# Content

## 1. Authors

Claes Martinsen, clae@itu.dk
Niels Martin Søholm Jensen, nmar@itu.dk
Jan Aagaard Meier, jmei@itu.dk

## 2. Abstract

Our system is based on how voters are currently being registered at polling stations during elections. Today all voters are registered in books at the polling stations. When a voter want to vote he hands in his voter card and his name is looked up in one of these books.The name is marked in the book, and he is handed a ballot. It is not possible to register twice, unless the voter is first unregistered.

The goal of our system is to make the current paper-based process (looking into a paper based book) into a digital process where the system will handle the task of registering a voter using his voting card at a polling station.

When a voter has been authenticated, registered and has received his ballot, the job of the system has been completed. We wish to empthasize that the system does NOT have anything to do with registering which party a voter has voted for (e-voting), the goal is to ensure that a voter is unable to vote more than once at a given election.

## 3. Requirements

### 3.1 Mandatory Requirements

The system must do the following:

1. Ensure that each voter only receives one personal voter card.
2. Ensure that each voter can only be handed one ballot at the polling station.
3. The system must implement a protocol that ensures secure (distributed message passing between processes.)
4. The voter must be able to register at any table at a polling station.
5. The system must be able to receive data from all polling tables at a given polling station at any time during the election day.
6. The system must ensure that registering a voter takes, at worst, as long as it used to in the original solution[1].
7. The system must be able to, centrally, generate voter cards for all voters in Denmark.
8. The system must show in a clear fashion when a voter has successfully been registered

### 3.2 Comments on mandatory requirements

1. We have decided to lighten this requirement, so that the system does not enforce that only one polling card is generated, since the numbers on the voter cards become more

[1] Original registration solution: Looking up the voter in a book containing all expected voters at the given voting place.

of an identification aid rather than being a ‘token’ which grants the ability to vote once. Preventing voter cards from being generated more than once causes more trouble than it does good. See also section 6.2.5.

2. The process of registering a voter and handing out the ballots is described in 6.2.4. Polling workers are instructed in how to operate the system. If any situation should arise where the registration process halts or the poll worker has made a mistake, an official must be called upon to resolve any issues and provide administrator password to possibly unregister the voter.

3. Our early assumption was that using a closed local network would in itself qualify as a secure connection. Domain insight obtained towards the end of the project period indicate that this was a too optimistic assumption. The DBMS used by our program supports encrypted message passing but it is not enabled for convenience and our privileges with regards to servers at ITU are also limited. See [Security Reflections, groupCJN, nmar / All, 2011-12-11] for more details.

4. All polling tables are connected to the same database, so all tables can check the validity of a voting card. See section 6.2.4.

5. With 20 polling table clients at one station, and polling workers scanning as fast as they can, approx one worker scan each 5 second there will be an average rate of database updates of one update per 250ms. This is not at all unreasonable for accessing tables of the size the system is working with, perhaps up to 10.000 voters per polling station, over a local network.

6. If there are no issues with registering the voter (e.g. the voter has not already been registered, and all work can be handled by the polling worker without help from and official) the registration only takes 3 operations: scanning the card / inputting the card number, finding the voter and marking him as registered. Compared with looking him up in a book, perhaps turning several pages and double-checking that you are checking of the right person we feel confident that our system fulfills this requirement.

7. Possible, but not very likely compared to what is being done in reality. A more likely scenario is where voter cards are generated at individual municipalities. Both scenarios are supported in our system, and if voter cards are already generated, e.g. if they have been generated centrally and a municipality then tries to generate them again the system warns the user but does not prohibit re-generation. See also section 6.2.5.

8. The system shows a dialog box, when a voter has been registered. The polling worker is not able to continue scanning until he has accepted this dialog message.

### 3.3 Secondary Requirements

The system may do the following:

1. The system may be able to distribute voters across polling stations automatically based upon some known location data (such as school district or parish).
2. The system may be able to visualize the current voting percentage at the specific voting station.
3. The system may be able to show how many voters was registered at a given polling station during a specific time interval (possibly visualize using diagrams).
4. A voter may be able to vote at any given polling station in Denmark (Real-time nationwide authentication of every voter registration).

### 3.4 Comments on secondary requirements

1. We do not have any data on school districts or parishes available, so we have decided not to look further into this requirement. Given the right data and our current DAO setup it would proberly just be a matter of formulating the right LINQ queries.

2. The log in the polling station window shows the number of voters who are currently registered as having voted.

3. The log window was primarily implemented for debug purposes, and for determining what has happened in the case that a voter tries to authenticate with a card that has already been used. Technically it can also give an overview over how many voters have been registered at a specific time-interval, but it does not give a very good overview, only a list over the log-operations. We have decided not to work further on it, since this is not essential to carrying out the actual election but more for reflection upon the course of the election after it has happened.

4. This would have been very nice to achieve, but we have discovered, that there are plenty of difficulties in securing a system that is not distributed. For further reflections on this topic see the [Security Reflections, groupCJN,  nmar / All, 2011-12-11] document.

## 4. Overview

Today voting registration and ballot hand-out is carried out solely by hand. Voter cards are created and printed digitally, but all handling of voter cards on the election day happens by hand, by looking up voters in these books. This means that, at least in many places, voters are assigned one specific table where they have to queue up to be registered, even though other tables may be free.

The books also introduce the possibility of human error, given that the polling worker might cross of the wrong voter in the books. With this being a possibility, polling workers might also be more inclined the let a person vote even though he has already been registered once, given that they registration *might* stem from a human error, and there is no way to determine when the user was first registered, or by whom.

In a digital system, it is possible to overcome these limitations. The poll worker enters a number or scans a bar-code that uniquely identifies the voter (most likely based on CPR number) and the system then ensures, that that exact voter is marked as having voted. Furthermore, the system also logs all access, making it possible to check exactly when a voter was looked up, and when he was registered.

When the poll worker scans a voter card the system will respond with a dialog, allowing him to compare the data on the screen with the data on the voting card, in order to confirm that he is registering the right voter. The system still support manual input of CPR numbers but it should only be for special cases. Bar-codes eliminate the issue of accidentally entering the wrong voter data completely, since CPR numbers are entered into the program automatically.

Our system consists of individual local distributed systems. All polling stations must have a physical server (voter box) with a number of clients (polling station clients) communicating with the server.The system is not communicating in any way over the Internet. Another part of the system is able to generate voter cards centrally and produce expected voter lists, these are then to be fed into the above mentioned voter boxes to be placed at the polling stations. Further system analysis can be found in [System Architecture, groupCJN, nmar / All, 2011-12-11] and system design can be found in the .bon files.

The team consists of Niels (Central module and architecture) , Jan (database module) and Claes (Polling Table).

The roles and responsibility have overlapped somewhat. The comments in the code states who have written the actual code.

## 5. Dictionary

Your dictionary of classifiers, as generated from your analysis phase, and updated during your design phase.  Your first draft submitted for review need not contain anything here.

### 5.1 Domain Terms

- **Polling station** - one of the physical polling locations (buildings) throughout the country. A polling station contains one or more polling tables.
- **Polling table** - the table where a voter card is shown and a ballot is received. Used in our software as the name of the client which is used at the polling table.
- **Voter card / polling card** - a voter card is personal to the voter. The voter receives his voter card prior to the election.
- **Ballot** - a ballot is anonymous, and is handed to the voter at the voting place after he has been autheticated based on his voter card (or through other methods?)
- **CPR register** - a database containing the names and cpr. number of all citizens in Denmark.
- **Folkeregisteret** - a database containing all current and former address of all citizens in Denmark.
- **Poll worker** - (Danish: Valgtilforordnet) A local volunteer who exchanges voter cards for ballots at a polling table.

- **Polling official / official** - the person at a polling station who makes sure that everything works correctly. He is the one with the password to startup polling station clients and unlock voters.
- **Voter box** - a server located at every polling station that stores a database of voters registered at the polling station. (When used for postal votes it can be located at a municipality)
- **Voter box client** - The client station (e.g. laptop) where a poll worker registers votercards

### 5.2 Technical Terms:

- **DAO** - Data access object
- **DO** - Data object

## Assumptions:

- We assume that the voters are distributed, using the central module, over a closed Ethernet network where somewhat skilled personal operate the central module and voter boxes as well as the actual setup of polling tables clients and voter boxes at the polling stations.
- The voter box and polling tables at the polling stations are under strict supervision (just as the ballots etc. are). Officials are assigned to keep intruders from tampering with the Ethernet cables.
- One or many polling tables are used for registering postal votes before the election at a given location in each municipality (usually the town hall or the like). Skilled personal then distribute the voters using the central application so each polling station receives a voter box fully prepared with all voters assigned to that polling station.
- All voter cards are generated (and printed) centrally by skilled personel.
- The system is not involved in the distribution of the voter cards via mail, only in the generation of these cards.

## 6. Example

### 6.1 Technical example (setup)

The system works on a database called groupcjn. Limitations in the LINQ to entity mapping does not make it possible to change the name of the database, neither the names of any of the tables.

A database for use with the project is provided at the following adress: mysql.itu.dk, user: groupCJN, pass: abc123, dbname: groupcjn. A web interface can be found via http://itu.dk/mysql

In case the above mentioned database at ITU proves unavailable we have provided a generator at `DBComm.DBComm.Program` which will create a new database on localhost, with the user id groupCJN and password abc123. By default this database is populated with 10 municipalities, 100 polling stations and 500 voters.
The system requires at least the following privileges:

- Central: create databases, drop databases, create tables, create stored procedures, create indices, insert

- Polling table: select, update, call stored procedures

To run the application of your own server, the program requires database access with user: groupCJN and password: abc123, and the above mentioned privileges.

The program is not provided as a fully compiled and ready to run .exe file. This is due to some issues with the MySQL connector, which uses sockets, and the default .NET client profile, which does not allow sockets. It is possible to fix these issues, by manually editing some text files, but we chose not to require the reader to go through these step, and have instead just provided easy to use projects, which just require the user to select them as startup projects in Visual Studio and run them.

The connection credentials should be filled in by default for use with the above mentioned MySQL database at ITU. The only thing required is the before mentioned password 'abc123'.

## 6.2 Use of the program through various phases of the election.

The application supports multiple processes of the voter card process:

- Distribution of voters from a central database into several smaller databases.
- The generation of voter cards in batches, bounded either by municipality, polling station, or simply by the number of voters in each batch.
- The registration of whether or not a voter has voted at the actual day of the election.
- Registration of postal votes.

In the following paragraphs the steps that follow the calling of an election.

### 6.2.1 Distribution of voters from the central database into smaller databases

A diagram of the distribution process, as explained below, can be found in the diagram attachments [Diagram of voter distribution and network, groupCJN, clae / All, 2011-12-04]
The set of all eligible voters is extracted from a central database outside the system, probably the CPR database. This process happens centrally, most likely at the ministry, where people who have been thoroughly educated in the use of the system handles its use. From here, there are two options – the system may distribute the full voter database onto several smaller databases, while making sure that each voter is only present in one of the new instance, or batches or voter cards may be printed centrally. The printing of voters is described in a later paragraph, and works the same way, regardless of whether it happens at the central, or in a municipality.

The first distribution of voters splits the full database into databases only containing entries from one municipality. The system lets the user set up a filter, describing a specific subset of voters and then transfer this subset to a specified server in the local network. In the current version of the system the transfer of data happens via a closed local network where the voters are transferred to the municipality-databases, called voter-boxes. The voter-boxes are then distributed to the correct municipalities. Since the actual transfer of data between the central server and the voter-boxes happens over a closed local network, with no connection to the Internet, there is no chance of the data being intercepted / altered during this process. Furthermore the system lets the user validate that the correct data-set is actually present at the current voter-box.

Another means of transferring the data could have been to produce a set of encrypted USB-keys, each with a dump of the data that would otherwise have been transferred directly to a voter-box. These keys are then distributed to the right municipalities, each of which is then responsible for putting the data into their own databases. This provides for a more flexible system, where the ministry does not have to manage and set up the full server-park, and each municipality can just use some computers that they have lying around to store the data. We have learned, that this second solution is closer to what is actually happening in alternative systems, but we have chosen to keep our implementation of the distribution more central since we feel that it provides a greater degree of control and security that all servers are setup and initialized correctly, and that no voter is lost in the progress.

#### 6.2.2 Registration of postal votes

In paper-based systems, persons who cannot be present at the day of the election have the ability to give their vote at a municipality office within a set time before the election. We wish to also support this functionality in our system. When the municipality has received their database from the central they are able to connect a polling station client to this server, in exactly the same way as they would do on the election day, and register the voter via the same user interface.

#### 6.2.3 Further distribution of voters

After the deadline for postal votes has passed, but with plenty of time before the election the municipality is responsible for distributing the database from their voter-box (which now acts as a new central server) unto several smaller voter-boxes (each representing a polling station). This process happens in the same way as the first distribution from the central server, and the full status of a voter is copied unto the voter box, such that persons who have already submitted a postal vote are not able to vote again.

#### 6.2.4 Registration of voters on the day of the election

The polling station boxes set up by the municipality are transported to the polling stations where they are connected to one or more polling station clients. The clients provide a simple user interface, which by default only supports two actions, looking up whether or not a voter has cast his vote and, if no vote has been registered, register the voter. The task of registering the voter is carried out by volunteering polling workers, who have no special training in the use of the system, other than perhaps a small walk-through introducing them to the registration procedure and formalities of voter registration.

At each polling station there will also be at least one person present, who has been more thoroughly trained in the use of the system. This person will ideally be capable in the use of IT systems. This person will be equipped with a special password, which grants him access to further functionalities in the system. He is responsible for starting all the polling station clients, and his password is also the password used the access the database. These functionalities are built into all polling station clients, allowing him to assist at a specific computer when something goes wrong, but can only be accessed via his password. These ‘administrator’ functionalities are the ability to unregister a voter, and to access a log, showing all database accesses. This will help him decide which action to take, should one of the failure scenarios described in [Polling Use Case, groupCJN, All members, 2011-12-04] occur.

The process of registering a voter, from he arrives at the polling table until he leaves, with or without a ballot is depicted in the diagram in [Voter card registration process, groupCJN, clae, ver.1.2]. All polling stations are connected to the same voterbox at the polling station, so the voter may approach any table he wishes.The polling worker scans the voters card, and the code on the card (a hash-value of the voter's CPR number is entered into the program). He then clicks the button to find a voter and a new window is shown, if the voter is found in the database. If the voter is not registered as having voted the polling worker may click a button to register the user, or cancel the operation. If the voter is already the polling worker will have to enroll the help of an official, who can take various actions as described in [Polling Use Case, groupCJN, All members, 2011-12-04]

#### 6.2.5 Generation of voter cards

The description of this step has been deferred until now, since it may happen at several points in the process. Our initial thoughts when designing the system was, that all cards would be generated at the same point – at the central polling station. Through the project we have become aware, that this may not always be the way it works, and therefore the system supports printing of voter cards at both the central and at the different municipalities.
The printing works by allowing the user to filter voters through the same interface as that is used to filter when distributing voters into voter boxes. When a voter card is first printed, the corresponding voters 'card printed status' in the database is changed to true. This does not mean, that a voter card cannot be printed again, but simply that the user will be informed that the card has already been printed before, if he tries to print again.
The system allows a voter card to be printed multiple times, since cards may disappear during transport, be lost in a fire, and so on. Nevertheless, it is recommended to think twice before a card is printed again, and verify that a new copy is actually needed in order to avoid confusion regarding multiple copies of the same voter card at the voting day.
Multiple copies of the same card do not pose a security risk with regards to persons being able to vote multiple times, since the 'voted' status of a user is not registered based on any of the numbers printed on the voting cards, but on their CPR number, they may simply cause unneeded confusion for the polling workers at the polling station. We have decided to lower the requirement [req no. 1] occur.about each voter only receiving exactly one polling card, since voitng cards in our system are not directly related to a users 'voted' status.

## 7. Bibliography

The Code Project, especially the LINQ to entity tutorial http://www.codeproject.com/KB/linq/linqtutorial.aspx

## 8. Revision History

**Public Git repository:** https://github.com/nielssj/GroupCJN-E2011

## 9. Related Documents

### 9.1 Documents

DAO Testing strategy, groupCJN, jmei, 2011-12-10
Voter Card Generator Testing, groupCJN, nmar, 2011-12-13
Polling Table Testing, groupCJN, clae, 2011-12-10
Polling Use Case, groupCJN, All members, 2011-12-04
Security Reflections, groupCJN,  nmar / All, 2011-12-11
System Architecture, groupCJN, nmar / All, 2011-12-11

**9.2 Diagrams**
Voter card registration process, groupCJN, clae, ver.1.2
Diagram of voter distribution and network, groupCJN, clae / All, 2011-12-04
(See Graphical BON for an overview of the system)

**9.3 Bon documentation**
DVL_event_scenario_DAO.bon
DVL_formal_central.bon
DVL_formal_DAO.bon
DVL_formal_polling_table.bon
DVL_informal.html
DVL_informal_classes.bon
DVL_informal_system_and_clusters.bon
Visio-Overall_GraphicalBON.pdf
(All the informal BON have been compiled to HTML with BONc and are provided on the attached CD for convenience)