primarily assesses the application.
4. other_member: Another member of the board.
5. other: Other persons with relation to the application.

D6. Person_Org

Comprises all persons and organizations that the Foundation has a contact with.

Examples: Board members, employees, applicants, contact persons, etc.
Data source: Persons and organizations are created by the secretaries.
Data use: The information is used everywhere in the application process.

Data volume:	Proposed solution:	Code:
1. Around 600 persons and organizations are created a year.	Contacts are related to applications with the fields outlined below. Alt B: With this solution a shared list of contacts is established and connected to applications. Unique key is civil registration number and email address.	

Fields and relationships:	Proposed solution:	Code:
2. name.	Yes	
3. title.	Yes	
4. address.	Yes	
5. zip_code.	Yes	
6. country?	Country codes may be used. The customer can maintain the codes. The on-line application form also contains country codes, but the customer cannot maintain them.	
7. private_address.	Yes	
8. private_zip_code.	Yes	
9. private country?	Yes	
10. telephone_1	Yes	
11. telephone_2	Yes	
12. email_1	Yes	
13. email_2	Yes	
14. CPR (civil registration number), for some persons.	Validated with modulus 11. A warning is given.	
15. CVR (company registration number), for some organizations.	There are few CVR numbers, so the customer estimated that it isn't important to look up the CVR numbers.	
16. birth date (for foreigners).	Yes	
17. expert_area	Engineering or medicine?	
18. bank_account (bank code and account number).	The original number that the applicant provided is always filed. The Foundation can later change the number if needed.	
19. Belongs_to: Relation to an organization (D6) that the person or organization belongs to.	Yes	

D7. Reporting

Each application has one or more planned reports, typically in connection with payments.

Examples: A final report or a quarterly report of activities and results.
Data source: Created by a secretary.
Data use: Used to generate reminders if the report doesn't turn up.

Data volume:	Proposed solution:	Code:
1. Around 400 reportings are created a year, typically one for each payment.	Payments are recorded in the system and later sent to Navision for payment through the bank and registration in the tax system.	

Fields and relationships:	Proposed solution:	Code:
2. report name.	Yes	
3. date: The planned or actual report date.	Yes	
4. state: planned, actual, accepted.	Yes	
5. application: Relation to D2.	Yes	

D8. Document

Documents comprise electronic documents such as emails, Word, pdf, picture files and scanned documents.

Examples:
Data source: Documents arrive electronically or are created by the employees.
Data use: Documens about a specific application, reporting,etc.

Data volume:	Proposed solution:	Code:
1. Around 2000 documents are filed a year.		

Fields and relationships:	Proposed solution:	Code:
2. document name.	The document is filed on the application in an application folder, and named with a meaningful name.	
3. type. See possibilities in the requirements note below.	It is possible to have document types. However, in practice this isn't followed by the users so the value is questionable. We recommend to place key documents in the application record and relate dependencies to this, e.g. final report.	
4. media. See possibilities in the requirement note below.	All types are supported. Mails are filed in Outlook format, msg file. Excel, Powerpoint, etc. can also be filed.	
5. belongs to: Relation to an entity ("file card") in one of the data classes D1 to D7.	Documents are related to a specific application or application round (i.e. business object), but may also be connected to objects in the application record.	

Requirement note, Type
Includes these document types:

1. Application
2. Documentation: That the project can be done.
3. Instalment_report
4. Final_report
5. Other

Requirement note, media
Includes these media types:

1. pdf
2. mail
3. doc
4. docx
5. jpg

D9. Template

Templates can be used as starting points for documents.

Examples: Grant letter, rejection letter, reminder on yearly report.
Data source: A template can be created or changed by a secretary.
Data use:

Data volume:	Proposed solution:	Code:
1. There are around 40 templates today.	Supplier C offers Word and email templates where data from the objects (e.g. the application) are inserted automatically. In this way it is easy to create new documents in Word or email. As part of the project, supplier C teaches the customer how to make these templates, after which it is the customer's responsibility to create the templates.	

Fields and relationships:	Proposed solution:	Code:
2. template_name.	Yes	
3. text.	Yes	
4. automatic_fields (filled in automatically).	Supplier C uses Microsoft Office QuickParts to insert data from applicant and application directly in the document. During the project it is agreed exactly which QuickParts are needed.	

E. Other functional requirements

Most system functions are simple creations, deletions, edits, and queries that need no further specification. They are implicitly given by the task descriptions (Chapter B4), system integrations (Chapter F) and data descriptions (Chapter D). In addition, the system must be able to perform the functions specified in this chapter.

E1. System generated events

The system must handle these events:	Proposed solution:	Code:
1. When an application arrives through the web-site: Create an application. Send a message to the secretary.	Offered.	
2. If a board member doesn't reply on time: Send a reminder to the secretary and/or the board member.	As described in the tasks.	
3. If a final report, instalment report or additional documentation doesn't arrive on time: Send a reminder to the secretary and/or the applicant.	As described in the tasks.	

E2. Overview and reports

There is a need for overview and reports. Some of them are already suggested as solutions in the tasks, e.g. C20-1. F8 requires transfer of data to Excel. Some reports can be made this way.

Report requirements:	Proposed solution:	Code:
1. A report is needed to support subtask C20-8. The format isn't important as long as the subtask is supported efficiently.	Offered. See C20-8.	
2. The system contains a report generator that can extract and combine all data described in Chapter D.	Structured data can be transferred to Excel, not documents, e.g. in pdf.	
3. The report generator can be used by a typical secretary.	The secretary transfers data to Excel and generates reports there.	
4. The system must be able to show all reports on the screen as well as on print.	Excel can do it.	

E3. Business rules and complex calculations

None.

E4. System administration

System administration comprises the following subtasks.

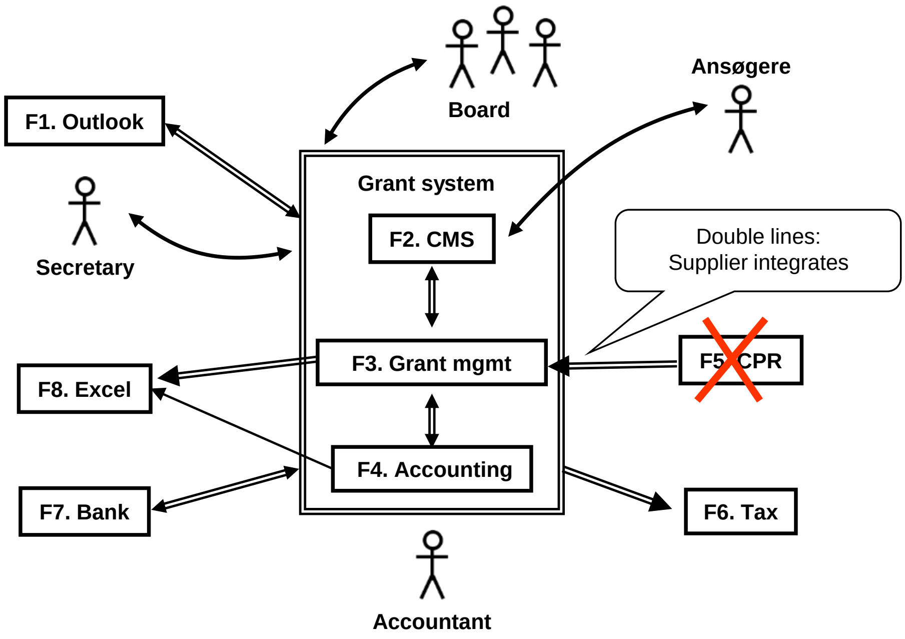
Subtasks in system administration:	Proposed solution:	Code:
1. Set the length of time monitors, e.g. the number of days from a missing report to a reminder.	The supplier helps initially. The customer can do it himself.	
2. Specify which data changes will stop which time monitors.	The supplier helps initially. The customer can do it himself.	
3. Specify which actions will start which time monitors.	The supplier helps initially. The customer can do it himself.	
4. Specify which secretaries can replace whom.	The supplier helps initially. The customer can do it himself.	
6. Create, modify and delete role types, categories, etc.	The supplier helps initially. The customer can do it himself.	
7. See and search in the log of changes.	Can be done by system administrator and to some intent by the secretary.	

Other changes and extensions of the system are covered by the maintenance agreement (L5). Migration to another system is covered by data conversion (J4).

F. Integration with external systems

The grant system must integrate more or less closely with the external systems shown in Figure F (context diagram). The integration consists mainly of data transfers.

Figure F: Integration with external systems



Integrations to be supported:	Proposed solution:	Code:
F1. The customer's email system. All mails received or sent concerning applications must be filed by the grant system.	When a mail is received, it can be attached to the application with drag-and-drop or by retrieving the application and attaching it.	
F2. The customer's web-site. It must be possible to receive applications through the customer's web-site (C41).	See C41, etc. With Alt. A supplementary documentation can also be uploaded.	
F4. Accounting system. Grants and payments must be transferred to the accounting system.	See C14, etc.	
F5. CPR-register (civil registration). The grant system should be able to validate CPR-numbers, e.g. to succeed with tax registration. As a minimum, validation must take place at the tax registration.	No integration is offered, only modulus-11 check.	
F6. Tax. Reporting to Tax of the paid grants.	Done through the accounting system.	
F7. Bank. Transfer of pay orders to the bank.	Done through the accounting system.	
F8. Transfer of data to Excel spreadsheets. All data in Chapter D must be transferable.	Offered, see E2-3.	
F9. CVR-register (company registration). Validation of CVR-numbers.	This integration is not included.	

G. Technical IT architecture

G1. The supplier or a third party operates the system, etc.

The customer will make sure that not only the supplier can perform operation, support and maintenance. Among other things, this requires that a widely used technology is used. See also section J6.

Section A3 describes the parts that are offered.

Platform requirements:	Proposed solution:	Code:
1. The supplier or a third party selected by him, operates the system and uses the necessary equipment to meet L1, L2 and L3.	The system is operated by three specialized operators, one for Flowsystem C based on SharePoint, one for Navision og one for Wordpress.	
2. The system uses a widely used technology in order that others can operate the system, provide support and maintain it. This is valid for the grant management system itself as well as the CMS system.	All the systems are widely used.	
3. Board members must be able to see and assess grant applications (C20 og C21). It is desirable that it can take place also from mobile devices, e.g. iPhone. It is not sufficient to be able to download all grant applications, because it is necessary to see other's assessment - also during board meetings.	The access is browser-based. Display of documents assumes that the specific device (e.g. iPhone) has a viewer for showing the document. Supplier C cannot take responsibility for that these viewers work correctly in all cases.	
4. The customer's own equipment.	The customer must acquire PC's and internet access in order to use the system and establish VPN connections.	

H. Security

H1. Login and access rights for users

Login is not a separate user task, but subtasks that occur in many tasks. The system must support the following subtasks relating to the user's access rights.

User roles
Applicant
Board member
Secretary
System administrator

Subtasks for user access:	Proposed solution:	Code:
1. Identify the user.(See section H5-3 about the length of passwords.)	Microsoft's AD solution, also when the system is used through the web browser.	
2. The user has been away from the system for some time. Avoid that another user accesses the system with the rights of the first user.	Microsoft's solution.	
2p. Problem: If the system logs out automatically, it is cumbersome to log on again.	Microsoft's solution.	
2q. Problem: If the system logs out automatically, entered data may be lost.	Microsoft's solution. For Wordpress we use the standard model for this product.	
3. Check that only authorized users get access to system and data. (See the requirement note below.)	The supplier configures the system in such a way that the requirement note below is handled. In principle all 6 roles can be created, but in practice it is complex to handle that many roles. Instead we recommend to use two roles, user and administrator, corresponding to the two primary roles in SharePoint. This minimizes administration.	
3p. Avoid that users have a password for each system.	Microsoft's solution. For Wordpress and Navision there are separate passwords. If necessary, Navision can use Microsoft AD as user source.	

Requirement note: Possible access rights

1. Right to see all application data in Chapter D, print reports and see the log of changes (secretaries).
2. Right to create and update application data in Chapter D (with logging), (secretaries).
3. Right to see applications, other's comments and note own comments (board members).
4. Right to create and update own applications (applicants). Offered as part of Alt. A.
5. Right to configure the system (see subtasks in E4) and assign rights (see H2), (system administrator). Initially the secretary is also system administrator.
6. Right to see applications and comments for publish-worthy applications. (The web-editor has the same rights as a board member).

An employee may have several rights. As an example, a specific secretary may have rights 1, 2, 3 and 5.

H2. Security management

The work in security management includes the following subtasks. Security management is based on Microsoft Active Directory.

Subtasks for security management:	Proposed solution:	Code:
1. Assign and remove rights for a user.	Microsoft's solution.	
1a. First create the user.	Microsoft's solution.	
1p. It happens that users share passwords in order to easier replace each other. Should be avoided.	Microsoft's solution.	
2. Get an overview of who has which rights and whether some rights have not been assigned to anyone.	Microsoft's solution.	

H3. Protection against data loss

Data may unintentionally be lost or misinterpreted in many ways.

The system must protect against:	Proposed solution:	Code:
1. Local concurrency problems, for instance that user A changes some application data while user B also changes it.	Microsoft's solution. When several board members update their opinion at the same time, some updates may be lost. Since a board member can only change his own opinion, there is no conflict here. At other updates, the first one will win. The following update of "old data" will not take place and the user will be warned that data has been changed since he saw the old value.	
2. Disk crash, fire, theft, etc.	The operator will ensure this.	

H4. Protection against unintended user actions

An unintended user action means that the user happened to do something he didn't intend to do, e.g. hitting the wrong key or using a command that does something he didn't expect.

Requirements:	Proposed solution:	Code:
1. Unintended user actions may not cause the system to close down, neither on the client nor on the server.	Microsoft's solution.	
2. All data entered must be checked for format, consistency and validity. In case of doubt, the user must be warned and asked what to do.	The supplier provides reasonable checks.	
3. The user must be able to correct mistakes easily.	Microsoft's solution.	
4. The user must be able to interrupt functions that take a long time, e.g. a long data transfer, without compromising data integrity.	There are no large data transfers or reports in the system. We estimate that it isn't a problem.	

H5. Protection against threats

The customer has only identified a few threats. The supplier is expected to know and protect against others.

Protection against threats:	Proposed solution:	Code:
1. The supplier is expected to list the threats that are most significant for this kind of system and specify the protection he recommends.	Microsoft's solution.	
2. The system must protect against operator staff transferring money from the customer's bank.	Handled by the bank, which requires NemID (Danish electronic signature). Navision uses the solution in its payment module.	
3. The system must protect against operator staff getting access to account and application data.	Cannot be enforced because the operator wouldn't be able to provide support. But the staff has professional secrecy and is certified.	
4. The system must ensure that the user only has administrator rights from the customer's office.	These functions are behind a firewall and require VPN access. This applies to the board as well.	
5. The system must prevent that someone forges the bank account number prior to the payment.	The system can in the payment list show what originates directly from the applicant and what has been changed (and logged).	

I. Usability and design

I1. Ease-of-learning and task efficiency

The customer will early in the project make sure that the grant system has the necessary usability and efficiency for users and applicants. See section B3.

Requirements:	Proposed solution:	Code:
1. The secretary must be able to carry out the tasks in Work Area 1 without serious usability problems (see definition below).	With a functional version of the system, a secretary carries out examples of tasks alone. On the way, the secretary may ask the supplier's expert. The secretary assesses whether the system is sufficiently efficient and easy to use. Offered.	
2. Board members must be able to carry out the tasks in Work Area 2 without serious usability problems.	With a functional version of the system, two board members carry out examples of their tasks alone. On the way, they may ask the supplier's expert. The board members assess whether the system is sufficiently efficient and easy to use. Offered.	
3. Potential applicants must be able to carry out the tasks in Work Area 4 without serious usability problems.	A think-aloud test with three potential applicants is made. In a think-aloud test, the user cannot ask when in doubt. See the requirement note below. This is the customer's own responsibility.	
4. If the customer finds that the system isn't sufficiently efficient and easy to use, the supplier can choose to improve the system and try again, or cancel the contract.	~~On the condition that the supplier has made a good professional effort, the supplier can get a compensation of __?___ DKK.~~ The rules for cancelling the contract are stated in App. 1D, proof of concept.	

Requirement note: Serious usability problems

A **serious** usability problem is a situation where the user:

a. is unable to complete the task on his own,

b. or believes it is completed when it is not,

c. or complains that *it is really cumbersome*,

d. or the test facilitator observes that the user doesn't use the system efficiently.

The customer will primarily assess the usability based on the observed serious usability problems.

Requirement note: Test tasks

A good test task corresponds to something a real user would have to do. It must be presented in such a way that it doesn't guide the user. Here is a good and a bad example:

Test task 1 (good): Apply at the Y-Foundation: You are interested in publishing a book. Does the Y-Foundation support this? How do you apply? How will the application be handled? When can you apply? Apply for a grant for the book.

Test task 2 (bad - guides the user): Apply at the Y-Foundation: Go to Yfoundation.dk. Select APPLY FOR A GRANT in the topmost line …

I2. Accessibility and Look-and-Feel

Requirement:	Proposed solution:	Code:
1. The user interface must be in Danish.	Offered. We recommend that the administrator part in SharePoint is in English. The same applies for Wordpress.	

J. Other requirements and deliverables

J1. Other standards to obey

Requirements:	Proposed solution:	Code:
1. The system must follow good accounting practice. The supplier must obtain the necessary auditor approval.	We assume that Navision suffices.	
2. The system must meet the laws on handling of personal data.	The supplier states that there is the necessary protection because all confidential information, e.g. CPR numbers, are password protected.	

J2. User training

The customer wants to deliver a large part of the training himself. The idea is to train a secretary (super user) first, who then can train others.

Requirements:	Proposed solution:	Code:
1. The supplier must train a secretary, who can train and guide other employees, board members and applicants. The training must enable the secretary to carry out all tasks in Chapter C with exception of area 3 (web-editing).	The training takes place while the secretary participates in the development. The secretary also gets the basic knowledge for configuration of the work flow (see E4).	
2. The training must be done as part of the deployment.	Takes place before the deployment.	

J3. Documentation

The customer expects that users can use the system without documentation.

J4. Data conversion

The supplier must convert the following data from the existing systems:	Proposed solution:	Code:
1. Transfer accounting data from C5.	Done by the supplier of the new accounting system.	
2. Transfer applicant data, application ID, application name and granted amounts from Excel.	Offered.	

J5. Installation

Installation takes place at the operator's site.

Requirements:	Proposed solution:	Code:
1. The supplier must install all parts of the delivery.	Offered.	
2. The supplier must also install the converted data.	Offered.	
3. The supplier must assist the customer with the deployment.	Offered.	

J6. Phasing out

In this section "customer" means the customer's own staff or third party authorized by the customer.

Requirements:	Proposed solution:	Code:
1. On request, the supplier must extract all data described in Chapter D and all accounting data in a format that is suited for import in other systems.	The entire database can be extracted or copied.	
2. The customer must be able to extract all data described in Chapter D and all accounting data in a format that is suited for import in other systems.	Offered.	
3. The supplier must loyally assist with phasing out the system and transferring it to another supplier.	Offered.	
4. The supplier must carry out the work at a fair price that covers time and material.	Offered.	

K. The customer's deliverables

The following list of the customer's deliverables and services must be complete. The supplier cannot expect more from the customer. If necessary, the supplier must add to the list in his proposal.

The customer delivers:	Proposed solution:	Code:
1. An IT work desk.	Available at both parties at demand. It is assumed that the customer has VPN access when AD access is needed.	N/A
2. Production data for testing and conversion.		N/A
3. Domain expertise corresponding to 5 hours a week during the entire project period.		N/A
4. Test persons for usability tests.		N/A
5. A secretary for training.		N/A
6. Expertise for validating converted data.		N/A

L. Operation, support, and maintenance

This chapter specifies the supplier's responsibilities after delivery of the system itself. The requirements can only partly be verified (tested) at the deployment. The full verification takes place later, at the operational test. The requirements must be met by the supplier and the operators in combination.

Solution note
Everything in Chapter L is offered. However, the details are not available yet.

L1. Response times

It is important that response is so fast that users are not delayed. Response time is particularly important during the busiest hours, the **peak load** period, which is the week close to an application deadline. Here uploads of 50 on-line applications per hour are expected.

When the system is operating, it must be able handle the number of transactions specified below with the specified response time. The figures are estimated from task frequency (Chapter C), data volumes (Chapter D) and statistics from the present operation about peak load periods. The figures are the **nominal** load, i.e. the supplier is not responsible for response time if the actual load exceeds the nominal load.

Nominal load

1. Filing of mail: 10 per minute.
2. Show overview of an application: 2 per minute.
3. Update an application: 2 per minute.
4. The public web part: 10 pages retrieved per minute.
5. Upload of applications: 2 applications of 10 MB uploaded per minute.
6. Board meeting: 5 users look at the applications and take notes.

Solution note: Measuring response time
The response time is the period from the user sends his command to the result is visible and the user can send a new command. A command means a key press or a mouse click. All measurements are made in peak load periods with the actual number of users, assuming that the actual load is within the nominal load above.

Production work: Measurements are made on the operational system.

The public web part: Measurements are made on a PC connected to the Internet through a 10/2 MB connection (10Mb down, 2 Mb up) with low traffic on the route to the servers, but with peak load of the servers themselves.

Response time requirements:	Proposed solution:	Code:
1. **Fractile.** The times specified below must apply in almost all cases.	In any one-hour period, _98_% of the response times must be within the limits. (The customer expects 98%.)	
2. Response time measurements must be made at demand when the customer finds the system too slow.	Measurements are made with a stop watch. Or: The system measures all the time.	
3. When moving from one field to the next, the user's typing speed must not be slowed down.	Typing is possible within _0.2 s. (The customer expects 0.2 s.) On the same screen/form.	
4. When moving from one screen to the next, data must be visible and typing possible within the mental switching time (around 1.3 s).	Data is visible and typing possible within _2_ s. (The customer expects 1.3 s.) This depends on the complexity and design. Switching time must be short, but with e.g. change from one grant application to another, a longer switching time doesn't necessarily cause problems. Suppler C usually divides transactions in four categories depending on their complexity: Simple, Normal, High, Advanced. Response times are typically 2, 5, 7 and 10 seconds. We expect that most of the customer's transactions are simple or normal.	
5. Lookup in drop-down lists must allow selection from the list within the mental switching time.	Selection is possible within _0.2_ s. (The customer expects 1.3 s.) As for #3 above.	
6. Reports used frequently must be visible within the mental switching time.	The report must be visible within ___ s. (The customer expects 1.3 s.) The response time depends on the amount of data retrieved and the computation complexity. The list with traffic lights outlined above, has a response time around 1-2 seconds.	
7. Reports used occasionally must be visible before the user loses patience.	The report must be visible within _20_ s. (The customer expects 20 s.) Complex reports may take longer, but we don't expect that the customer has any of these.	
8. Login must be completed before the user loses patience.	The user can start working within _10_ s. in addition to the time he spends typing name and password. (The customer expects 10 s.)	
9. Repeated login when the user temporarily has left the system must be completed before the user loses patience.	The user can start working within _4_ s. in addition to the time he spends typing his identification. (The customer expects 4 s.)	

L2. Availability

The system is out of operation when it doesn't support some of the users as usual. The cause of the breakdown may be:

1. The customer's issues, e.g. errors in the customer's equipment.
2. External errors, e.g. power failure.
3. The supplier's issues, e.g. errors in software or configuration.
4. Planned maintenance.
5. Insufficient hardware capacity.

Solution note: Measuring availability
The supplier states that the combined availability **isn't** the product of the availability of the three operators (e.g. 98% * 98% * 98% = 94%). This is because the systems usually are used one at a time.

A breakdown is counted as at least 20 minutes, even if normal operation is resumed before. If the following period of normal operation is less than 60 minutes, it is considered part of the breakdown period.

The **operational time** in a period is calculated as the total length of the period minus the total length of the breakdowns for which the supplier is responsible. The **availability** is calculated as the operational time divided by the total length of the period. When only some of the users experience a breakdown, the availability may be adjusted. One way is to calculate the availability for each user and take the average for all users.

Availability requirements:	Proposed solution:	Code:
1. The availability must be calculated periodically. The calculation should compensate for the number of users experiencing breakdowns.	The availability is stated monthly and calculated as described above.	
2. In the period from 8:00 to 18:00 on weekdays, the system must have high availability.	In these periods the total availability is at least ___%. (The customer expects 99.5%, i.e. the system may be down 13 hours per year in these periods). Navision: 99% (8:00-21:00 weekdays, 09:00-18:00 sat+sunday) Wordpress: 99% Flowsystem C: 98%	
3. In other periods the availability may be lower.	In these periods availability is at least _98 %. (The customer expects 99%). Planned service windows are not considered downtime and thus don't influence availability. In this connection, the supplier will inform the customer and take the customer's demand into consideration.	

L3. Data storage

The data volume is specified in Chapter D. Data must be stored as follows:

Data storage requirements:	Proposed solution:	Code:
1. The system must give access to data for the last 5 years with the response times specified in L1.	Yes	
2. The system must give access to archived data for the last 20 years with response times as for occasionally used reports (L17).	No problem, even with large amounts of pictures.	

L4. Support

Support comprises help to users, configuration changes, and monitoring of the operation. The supplier provides support and involves the three operators as needed. A "supporter" means a qualified employee from the supplier or the operator.

The customer's local IT supporter is the user's first point of contact for assistance. Thus the supporter only has to assist when the local IT supporter cannot remedy the problem. The supplier may communicate with the local IT supporter or directly with the user.

Requirement note: Handle a request

When a supporter receives a request, he can perform one or more of the following subtasks. All subtasks except e (escalation) end with a **reply** to the user. The request is **closed** when nothing more can be done about the request (subtask f).

a. Help user: Assist the user in solving the problem or circumventing it. If needed contact the user for clarification. Assistance is considered a valid reply.

b. Change configuration: E.g. start servers, change settings, replace printer cartridges, install software. Reply to the user when it has been done.

c. Order equipment or help from another organization: Reply to the user about the expected delay.

d. Defect: The support organization cannot solve the problem. Report it to the maintenance organization. Reply to the user that it has been done.

e. Escalate request: The supporter cannot fully solve the problem himself. Pass the request on to another supporter. This person may again perform one or more of the subtasks.

f. Close the request: Nothing more can be done about the request. This may happen at the first point of contact. The request may also escalate several times, wait for external delivery or wait for a reply from maintenance before it can be closed. Reply to the user that the request has been closed.

Support requirements:	Proposed solution:	Code:
1. **Fractile.** The response times specified below must apply in almost all cases.	_95_% of the response times must be within the limits. (The customer expects 95%.)	
2. The supplier must handle user requests for help. Help comprises software and operation under this contract. See the requirement note above.		
2p. Problem: The customer finds it hard to decide which product a specific problem relates to. It is even harder to mediate between several suppliers.	The supplier involves the necessary other parties (accounting and web site) on his own initiative. There is no support on internal IT, PC's, network, printers, etc.	
3. Direct contact: In the period 9:00 to 17:00 on weekdays, users can quickly contact a supporter by phone or in person.	In this period, contact is available within _10_ minutes. (The customer expects 10 minutes.)	
4. For a direct contact, the supporter handles the request on the spot as far as possible.	*On the spot* means what can be done within 5 minutes.	
5. Indirect contact: Requests sent by email, sent by web, or escalated from the direct contact. The user gets a reply within a few hours.	The supplier replies within __ work hours (9:00 to 17:00 on weekdays). (The customer expects 3 hours.) This depends on the criticality of the case. See the appendix "subscription and support" for response times.	
6. The supplier monitors request handling to see that requests are closed and response times met.	Yes.	
7. The supplier records data for computation of support response time, and identification and prevention of frequent problems.	The supplier keeps a log of all steps in the request handling and the cause of the problem.	
8. The supplier monitors the operation in order to foresee availability problems, and changes the technical configuration so that availability is maintained.	Yes.	

L5. Maintenance

Maintenance includes defect removal, system updates and system changes.

Requirements for defect removal:	Proposed solution:	Code:
1. **Fractile.** The response times specified below must apply in almost all cases.	_95_% of the response times must be within the limits. (The customer expects 95%.)	
2. The supplier keeps a log of reported defects as well as change requests.	Yes	
3. For a reported defect that the customer considers business critical, the supplier quickly decides whether it is possible to circumvent it temporarily, or possible to circumvent it permanently (i.e. reject it) .	In the period from 9:00 to 19:00 on weekdays, the supplier completes the assessment within _3_ hours. (The customer expects 3 hours.) Only 9:00 to 17:00 for Flowsystem C.	
4. Business-critical defects that cannot be circumvented are removed quickly.	They are removed within __hours. (The customer expects 24 hours.) The work starts within 4 hours from the request. We cannot guarantee when the problem is remedied, but the work continues until a solution is found. The customer is expected to participate in the problem analysis.	

Requirements for system improvement:	Proposed solution:	Code:
5. The supplier installs new versions and releases of the delivered software without unduly delay.	New versions and releases are installed continuously in service windows. Planned downtime for this purpose isn't included in the service goals and are agreed with the customer well in advance. Service windows will primarily be evening/night and weekends.	
6. Within a period of 3 years, the supplier offers changes at a fair hourly rate.	Follows Supplier C's standard contract that is offered to all customers.	

M. Examples of existing documents

(Not included in the anonymized version)