# Customer Errors in Internet Banking

Kai A. Olsen
Høgskolen i Molde & Universitetet i Bergen[1]

**Abstract**

Internet banking is based on letting the customer provide data entry and to give the necessary commands. Then typing errors are made. These are exemplified by a real life situation where NOK 500,000 was sent to the wrong account due to a typing error. In this paper we shall study this example, and also discuss the general case of typing errors made by Internet bank customers.

A study where 1800 transactions were entered into an Internet banking simulator by test persons provides background data. The study tells us which type of entry errors are made, and how often they occur. Some improvements to current banking interfaces in order to catch more errors are proposed.

## Introduction

Internet banking is commonplace. There are many advantages to be able to perform all standard banking services at home, 24 hours a day, 365 days a year. The tasks are formalized and the size of data that has to be entered is limited. In this respect Internet banking is the ideal Web application. Since all transactions can be performed automatically Internet banks can offer a good deal, low charges and high interest rates. Internet banking was tried as early as 1981, but needed the Web to become successful. Today a large part of all bank transactions are performed on the Internet. In some countries, as in Norway, Internet banking has a major market share.

However, Internet banking requires that the customer performs all data entry herself and that she operates the user interface in the correct manner. This is not always the case. There are many examples of customers that have transferred money to the wrong account or performed any other type of error that it is possible to make. Usually this has few consequences. Most account owners will return money that is received erroneously, and when the correct transfer have not been made one will always get a second chance, albeit with a fee for not keeping the due date.

Occasionally a keying error has serious consequences. A well-known case is the one of Grete Fossbakk, a Norwegian bank customer. She wanted to transfer a

[1] *This paper was presented at the NIK-2008 conference. For more information, see* `//www.nik.no`

large amount to her daughters account, typed twelve digits instead of the standard eleven, and the money went to the wrong account. In this case the account owner happened to be a person with no assets, on social security and addicted to gambling. In just a few days he had managed to loose a major part of the amount. The police confiscated the remaining, and the account owner was given a nine month prison sentence for using money that was not his own. However, this did not help Fossbakk who lost 400,000 NOK (approx 50,000 euro).

In the end Grete Fossbakk took the case to litigation with support of the Norwegian Consumer Council. In a few days before the court case the bank backed down and returned the full amount, with interest! Apparently they had seen that the effects of Fossbakk's error, and she clearly had made one, was due to faults in their Internet interface. It became apparent that the same fault was a part of many Internet banking systems in Norway and in other countries (feedback from Olsen, 2008, provided many examples). Interesting enough, it also became clear that these interfaces had not been tested for usability.

What happened in the Fossbakk-case was that the Web interface only accepted eleven digits in the account number field – the standard length of Norwegian account numbers. In her famous mistake Grete Fossbakk should have typed 71581555022, instead she entered 715815555022 (inserting an extra 5). The banking interface stripped off the last digit, and transferred the amount of 500,000 NOK to account no 71581555502.

In this paper we shall study what kind of key entry errors that a user may make using a Web-based banking system, both with regard to the Fossbakk case and in general. As a basis for our study we shall use data from a banking simulator. This system mimics the interface that was used in the Fossbakk case, used by a group of banks that had close to twenty percent of the Norwegian market. The simulator registers all error situations. A software analyzer is used to draw conclusions based on the data, which in this case have been provided by 69 testers who have entered thirty transactions each.

In addition to the data entry task, we shall study the confirm-operation that is a vital part of all Internet banking. That is, the situation where the complete transaction is presented to the user, allowing the user to cancel, edit or confirm.

## Background

A few studies on data entry were performed in the sixties, seventies and early eighties. These were the days of the keypunch. Interfaces were simpler, and there were fewer methods of finding erroneous data or correcting data. Since the hardware was so different from today, there are few findings that can be generalized. However, Card et. al (1984) tell us that keying errors are frequent, and that the second most used key on the keyboard is the backspace key. Today, we find several studies on entering data using cell phones, PDA's, voice input, etc. (see for example MacKenzie et al, 2002 and Kushniruk et al, 2004), but these are hardly relevant for our case where most users have access to a full PC with a large display and keyboard.

We should expect a research interest in this area as banks and other organizations started to allow customers to use their systems over the net. While punching most often were performed by professionals most Internet users are amateurs, even if many of these have some experience. No training is offered, and in many cases money is involved. Thus errors, as we have seen in the Fossbakk case, may have serious consequences. However, there are very few studies performed in the later years in this area. It seems that researchers have lost interest. Perhaps data entry is to dull in a world that offers so many fascinating areas of study, such as touch displays, cell phones, email, net meetings, Internet newspapers, Wikipedia, blogs, YouTube and Facebook .

While researchers may focus on other areas we should expect that the organizations behind the Web pages had performed extensive user studies. However, we have not found any report which focus on the data entry part. In the Fossbakk case it became clear that the banking group that used the interface in question had not performed any kind of usability studies. In the work that went preliminary to litigation it also became clear that they had no idea what was meant with a usability study. This may be called gross neglect, and may be one reason why the bank backed down in this case. There is, however, a possible explanation. The Internet came as a surprise to most organizations. Some saw the possibilities early and wanted to get their products out as soon as possible, others were forced to follow suit. Internet applications were of course a novelty to most customers, and the banks met few requirements from this side. It is as with the early cars, customers were amazed that it worked. Requirements for road stability, operation, comfort, safety and fuel economy came many years later.

Today, however, the Internet is becoming a mature technology and it is time to study what we have, to criticize and to offer ideas for better user interfaces.

## Internet banking interfaces

Internet banking may be offered by a brick and mortar bank, i.e., where the Web application is an add-on to the physical bank. A customer may then perform transactions both by traditional means, using checks, transaction forms and by a visit to the banking offices, or by using the Web interface. These banks have met competition from the pure online banks that meet customers only on the Web. The latter may offer better interest rates and lower fees than other banks since they have lower costs. In many countries, such as in Norway, we have seen that the new online banks have taken large market shares from the incumbents.

However, when it comes to the Web user interfaces there are few differences. The incumbents and the pure Internet banks offer both the same services: transferring money from one account to another, present the account balance, view last months transactions, look into archived records, retrieve a yearly statement and set up different forms of automatic transfers. There is nothing radical here. These are the same services that we performed in the physical banks in the seventies or by use of the phone-based banking systems in the eighties. One reason for the lack of development is that banks often use the same back-office system. That is, each bank or banking group may have its own interface, but only on top of a common platform.

We shall focus on the transaction part, e.g., paying a bill. The goal is then to transfer a given amount, on a set date, from one account to another. Additional data may be provided, such as a message to the receiver or a Customer Identification Number (CID). These systems may vary from country to country. In some countries, such as Norway, transfers are made simpler by the fact that all banking account numbers are nationwide. In other countries routing numbers or SWIFT codes may be needed to identify the receiver's bank. The account number will then only specify an account within one bank.

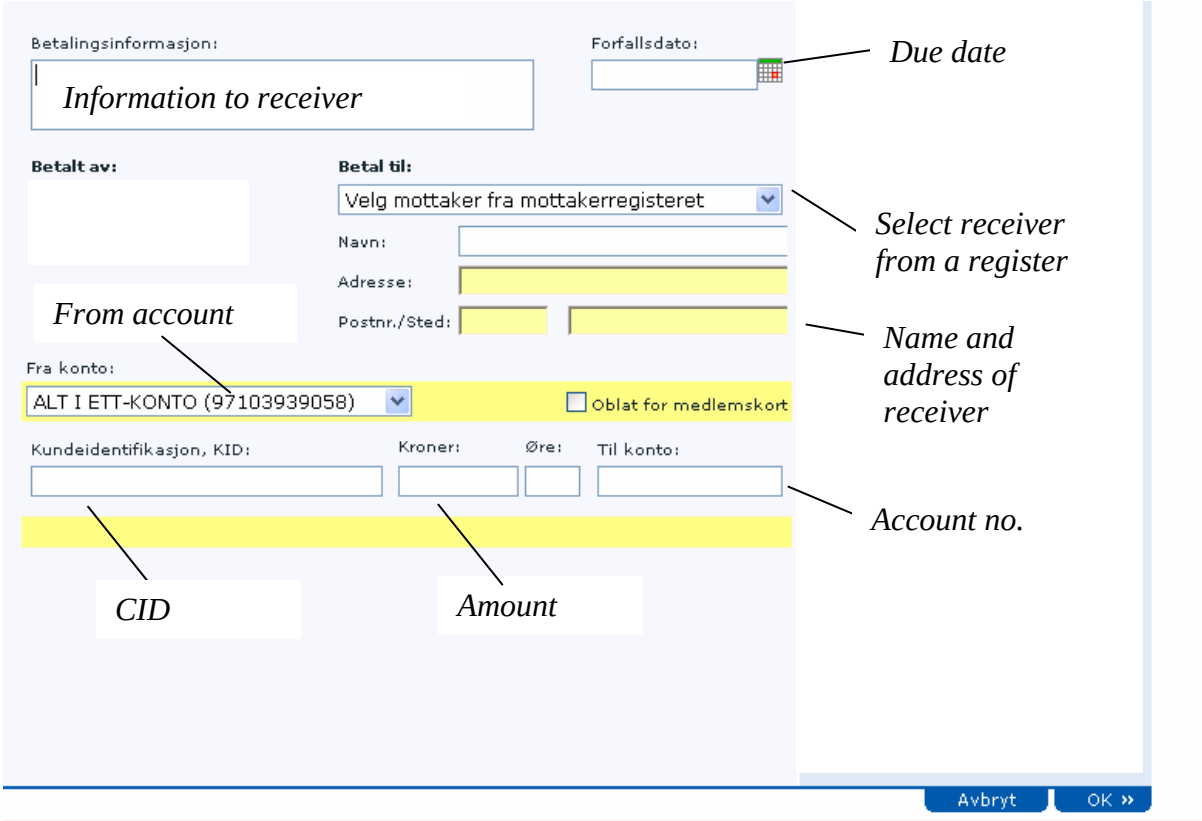


Figure 1 A typical transaction form

A typical Web-form used to perform these actions is presented in Figure 1 (here from Skandiabanken). Forms may vary from bank to bank, but are more similar than different. We find fields for the due date of the transaction, for a message to the receiver or a CID number, the amount, and the receivers account number. Both account numbers and CID have a checksum digit. There is thus the possibility of performing some simple checks on the validity of these data items.

The layout of the form is based on the traditional paper form, thus making it simpler for new users to start with Internet banking. After entering these data the user click on the OK-button. The data is then usually presented in another form, and the user is asked to confirm. Some banks require a code to verify the confirmation.

As seen, Internet bank interfaces are simplified by the fact that few data items are needed. Many banks, as Skandiabanken, offer a register over previous receivers. Thus the name, address an account number have only to be entered the

first time. However, even if there are few fields, even if limited amounts of data that has to be keyed in, there is, as we shall see, still ample room for making mistakes.

We shall study the problem by the use of an Internet banking simulator, and by letting 69 users enter thirty transactions each.

## The simulator

Usability data on the actual use of Internet banking is limited. We have not been able to find any publications in this area. In the litigation process we were able to get some overall data from the banks, but even here it seems that few data are logged on the server side. There is also the problem that there is no history of what actually happens on the users PC. However, provided that the banks were really interested in performing usability tests, there should be no problem of logging all user-input.

In our case the solution was to develop a banking simulator. This simulator performs all front-end operations, similar to a real banking system.

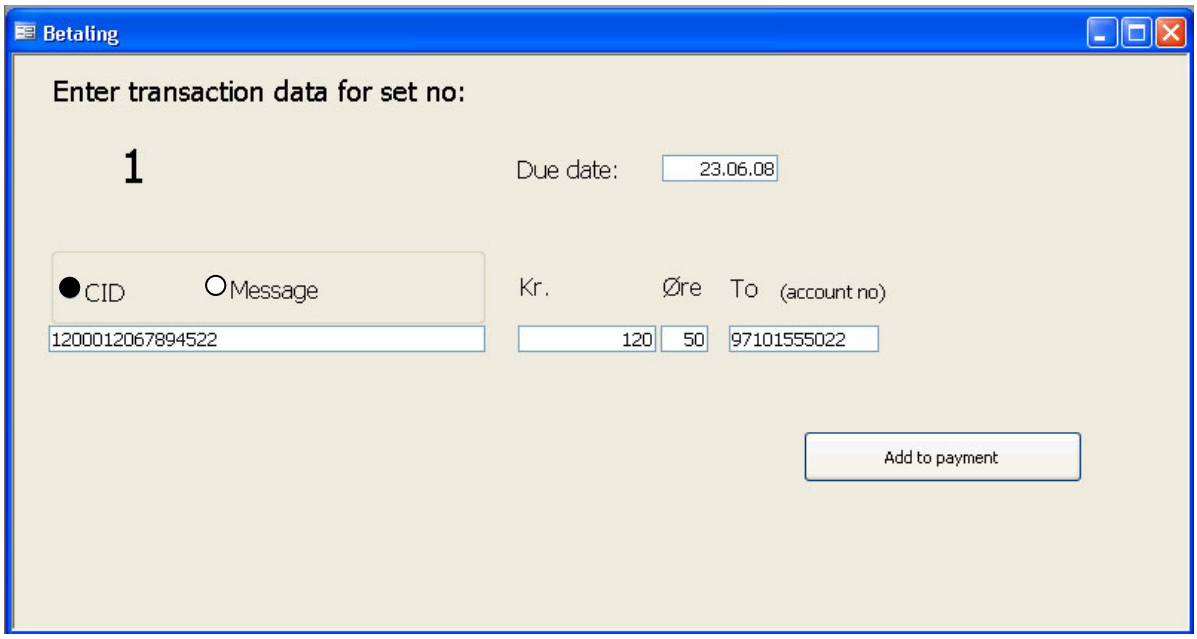


Figure 2 The simulator's transaction form

The simulator consist of a transaction form (Figure 2), similar to the one used by most of the Internet banking systems. It is nearly identical to the one used by the system where Grete Fossbakk made her big mistake. This system had a possibility of choosing a receiver from a register. This option was not used by Fossbakk and has not been included in the simulator. After all, we are interested in the keying part which is necessary for every new account even if a register is used.

After entering all data on the transaction form and after clicking on the “Add to payment” button, this interface provides a “confirm” form, which allows the user to cancel, edit or confirm the transaction.

The advantage by using our own system is that all data can be logged. The simulator registers all keying, and all button-clicks. So if the user hits the OK button on the transaction form, and chose to edit instead of giving the confirmation this is registered. All times are noted as well, something that allows us to see how fast each user manages to enter the data.

## Usability test

In the usability test 69 persons, students taking a college course in user interfaces, and two classes of high school students taking a course in IT, where asked to type in thirty transactions each. The transaction were offered as printed sheets, each transaction providing a due date, a message or a CID code, an amount and the receivers account number. After removing outliers we were left with data on 1,778 transactions. The data in the transactions are from real life situations. However, since Fossbakk made the error of adding an additional extra digit to a sequence, many of the account numbers used have sequences of two, three, four and even five similar digits – not uncommon to real life account numbers, but occur with a higher frequency in the test cases than elsewhere.

Clearly, we should expect higher error rates in the test than in the real world. The sequence of similar digits may result in higher error rates, but more important is the fact that in the test case users enters a large set of transactions. This occurs rarely with ordinary users in the real world. In addition the simulator does not offer any error messages. Everything is recorded, even wrong dates that may have been handled by a good user interface. In addition, one should consider that no money is involved in our case. Users may be more careful in a real life situation where real money is involved.

On the other side all our test persons have a relevant background as IT-students either in high school or in college. For this test it implies that all have Web experience, and that they can handle a keyboard and a computer.

## Results on typing errors

The students got 124 account numbers wrong. Since the transactions were entered from a list a typical error was to enter data on the wrong set, for example reentering data or enter parts of the data from one transaction and parts from the next. This occurred in 36 cases. In the following we have not taken into account this type of error since it will occur more frequently in a test case than in the real world[2].

This leaves us with 88 other types of errors in the account number, 5.1 percent of the total. The testers entered 44 erroneous CID codes, which is 3.6 percent of the transactions that had a CID number (the others offered a message to the user). The amount field (Kr-part) had 16 errors, 0.9 percent of the total and the date field had 20 errors, an error rate of 1.1 percent.

[2] The problem could have been avoided by offering transaction data on the computer display, with one transaction at a time. However, this would have required a second display on the computer to avoid interference with the data entry form and was therefore considered impractical.

| Field | Average length | Number of errors | Error rate in percent | Error rate/ length |
|---|---|---|---|---|
| CID (variable length) | 12 | 44 | 3.6 | 0.3 |
| Account number | 11 | 88 | 5.1 | 0.5 |
| Date | 6 | 20 | 1.1 | 0.2 |
| Amount (Kr-part) | 4 | 16 | 0.9 | 0.2 |

Table 1. Comparison of field length and error rate

Error rates are presented in Table 1. In the rightmost column we show the relation of error rate divided by field length. As seen small fields have a smaller error rate than the longer fields. However, the students seem to have been more careful in typing CID numbers than account numbers. There is a difference however, as CID numbers comes in various lengths.

| Type of error (Account no) | No of cases |
|---|---|
| Too long number | 36 |
| (adding digit in sequence of similar) | (18) |
| Too short account number | 31 |
| (missing digit in sequence if similar) | (21) |
| Wrong 11-digit account number | 21 |
| **Sum** | **88** |

Table 2. Types of errors in account numbers

If we analyze the types of errors that were made in the account number we see that 36 out of the 88 wrong numbers were too long, 31 too short and 21 had the correct length (11 digits) but with a typing error (Table 2). From the table we see that a typical error, when the number has an incorrect length, is to add or drop a digit in sequences of similar digits. For example, this was done in half of the too-long cases, and in two-thirds of the too-short cases. That is, the error that Grete Fossbakk made is a very typical error.

| Type of error (CID) | No of cases |
|---|---|
| Too long number | 16 |
| (adding digit in sequence of similar) | (8) |
| Too short CID number | 21 |
| (missing digit in sequence if similar) | (18) |
| Correct length, but wrong number | 7 |
| **Sum** | **44** |

Table 3. Types of errors in account number

Looking at CID numbers (Table 3) we will se a similar pattern. 16 out of the 44 errors here were due to a too long number, 8 of which occurred by the user adding an extra digit in a sequence. In 21 cases the CID was too short, all of 18 that occurred in a sequence situation.

We checked the data to see if there were more errors towards the end of the data set, for example due to the fact that the test person got tired. However, there was no significant difference. Indeed, a small discrepancy indicated that users typed better at the end that at the start of the test.

## Confirmation

In the process documents passed in front of the litigation, the bank stressed that Fossbakk had confirmed the transaction. That is, even if the interface had stripped off the twelfth digit and thus made a “new” account number, the final results were presented to Fossbakk in the confirmation form. However, instead of noting the error she confirmed the transaction. Thus, the bank argued that the faulty transaction was entirely Fossbakk’s mistake.


Figure 3. A confirmation form

The confirm form used by Fossbakk was similar to the one used in Figure 3 (here from Skandiabanken). The simulator uses a matching form.

Clearly, in our test the students have confirmed 88 transactions with the wrong account number (if we ignore the cases were they typed from the wrong data set). In addition, in order to study what really happens in the confirm phase, the simulator changed the entered account number automatically in every tenth transaction. That is, the user may have correctly typed 70581555022, a number that the simulator changed to 70581555502 before confirmation. Two other account numbers were also replaced by look-alikes in this manner. This was recognized by our test persons in only 5 out of 178 cases!

That is, we may conclude this part by noting that confirmation, while it may have some legal applications, offer no real value in detecting errors. It appears that most people perform the inspection while keying, not when the whole number is displayed onscreen. In many ways, this is efficient. While keying, we can concentrate on one digit at a time. After keying we have a large and complex number. If this *seems* correct, we hit the “confirm” button.

Psychologist Donald A. Norman explains this behavior in his book, *Psychology of Everyday Things* (1988). Here, a user confirmed deletion of his “most important work.” According to Norman, the user confirms the action, not the file name. Thus, the “confirm” part of the transaction has a minimal effect on detecting errors.

## Discussion

We see that users make mistakes both while keying and checking the input data. In this respect our test supports earlier findings. As long as transactions are entered in this manner errors are inevitable. However, with improved interfaces it is possible to reduce errors significantly. In the following we shall present a set of remedies that may be applied. All are simple to implement, some extremely simple, and all use the existing infrastructure.

More than fifty years ago, in his much cited paper “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information” Miller (1956) showed that humans on the average remembered seven digits of arbitrary numbers. The account numbers used in Norway and in this test are eleven digits long, which makes both entry and verification difficult. Just by grouping the digits this would be much easier. Compare 705815555022 and 70581555022 to 7058 155 5502 2 and 7058 155 5022. The error is recognized quite easy in the latter case. Even if the last digit is stripped off there is a clear difference between 7058 155 5502 and 7058 155 5022.

Today a checksum system is used both for the account number and the CID. But a one-digit checksum is no guarantee against mistakes. It did not help in the Fossbakk-case, and will, on the average, pass through as much as 8 % of all wrong account numbers. It should be easy to check if the account number entered is valid. It will not provide help against transferring money to the wrong account, but would provide important feedback when an illegal account number or a number that is not in use is entered.

In most cases the use of a register for previous used accounts limits the possibility of mistakes to first time use. In addition, the system should always present the name of the account owner. Some interfaces in Norway do this for accounts owned by organizations, but it is determined a break of privacy to present the name of personal account owners (Finansnæringens Hovedorganisasjon, 2006). There is, however, the possibility of showing only the first name or an alias. This would give the user a very good chance of identifying mistakes, without letting the system committing a breach of privacy.

But users make mistakes in other parts of the transaction as well. Typing errors in the date and amount part may to some extent be caught by pattern recognition. For example, if we transfer an amount to our power supplier every three months, a simple routine could give a warning if the amount or date was way off from previous values. Errors in dates may also be reduced by providing the user with a calendar, and by presenting the number of days until the due date.

Clearly, neither of these systems can remove errors, but combined they should be able to catch most of the mistakes that users perform. Many of these mistakes would have minor consequences, but even then one avoids extra costs and problems. In the end, the total effect of the improvements in the user interface may be checked with usability tests.

Of course, the final solution to this problem is to avoid all input by users. That is, to send an electronic bill to the bank in addition to sending it to the customer by

mail or email. Then user actions will be reduced to indicate payment of these pre-made transactions. However, it will only work in cases were an outside organization initiate the transaction. It would not help Fossbakk when transferring money to her daughter.

## Conclusion

A test based on a simulation of an Internet banking interface shows that users enters data incorrectly. While the actual error percentages may be higher in the test than in real life situations, the test should give a good idea of the types of errors that users make. Not surprisingly, the error rate increases with the length of the data fields. A common error is to add or remove a digit from a sequence of similar digits. The confirmation phase used in most Internet banks seems to be of little value. If numbers look similar the user often confirm without scrutinizing the values.

Current banking interfaces are rather primitive and do not help the user to catch errors. It is also apparent that many interfaces have not been subjected to usability testing. It seems that time to market were more important for the banks in their initial effort to get an Internet solution up and running than providing a good interface. However, we have had online banking for 25 years, if we count the pioneering systems, and it seems appropriate to demand better interfaces today. Some improvements in the current interfaces are suggested here, also a few which can be implemented with minimal effort.

## References


Card, S.K., Moran, T.P., Newell, A. (1980) The keystroke-level model for user performance time with interactive systems, Communication of the ACM, ACM Press, 23 (7).

Finansnæringens Hovedorganisasjon (2006) Trygghet ved bruk av nettbank – næringens anbefalinger, brev til Kredittilsynet av 18.10.

Kushniruk, A. W., Triola, M. M., Borycki, E. M., Stein, B. , Kannry, J. L. (2004) Technology induced error and usability: The relationship between usability problems and prescription errors when using a handheld application, *International Journal of Medical Informatics* Volume 74, Issues 7-8

MacKenzie, I.S., Soukoreff, R. W. (2002) Text entry for mobile computing: Models and methods, theory and practice. *Human Computer Interaction*, Lawrence, Erlbaum Associates.

Miller, G. (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, *The Psychological Review*, vol. 63, pp. 81-97

Norman, Donald A. (1988) The Psychology of Everyday Things, Basic Books, New York.

Olsen, K.A. (2008) A $100,000 keying error, IEEE *Computer*, April, vol. 41, No. 4.