TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

IT40LT

Vadim Kööp 142676

Application for door security log analysis

Bachelor’s thesis

Supervisor: Gert Kanter

M.Sc.

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

IT40LT

Vadim Kööp 142676

Rakendus ukse lukusüsteemi logi analüüsimiseks

Bakalaureusetöö

Juhandaja: Gert Kanter

M.Sc.

Tallinn 2020

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Vadim Kööp

08.01.2020

Abstract

The goal of this thesis is to develop the full-fledged computer application that would correctly read log files with data and provide for the company the correct information about the usage of the doors in the building for a certain period of time.

The company Transcom Estonia OÜ noticed that the monthly statistics of the usage of the doors in the building do not correspond to the expected result. At the moment the company uses FoxSec Access 1850 Version 17.2 application, Insight Review V5 application and web Excel script which do not have all the functionality that provides convenient work with data analysis.

The company wants to analyse log files more effectively.

To achieve this goal a survey of employees and separate interviews with the IT department were conducted. The applications used by the company were analysed. The UML diagram of the whole process was created. The functional requirements were described as use cases and nonfunctional requirements with the quality tables. The framework for development and design were analysed.

The development of the application was successfully completed. The unit tests were created for application testing. User acceptance tests passed successfully. Positive customer feedback has been received and the application has been approved.

This thesis is written in English and is 40 pages long, including 9 chapters, 11 figures and 3 tables.

Annotatsioon

Käesoleva töö eesmärgiks on luua täieulatuslik arvuti rakendus, mis loeb korrektselt andmefaile ning annab tulemuseks informatsiooni uste kasutamise kohta mingis ajaperioodis.

Ettevõte Transcom Estonia OÜ märkas, et avatud uste kuu statistika ei vasta oodatud tulemustele. Praegu seisuga kasutab ettevõte FoxSec Access 1850 Version 17.2 rakendust, Insight V5 rakendust ning Google'i skripti, mis ei võimalda vajalikke funktsioone andmete analüüsimiseks. Ettevõte soovib parandada uste turvasüsteemi oma hoonel.

Selle eesmärgi saavutamiseks viidi töötajate vahel läbi küsitlused ning IT-osakonnas intervjuud. Analüüsiti rakendused ning koostati kogu protsessi kohta UML-skeem. Funktsionaalseid nõudeid kirjeldati kui kasutusjuhtumeid ja mittefunktsionaalseid nõudeid kvaliteeditabeliga. Javarakenduste tüübid, arendus- ja disaini raamistik analüüsiti teaduslike artiklitega.

Rakenduse testimise jaoks loodi üksuste testid, mis oli samuti edukad. Kliendi tagasisiside oli positiivne.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 40 lehekülge, 9 peatükki, 11 joonist, 3 tabelit.

List of abbreviations and terms

API	Application Program Interface
SDK	Software Development Kit
JavaFX	Java-based platform for the graphical interface
Java	Computer Programming Language
CSS	Cascading Style Sheet
JRE	Java Runtime Environment
JUnit	Unit testing framework for the Java programming language
UML	Unified Modeling Language
User story	Informal, natural language description of one or more features of a software system.

Table of contents

List of figures

List of tables

1 Introduction

The main goal of the thesis is to develop a full-fledged computer application that would correctly analyse log files with the information about the door usage in the building and provide this information to the employees.

First, it is necessary to analyse the company's door usage workflow logic, analyse currently used applications, their strengths and weaknesses. It is needed to plan, design and implement the application that meets the client's functionality requirements. The application needs to be validated and user-tested.

For today, the Transcom Eesti OÜ using next applications:

- FoxSec Access 1850 Version 17.2
- Insight Review V5
- Excel script

All of the above applications have an old and poor interface as well as incorrect functionality.

At the moment the building has 13 doors. Each door has its own electronic card reader, which scans the cards and sents the information to the database. The information about the door usage stores in 2 type of log files. For each file analysis company uses different applications because of the different log file format and language.

The building of the company has 3 floors with several offices and over 300 employees. The movement tracking of all employees in the building is a complicated process because the doors are used several times every day and several of them could be opened by the button instead of a card.

1.1 Problem overview

The company Transcom Estonia OÜ noticed that the monthly statistics of the usage of the doors in the building do not correspond to the expected results. The company wants to improve log files analysis and identify the correct frequency of the door usage by the employees or guests for every month. The company has the door security system that scans employee's cards and stores information about the usage of the doors in the company's database, but they do not have a single application that combines the functionality of fragmented applications and that would provide the correct analysis of the log files. At the moment, the company uses the web Excel script, which is developed by the IT department, FoxSec Access 1850 application version 17.2 and Insight Review V5 application that does not have all the functionality that supports convenient work with log files analysis. The company would like to get an application that would correctly read log files and provide to the employees the correct analysis. The application should be easy in use and available on all office computers. The client requires that the data should be confidential and also could be applied for internal use.

1.2 Objectives

The purpose of the application is to provide an alternative solution to the problem that would improve the workflow of the previous application and would provide correct information. The main objectives are:

- Analyse the existing applications that were used before.
- Analyse the client's requirements.
- Create a UML diagram of the whole process.
- Define the functional and non-functional requirements for the development application.
- Test the developed application and make sure that it is working correctly.
- Develop intuitive design for application.
- Pass all the acceptance tests and get positive feedback from the client.
- The expected result is an application for the client.

1.3 Outline

The thesis is organized in the following way. Chapter 2 contains all theoretical about the work process, development and testing. Chapter 3 contains all information about methodology, planning and analysis. Chapter 4 is about analysis of the applications which were used by the company. Chapter 5 introduces the functional and non-functional requirements for application development. Chapter 6 is about the log files and developed application workflow logic. Chapter 7 is about the new graphical user interface design for developed application. Chapter 8 is about the result, feedback and future area for improvement.

2 Theoretical background

In this chapter would be presented all the necessary theoretical material for the understanding of this thesis. Application development requires a systematic approach and many different techniques exist. Best practises need to be considered. The log files contain structured information, which is both readable by humans and possible to process by machines, since it was created by automatic processes. There is a high volume of irrelevant information in the files, so it is convenient to work with such files using the regular expressions because they allow to analyse and extract the necessary information quickly.

2.1 Regular expressions

The regular expression is a special pattern which helps to find a specific text in string format. The regular expression makes search and editing processes more efficient and faster [1].

2.2 Process models

Software process model - is an abstract representation of a process that describe all the processes from particular perspectives. This model includes the following processes

1) Specification – detailed description of the purpose of the application and its functions. 2) Design and implementation – determination of the organization and implementation of the application.

3) Validation – comparison of what the client wants to see and what the application does.
4) Evolution – further application updates requested by the client. [2].

2.3 Iterative waterfall

Iterative waterfall model - is the extension of the waterfall model. Iterative waterfall model is almost the same as the waterfall model expect some modifications which helps to improve the performance of the software development. This model provides the opportunity to receive feedback from the client at each stage, including the previous ones in case of problems detection. Such a model is very simple to understand and it allows to change the requirements and if it is necessary it can make some changes at the previous stages. Such a model helps to reduce the development process and save a lot of time. This model is very suitable for projects such as this [3].

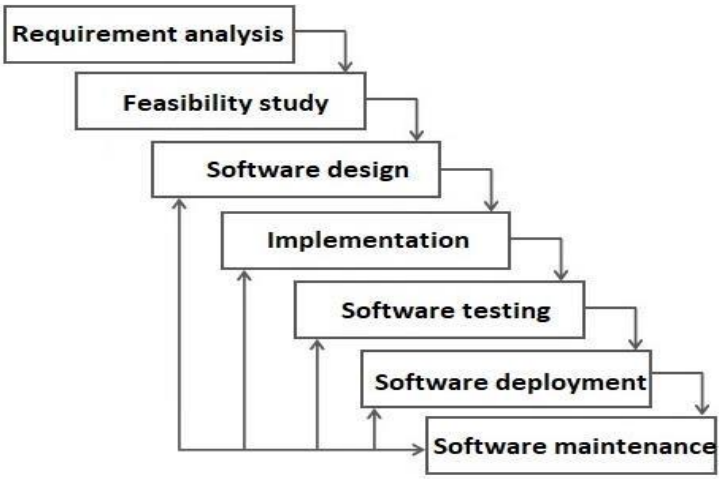


Figure 1. Iterative Waterfall model.

2.4 Requirements analysis

In systems engineering and software development, requirements analysis is focused on the project tasks and conditions which satisfy the client. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Types of requirements:

1) Customer requirements – all the requirements for the project provided by the customer side. This helps draw up a further action plan.

2) Architectural requirements – definition of the interaction between application packages, databases, and systems in terms of functional coverage. This helps identify the integration problems and gaps in functionality.

3) Structural requirements - explanations of what should be done by identifying the necessary structure of application development.

4) Behavioural requirements – explanation of the necessary behaviour of the developed application.

5) Functional requirements – explanation of the actions that have been accomplished. 6) Non-functional requirements - requirements that specify the criteria of the developed application.

7) Performance requirements - requirements based on factors of the life cycle of the developed application and the degree of reliability in their assessment.

8) Design requirements - all the requirements for the project provided by the client [4].

2.5 Software quality

The software quality model determines which quality characteristics would be analysed. It helps to give the correct assessment of the developed product. To identify all the features of a successful product, the development team have to conduct a conversation with the client of the product before starting the development.

Functional Suitability – characteristic, which reveals the needs when using the developed application.

This characteristic consists of the following sub-characteristics:

1) Functional completeness – the degree where the functionality covers all specified tasks.
2) Functional correctness – the degree where the product covers all correct results.
3) Functional appropriateness – the degree where functions improve the process of other tasks.

Performance efficiency – characteristic, which shows the performance relative to the amount of resources.

This characteristic consists of the following sub-characteristics:

1) Time behaviour – the degree where response time and processing, as well as the throughput of the product in the performance of its functions, meet the requirements.
2) Resource utilization – the degree of conformity of the quantities and types of resources used by the product in the performance of its functions.
3) Capacity – the degree where the maximum limits of product parameters meet requirements [5].

2.6 Model view controller

Model view controller pattern – the pattern which organise and involves the allocation of blocks responsible for solving various problems.

Modal - this component is responsible for the data and also determines the structure of the application.

View - this component is responsible for user interaction.
Controller - this component is responsible for the relationship between model and view [6].

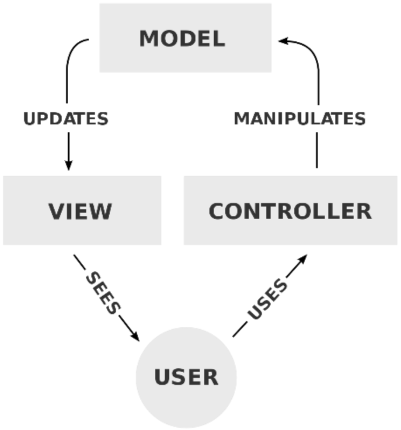


Figure 2. Model view controller.

2.7 Java programming language

Java programming language - is the general programming language that is used for desktop, web, mobile and enterprise application development.

Java includes next tools:
Java Development Kit (JDK) – allows to write and compile Java code.
Java Runtime Environment (JRE) – software distribution tool.
Integrated Development Environment (IDE) – helps during the development process. [7].

2.8 Java application types

Applications developed in the Java programming language can be divided into three groups according to their purpose and functionality:

- Java stand-alone applications
- Java Applets applications
- Java Servlets applications

The author would research every type of java application and decide which type is more appropriate for the Transcom Eesti OÜ.

2.8.1 Java stand-alone applications

A stand-alone application intended for offline use is compiled and run on a local machine running the Java Runtime System [8].

2.8.2 Java applets applications

Applet Java applications that are interpreted by the Java Virtual Machine built into all modern browsers. Each applet application is dynamically loaded over a network from a Web server. The main feature of applets is that they are real applications, not the files for storing information. With the help of applets, users can make Web server pages dynamic and interactive. Applets allow performing complex local processing of data received from the Web server or entered by the user from the keyboard [9].

2.8.3 Java servlets applications

Java servlets are the Java applications that run on a webserver to generate dynamic web content. Java provides support for web applications through Servlets, Struts or JSPs [10].

2.8.4 Summary of applications types written in java

Were considered three types of application types written in java – stand-alone applications, applets applications, servlets applications. Each development is good in its field. The choice of development will depend on the functionality of the project, which will be offered to the developer.

Table 1. Advantages and disadvantages of three types of application written in java.

Type of applications written in java	**Advantages**	**Disadvantages**
Java stand-alone applications	+ Secure + Anonymity + Quick installation	- Single computer - Control
Java Applets applications	+ Cross-Platform + Compatibility + Quick Execution + Scalable Work	- Support of browsers - Manual installation or update - Do not access client-side resources - Have to tag as trusted applets - Cannot work with native methods

Java Servlets applications	+ Fundamental architecture + Performance + Secure + Flexible	- Difficult design can slow down the application - Complex business logic makes the application difficult to understand - JRE need on the server to servlets

Based on this collected information, was chosen the Java stand-alone application type, based on the author’s previous experience with it and convenience in usage. The developed application would be working only with the log files. So it does not need an internet connection and other auxiliary technologies [11].

2.9 User interfaces

The user interface - is the point of human-computer communication in a device. This could include display screens, keyboards, computer mouses and etc.

Criteria for a good user interface:

1. Clarity - the interface does not have to be clogged up and is easy to use.
2. Concision - the interface should not be clogged with endless information. It should be made as simple as possible and maximum effective.
3. Familiarity - an interface that when the user is already familiar with it on a subconscious level.
4. Responsiveness - a good interface that provides good feedback to the users.
5. Consistency – consistent interface across the whole application without extra information.
6. Aesthetics - The interface should adhere to one style. You can not combine different styles in one because it can confuse the user.
7. Efficiency - interface should help users speed up their workflow.
8. Forgiveness - Each user can always make a mistake, so the interface should always provide the opportunity to correct their mistakes without catastrophic losses for the user [12].

3 Methodology

The development process for the application needs to follow the current industry standards and best practices. There are many standards and practices about how these types of application could be developed.

Because of the project scope and size, the chosen approach is - Iterative waterfall, which combines the classical step-by-step development methodology with the flexibility of agile approaches. If at any step it is necessary to go back to the previous stage, iterative waterfall approach allows that. The programming language is java because of its simplicity and the author has practical experience with it.

First, the analysis and planning would be produced. Secondly, a special design for the application that would meet all of the requirements of the client would be developed. Third, implementation and validation would be done. Fourth, Unit testing would be written for the main logic and critical moments of the application would be tested. Fifth, the applications would be compared with other applications. The application would have additional functionality which is not present in the previous applications that the company used. The application would be convenient and intuitive in use, as well as simplify the workflow. Once the solution meets all of the requirements it would be tested by acceptance testing. Sixth, the evaluation would be received.

3.1 Analysis and planning

For the best and most effective performance of the work, the author would meet with the client on an ongoing basis, analyse in detail each upcoming and taken step, edit the existing information and find out the main problems, also would learn and predict further requests of the client that may come during the process in the future, develop the application.

3.2 Implementation

During the product implementation using the Java FX programming language, the author would develop an approach that would allow achieving maximum efficiency and future expansion, change, complement, then write an app.

3.3 Design

Good design is one of the main components of the application because with it the users start communication with the application. There are certain rules that need to be maintained during the development process. Since it is these rules that will determine the continued success of the developed application [13 – 18].

Table 2. 10 Usability for User Interface design.

Category	Description
Visibility of status	The application should always inform users about what is happening.
Connection between application and the real world	The application should have understandable language.
User control and freedom	The application should have a clearly marked "emergency exit" to leave.
Consistency and standards	The application should not have different actions with one meaning.
Error prevention	The application should have a good design that will not cause additional errors.
Recognition rather than recall	The application should give only important information.
Flexibility and efficiency of use	The application should allow users to tailor frequent actions.
Aesthetic and minimalist design	The application should not contain irrelevant information.
Help users recognize, diagnose, and recover from errors	The application should give an error message and offer an alternative solution.
Help and documentation	The application should have documentation.

3.4 Validation

During the application validation process, the author would check if the application meets the expectations and requirements of the client, also the dynamic mechanism for validating would be checked and the actual product would be tested. The validation would involve next processes: application code running, black and white box testing, non-functional requirements testing, searching errors that the verification process cannot catch, checking the final product.

3.5 Evaluation

It is very important that all acceptance tests of the developed application would be passed successfully and positive client feedback would be received. All this would mean that the author coped with the goals and achieved the correct final result [19].

During the evaluating process the application, specific requirements would be analysed and identified, prototypes developed, discussed and tested, reviews and ideas from both sides would be selected and argued.

3.6 Tools

To implement this project, the main technologies that can be applied are Web, Java, C ++, C# and etc.

The Web cannot be used because it does not fit this project and internal company security rules. C ++ should be used in large and complex projects that require high efficiency. C# and Java could be used in such type of projects.

Due to the fact that the author has the most experience working with Java and JavaFX programming languages because of the most academic subjects at the university that were related to programming and application development in Java programming language. Java development choices would give the opportunity to the author to develop the stand-alone application more quickly and efficiently.

4 Analysis of existing the solutions used by Transcom Eesti OÜ

This chapter deals with the analysis of the FoxSec Access 1850 Version 17.2 application, Insight Review V5 application, web Excel script and employee's opinions about it [20 – 21].

Before creating a new application, the first step is to analyse the existing applications that the Transcom Eesti OÜ used. This can help with the user experience and the development in the future, for instance, with functional and non-functional requirements. Some features from the existing applications would be added to the developed application.

4.1 Door usage process

Each door has own electronic card reader, which scans the cards and sent information to the database. The information about the door usage stores in 2 type of log files because each log file has own language and format. Log files stores the next information: address, date, time, transaction ID, employee ID, employee name, door address, card reader model, door successful access, door access denied, door locking timer status open/close, door alarm, door open by the button, door open by the card, the door opened too long, door ID, door name and etc. The company has the opportunity to pull out all the selective information from the database to the log files, but it is not convenient and takes more time because of this formated is set by the default.

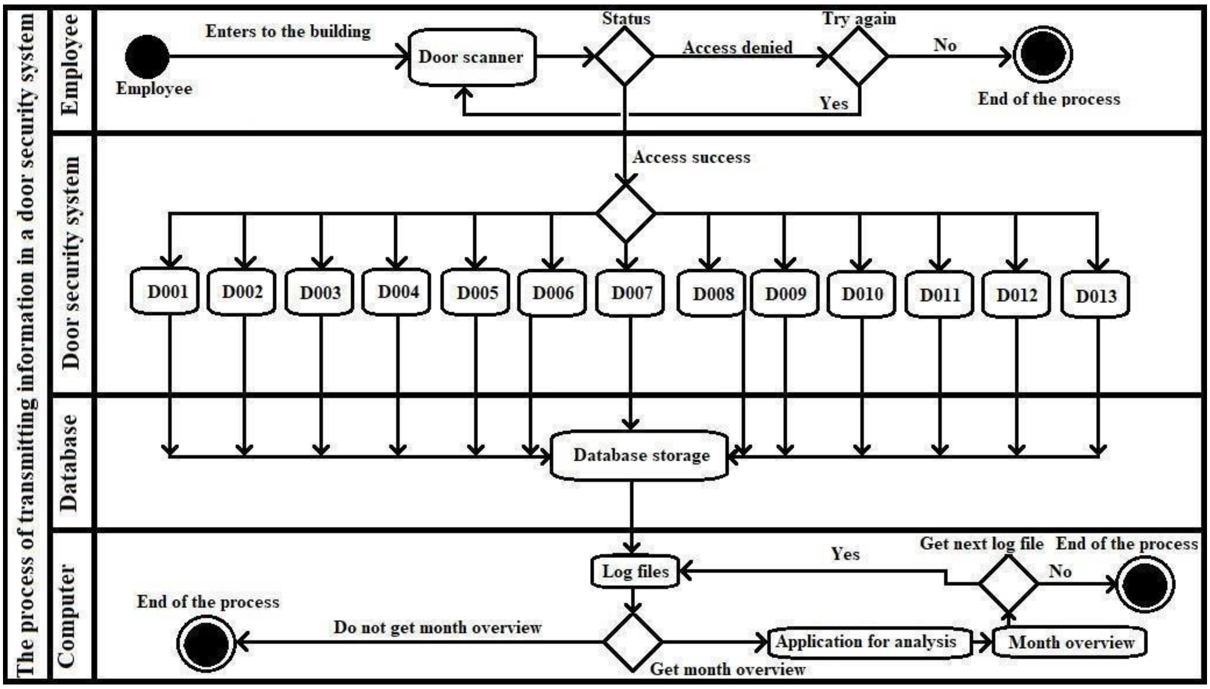
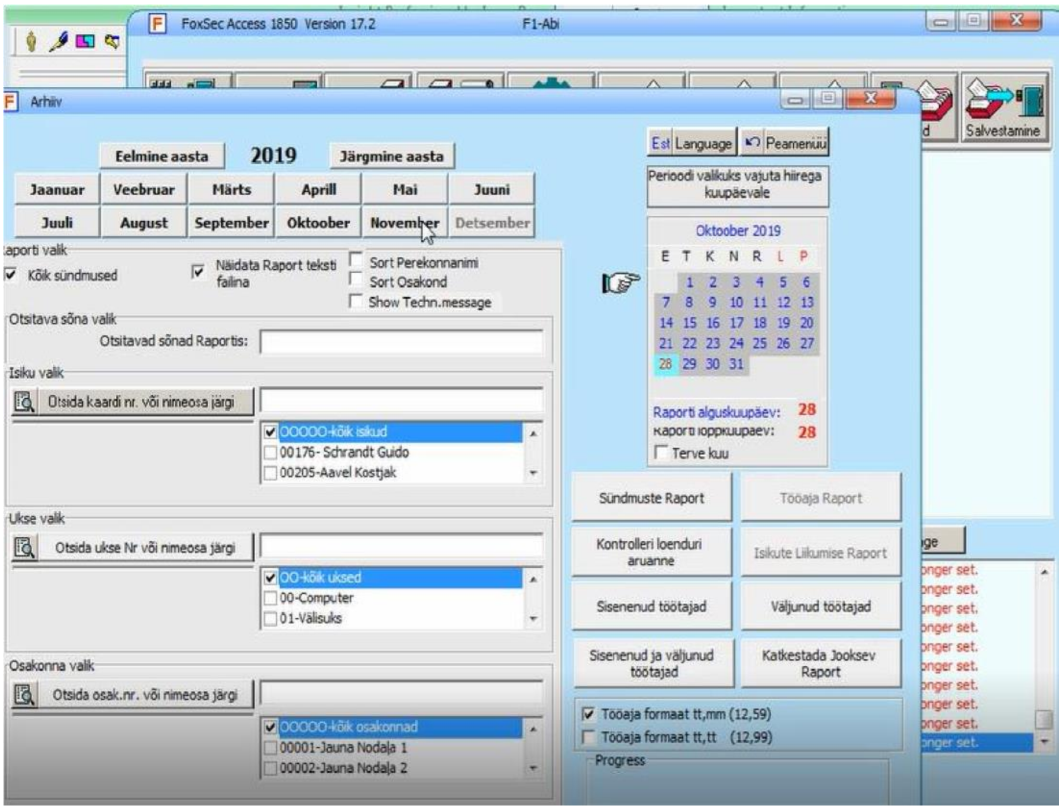


Figure 3. UML diagram of the whole process.

Figure 3 shows the whole process description. The employee validates the card, if it denied, the employee can try to validate it again or end the process, if it a success, the employee can open one of the doors. After that, all information about the door usage goes to the database storage. Then the company employee can get the log text file from the database storage and analyse it through the application or end the process.

4.2 FoxSec Access 1850 V17.2

FoxSec Access 1850 V17.2 application has fairly rich functionality and at the same time is very difficult for ordinary users to understand it. The application has a lot of extra features that the company does not use. The application has not been updated for a long time and therefore during the work processes it has issues and requires a lot of actions from the user, which slows down the analysis process (5.1 Functional requirements). The application also has an outdated design. This application does not meet all the requirements of the correct user interface (2.10 User interfaces) and modern design (3.2 Design). The application is used for Estonian log files analysis [22].



☑ Kõik sündmused
☑ Näidata Raport teksti failina
□ Sort Perekonnanimi
□ Sort Osakond
□ Show Techn.message

☑ 00000-kõik isikud
□ 00176-Schrandt Guido
□ 00205-Aavel Kostjak

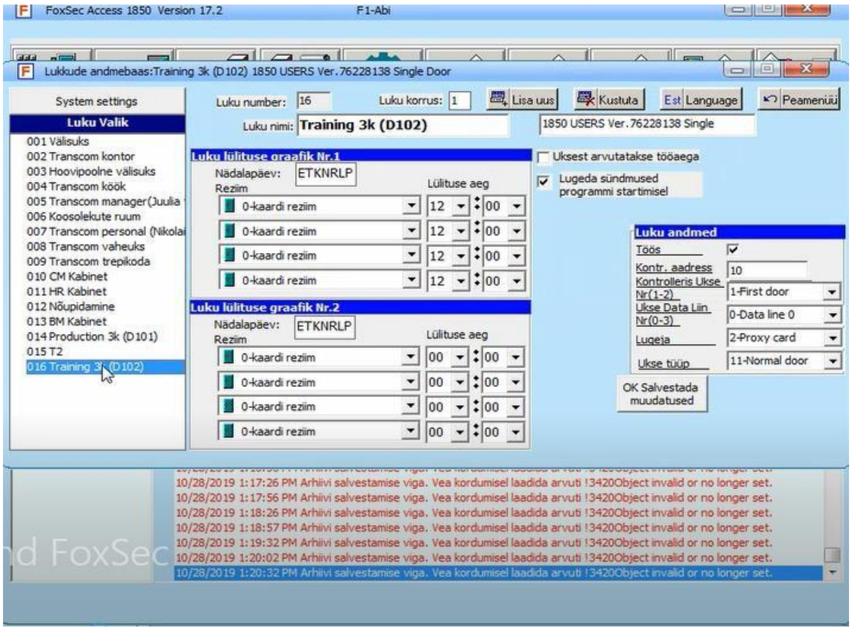
☑ 00-kõik uksed
□ 00-Computer
□ 01-Välisuks

☑ 00000-kõik osakonnad
□ 00001-Jauna Nodaļa 1
□ 00002-Jauna Nodaļa 2

Raporti alguskuupäev: 28
Raporti lõppkuupäev: 28
□ Terve kuu

☑ Tööaja formaat tt,mm (12,59)
□ Tööaja formaat tt,tt (12,99)

Figure 4. FoxSec Access 1850 V17.2 interface – 1.



FoxSec Access 1850 Version 17.2 F1-Abi

Lukkude andmebaas:Training 3k (D102) 1850 USERS Ver.76228138 Single Door

Luku number: 16 Luku korrus: 1

Luku nimi: Training 3k (D102)

1850 USERS Ver.76228138 Single

001 Välisuks
002 Transcom kontor
003 Hoovipoolne välisuks
004 Transcom köök
005 Transcom manager(Juulia
006 Koosolekute ruum
007 Transcom personal (Nikolai
008 Transcom vaheuks
009 Transcom trepikoda
010 CM Kabinet
011 HR Kabinet
012 Nõupidamine
013 BM Kabinet
014 Production 3k (D101)
015 T2
016 Training 3k (D102)

Luku lülituse graafik Nr.1

Nädalapäev: ETKNRLP

Reziim	Lülituse aeg
0-kaardi reziim	12 : 00
0-kaardi reziim	12 : 00
0-kaardi reziim	12 : 00
0-kaardi reziim	12 : 00

Luku lülituse graafik Nr.2

Nädalapäev: ETKNRLP

Reziim	Lülituse aeg
0-kaardi reziim	00 : 00
0-kaardi reziim	00 : 00
0-kaardi reziim	00 : 00
0-kaardi reziim	00 : 00

□ Uksest arvutatakse tööaega

☑ Lugeda sündmused programmi startimisel

Luku andmed

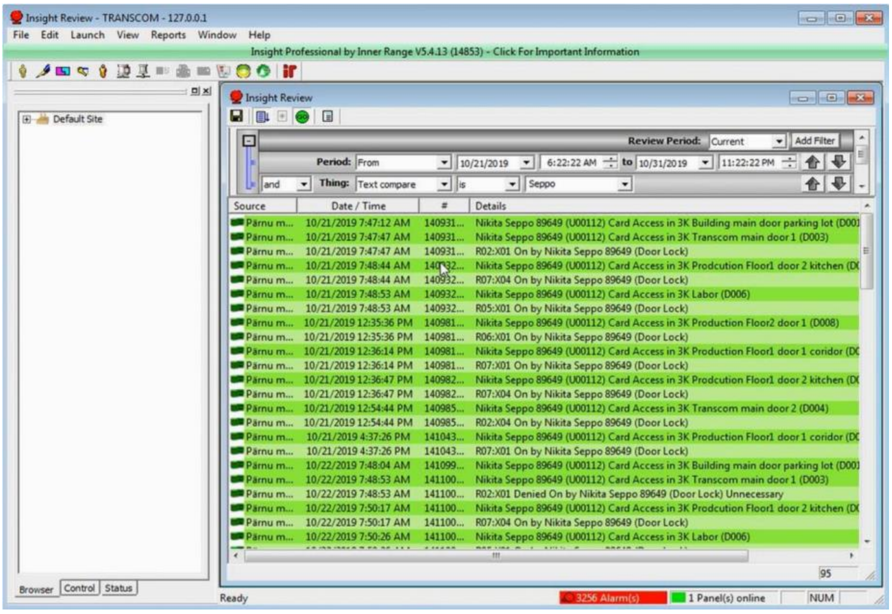
Töös	☑
Kontr. aadress	10
Kontrolleris Ukse Nr(1-2)	1-First door
Ukse Data Liin Nr(0-3)	0-Data line 0
Lugeja	2-Proxy card
Ukse tüüp	11-Normal door

```
10/28/2019 1:17:26 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
10/28/2019 1:17:56 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
10/28/2019 1:18:26 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
10/28/2019 1:18:57 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
10/28/2019 1:19:32 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
10/28/2019 1:20:02 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
10/28/2019 1:20:32 PM Arhiivi salvestamise viga. Vea kordumisel laadida arvuti !3420Object invalid or no longer set.
```

Figure 5. FoxSec Access 1850 V12.2 interface – 2.

4.3 Insight Review V5

Review Insight V5 application has almost all the necessary functionality that the company wants to use. The application is pretty convenient and easy in use, but sometimes there is a little inconvenience with the interface (2.10 User interfaces), also the application does not always give the correct results during the log files analysis (5.1 Functional requirements). The application has an outdated design this why it does not meet all the requirements of the correct modern design (3.2 Design). The application is used for the English log files analysis [23].



Insight Review - TRANSCOM - 127.0.0.1

Insight Professional by Inner Range V5.4.13 (14853) - Click For Important Information

Insight Review

Review Period: Current

Period: From 10/21/2019 6:22:22 AM to 10/31/2019 11:22:22 PM

and Thing: Text compare is Seppo

Source	Date / Time	#	Details
Pärnu m...	10/21/2019 7:47:12 AM	140931...	Nikita Seppo 89649 (U00112) Card Access in 3K Building main door parking lot (D001
Pärnu m...	10/21/2019 7:47:47 AM	140931...	Nikita Seppo 89649 (U00112) Card Access in 3K Transcom main door 1 (D003)
Pärnu m...	10/21/2019 7:47:47 AM	140931...	R02:X01 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 7:48:44 AM	140932...	Nikita Seppo 89649 (U00112) Card Access in 3K Prodcution Floor1 door 2 kitchen (D(
Pärnu m...	10/21/2019 7:48:44 AM	140932...	R07:X04 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 7:48:53 AM	140932...	Nikita Seppo 89649 (U00112) Card Access in 3K Labor (D006)
Pärnu m...	10/21/2019 7:48:53 AM	140932...	R05:X01 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 12:35:36 PM	140981...	Nikita Seppo 89649 (U00112) Card Access in 3K Production Floor2 door 1 (D008)
Pärnu m...	10/21/2019 12:35:36 PM	140981...	R06:X01 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 12:36:14 PM	140981...	Nikita Seppo 89649 (U00112) Card Access in 3K Production Floor1 door 1 coridor (D(
Pärnu m...	10/21/2019 12:36:14 PM	140981...	R07:X01 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 12:36:47 PM	140982...	Nikita Seppo 89649 (U00112) Card Access in 3K Prodcution Floor1 door 2 kitchen (D(
Pärnu m...	10/21/2019 12:36:47 PM	140982...	R07:X04 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 12:54:44 PM	140985...	Nikita Seppo 89649 (U00112) Card Access in 3K Transcom main door 2 (D004)
Pärnu m...	10/21/2019 12:54:44 PM	140985...	R02:X04 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/21/2019 4:37:26 PM	141043...	Nikita Seppo 89649 (U00112) Card Access in 3K Production Floor1 door 1 coridor (D(
Pärnu m...	10/21/2019 4:37:26 PM	141043...	R07:X01 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/22/2019 7:48:04 AM	141099...	Nikita Seppo 89649 (U00112) Card Access in 3K Building main door parking lot (D001
Pärnu m...	10/22/2019 7:48:53 AM	141100...	Nikita Seppo 89649 (U00112) Card Access in 3K Transcom main door 1 (D003)
Pärnu m...	10/22/2019 7:48:53 AM	141100...	R02:X01 Denied On by Nikita Seppo 89649 (Door Lock) Unnecessary
Pärnu m...	10/22/2019 7:50:17 AM	141100...	Nikita Seppo 89649 (U00112) Card Access in 3K Prodcution Floor1 door 2 kitchen (D(
Pärnu m...	10/22/2019 7:50:17 AM	141100...	R07:X04 On by Nikita Seppo 89649 (Door Lock)
Pärnu m...	10/22/2019 7:50:26 AM	141100...	Nikita Seppo 89649 (U00112) Card Access in 3K Labor (D006)

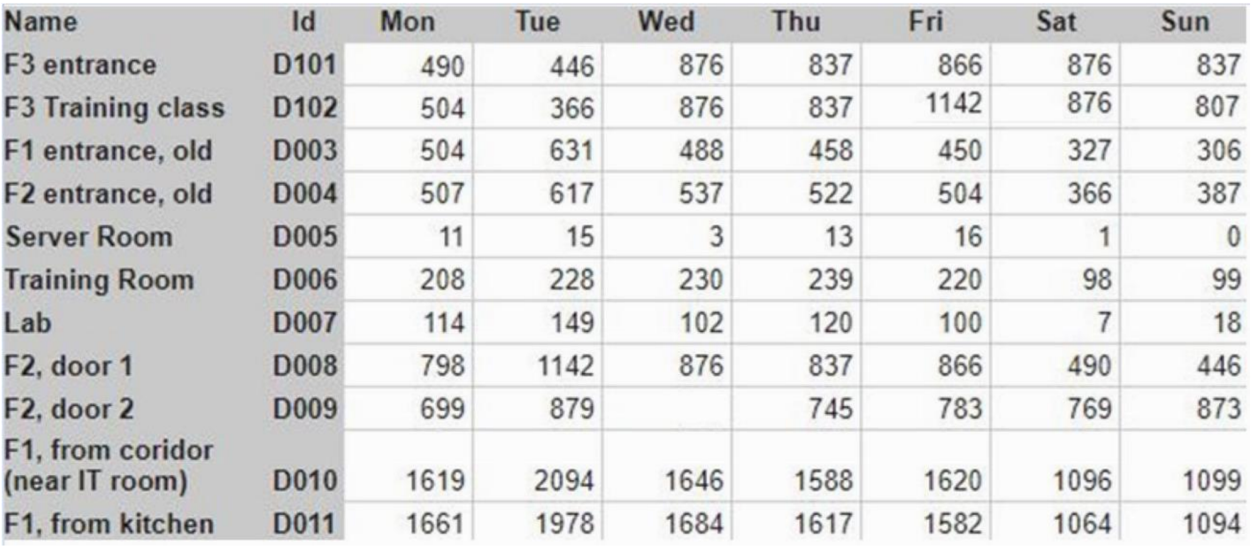
95

Ready 3256 Alarm(s) 1 Panel(s) online NUM

Figure 6. Insight Review V5 interface.

4.4 Web script

The Transcom Eesti OÜ has its own web Excel script, where an employee can analyse the log files. However, the resulting overview does not give the correct results (5.1 Functional requirements), also there were situations when the Internet connection was lost, and the company's employees could not access the web application, which reduced the efficiency of the work process (5.2 Non-functional requirements).



Name	Id	Mon	Tue	Wed	Thu	Fri	Sat	Sun
F3 entrance	D101	490	446	876	837	866	876	837
F3 Training class	D102	504	366	876	837	1142	876	807
F1 entrance, old	D003	504	631	488	458	450	327	306
F2 entrance, old	D004	507	617	537	522	504	366	387
Server Room	D005	11	15	3	13	16	1	0
Training Room	D006	208	228	230	239	220	98	99
Lab	D007	114	149	102	120	100	7	18
F2, door 1	D008	798	1142	876	837	866	490	446
F2, door 2	D009	699	879		745	783	769	873
F1, from coridor (near IT room)	D010	1619	2094	1646	1588	1620	1096	1099
F1, from kitchen	D011	1661	1978	1684	1617	1582	1064	1094

Figure 7. Web application of the Transcom Eesti OÜ.

Figure 7 shows the web application which demonstrates an overview number of successful entries in each of the 11 doors of the building, where the number of successful entries is summed up from the whole month and is given as a week, but with the number of successful entries per month. Also, there are missing 2 doors that have a different log file format, which is not supported by the script.

If employees want to get the correct analysis results, they should use FoxSec Access 1850 Version 17.2 application or Insight Review V5 application, depend on the log file format. Both applications are quite old and do not always give the correct result, that complicates the analyse processes.

4.5 User opinions and issues

According to user opinions of the above-listed applications are not suitable for a good analysis of the log files because errors occur in the results. Due to the fact that the applications have not been updated for a long time, they do not have a good interface and design, as well as the necessary functionality. Users want to get a full-fledged application that could work with all the log files that the company uses, as well as get the correct analysis results. Such an application would be able to facilitate and speed up the work of users.

5 Requirements

This chapter deals with functional and non-functional requirements for the Transcom Eesti OÜ application being developed.

5.1 Functional requirements

In consequence of the analysis of the existing applications and interviews with the client, was determined the main functional requirements for the future developed application [24].

User Story:

- As a customer, I want to be able to see the correct result of the analysis.
- As a customer, I want to be able to specify rules which are used to parse the log files.
- As a customer, I want to be able to parse the log files using specified rules.
- As a customer, I want to work with data from multiple files at the same time.
- As a customer, I want to see the results of parsing and sort them by timestamp, door ID, card id and event types.
- As a customer, I want to be able to filter the results by time range, door ID and card ID.
- As a customer, I want to be able to save the results that I see as a CSV file.
- As a customer, I want to be able to save summary usage statistics for the results that I see as a CSV file.
- As a customer, I want to be able to clear the application state without having to restart the application.

5.2 Non-functional requirements

As a consequence of the analysis of the FoxSec Access 1850 Version 17.2 application, Insight Review V5 application, web Excel script, were defined main non-functionality requirements [25].

Table 3. Non-functional requirements.

Category	Non-functional requirements
Security	Customer should not worry about data theft, because the application only works with files that the customer loads.
Interface	The application should have a good design, menu, understandable table with the month overview.
Reliability	The application should be accessible, restored after errors and should not crash.
Performance	The application should be oriented to a standard office computer.
Portability	The application should run on any computer operating system.
Scalability	The ability of the application to handle growing volumes of work.
Maintainability	The application degree where the application is repaired or improved.
Reusability	The ability to use existing assets in one form or another in the process of developing a software product.
Flexibility	The ability of the solution to adapt to possible or future changes in its requirements.

6 Proposed solution

In this chapter would be shown developments of the author's work. The chapter would be present all the information about the proposed solutions and detailed technical explanation of the developed application workflow logic.

6.1 Classification of log entries

There are many types of events. Some of them can be grouped together because logically they are the same for the people who are monitoring the log files. Some events should be ignored because they are not interesting at all. Flexible system of log entry type identification needs to be implemented. In order to be useful for the client after even long time and system changes as well as hardware changes, also changes the door usage policy across the company, adding the new type event classes should be possible. This way the user can specify the regular expression that corresponds to whatever rule they need to monitor. The traditional approach considers event types like successful access, denial of access, door malfunction and possible door abuse. If the user wants to, the rules and events can be as generic or as specific as the user needs.

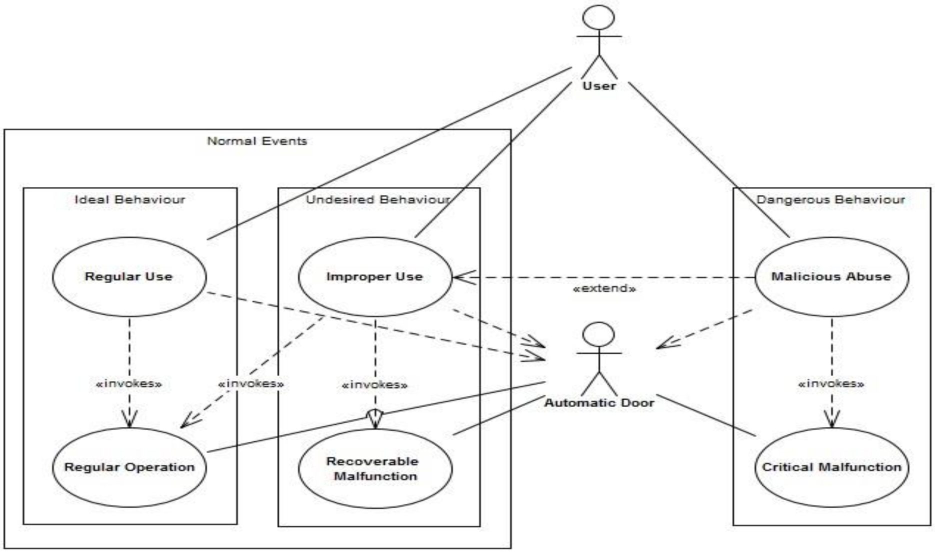
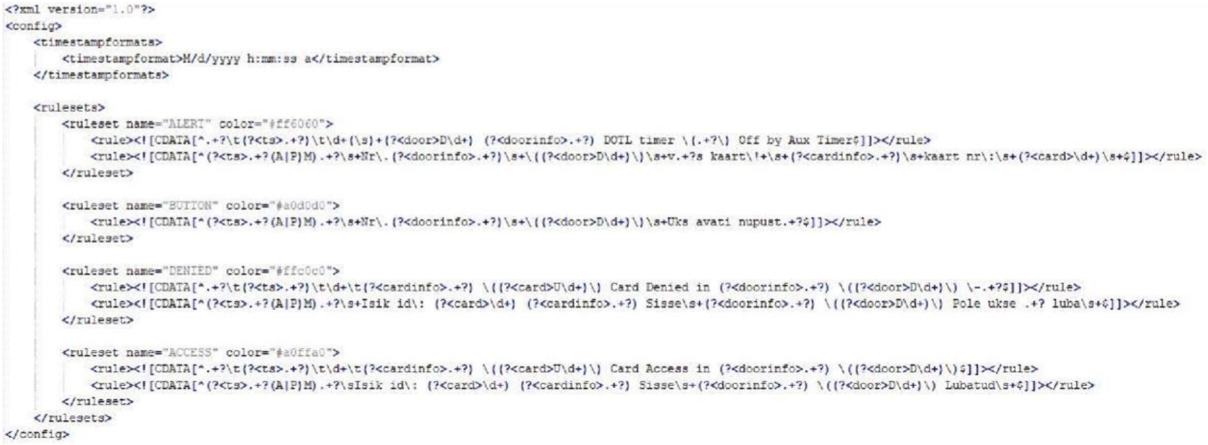


Figure 8. Possible classification of door usage events and behaviours.

It can be seen in Figure 8 that there are many possible types of events and many rules that will depend greatly on the needs of the client. The tracked behaviour may be very complex and therefore summary statistics produced by the proposed solution can be useful to analyse the data in later stages, which will be done outside the scope of this thesis.

6.2 Log parsing rules

Standard regular expressions are used as rules to parse the log files. Event type can be represented by one or more rules. There can be as many event types as needed. The rules are saved as an XML file. This XML file is read in real-time by the developed application and can be edited at any time. If the structure of the log files changes, then the XML would help to configure the workflow logic of the developed application. That allows not to redo the developed application code. The configuration also contains all possible timestamp formats that might occur in the log files.



```
<?xml version="1.0"?>
<config>
    <timestampformats>
        <timestampformat>M/d/yyyy h:mm:ss a</timestampformat>
    </timestampformats>

    <rulesets>
        <ruleset name="ALERT" color="#ff6060">
            <rule><![CDATA[^.+?\t(?<ts>.+?)\t\d+(\s)+(?<door>D\d+) (?<doorinfo>.+?) DOTL timer \(.+?\) Off by Aux Timer$]]></rule>
            <rule><![CDATA[^(?<ts>.+?(A|P)M).+?\s+Nr\.(?<doorinfo>.+?)\s+\((?<door>D\d+)\)\s+v.+?s kaart\!+\s+(?<cardinfo>.+?)\s+kaart nr\:\s+(?<card>\d+)\s+$]]></rule>
        </ruleset>

        <ruleset name="BUTTON" color="#a0d0d0">
            <rule><![CDATA[^(?<ts>.+?(A|P)M).+?\s+Nr\.(?<doorinfo>.+?)\s+\((?<door>D\d+)\)\s+Uks avati nupust.+?$]]></rule>
        </ruleset>

        <ruleset name="DENIED" color="#ffc0c0">
            <rule><![CDATA[^.+?\t(?<ts>.+?)\t\d+\t(?<cardinfo>.+?) \((?<card>U\d+)\) Card Denied in (?<doorinfo>.+?) \((?<door>D\d+)\) \-.+?$]]></rule>
            <rule><![CDATA[^(?<ts>.+?(A|P)M).+?\s+Isik id\: (?<card>\d+) (?<cardinfo>.+?) Sisse\s+(?<doorinfo>.+?) \((?<door>D\d+)\) Pole ukse .+? luba\s+$]]></rule>
        </ruleset>

        <ruleset name="ACCESS" color="#a0ffa0">
            <rule><![CDATA[^.+?\t(?<ts>.+?)\t\d+\t(?<cardinfo>.+?) \((?<card>U\d+)\) Card Access in (?<doorinfo>.+?) \((?<door>D\d+)\)$]]></rule>
            <rule><![CDATA[^(?<ts>.+?(A|P)M).+?\sIsik id\: (?<card>\d+) (?<cardinfo>.+?) Sisse\s+(?<doorinfo>.+?) \((?<door>D\d+)\) Lubatud\s+$]]></rule>
        </ruleset>
    </rulesets>
</config>
```

Figure 9. The XML file for the log files format.

6.3 Developed application system requirements

In order to run the application on the computer must be installed Java SE Runtime Environment which ensures the work of the developed application, even when the employees of the company have problems with the Internet connection. The application should be convenient and easy in use.

6.4 Application logic

The application follows the classical Model-View-Controller architecture. The model holds all of the data, the view is manipulated by the program user, the controller responds to user input and updates the model, which is reflected in the view. The class diagram of the application can be seen in Figure 10.

The application relies on JavaFX internal component functionality to provide basic sorting of the table view, but uses custom filters to allow selecting different date and time ranges (restricted by the earliest and latest timestamps as seen in the processed log files), as well as focus on entries that are related to specific doors or cards. Door and card IDs and names are automatically extracted from the logs. The application also makes it possible to export processed information in CSV format for further analysis using dedicated data tools. There are three export options: filtered log output, where all irrelevant entries are removed; door usage statistics, which provides a day-by-day breakdown of the number of different events associated with each door over the selected period; card usage statistics, which provides a day-by-day breakdown of the number of different events triggered by each card over the selected period. This functionality is the most important part of the application and was explicitly requested by the client, because currently used solutions are not making it easy.

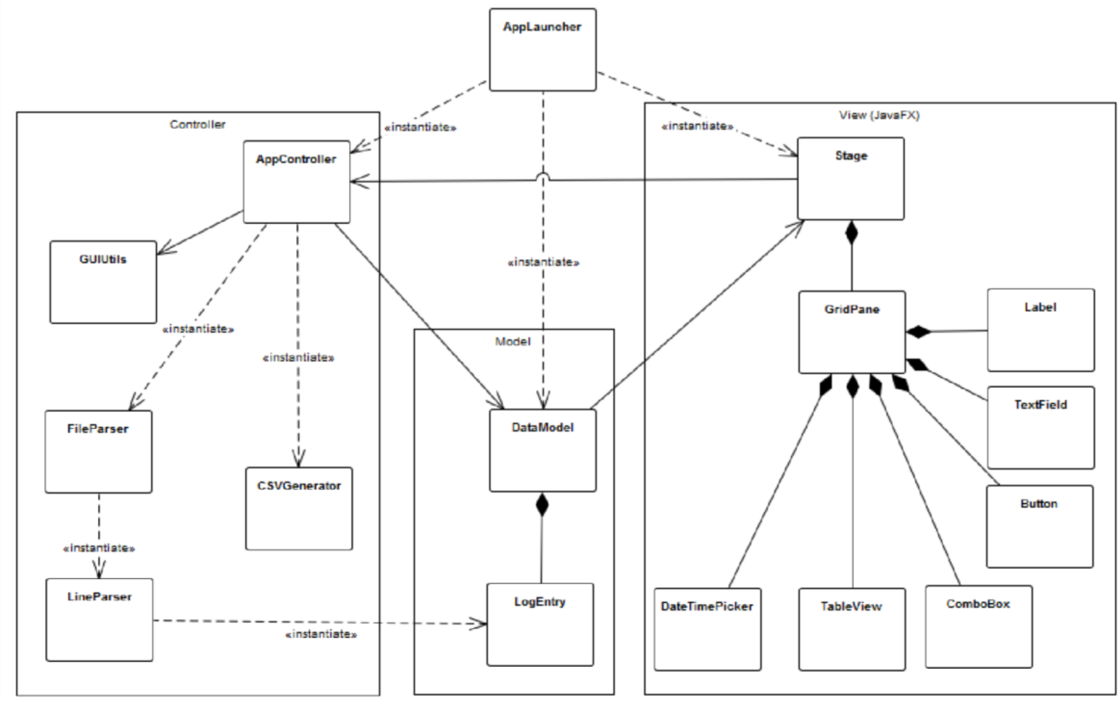
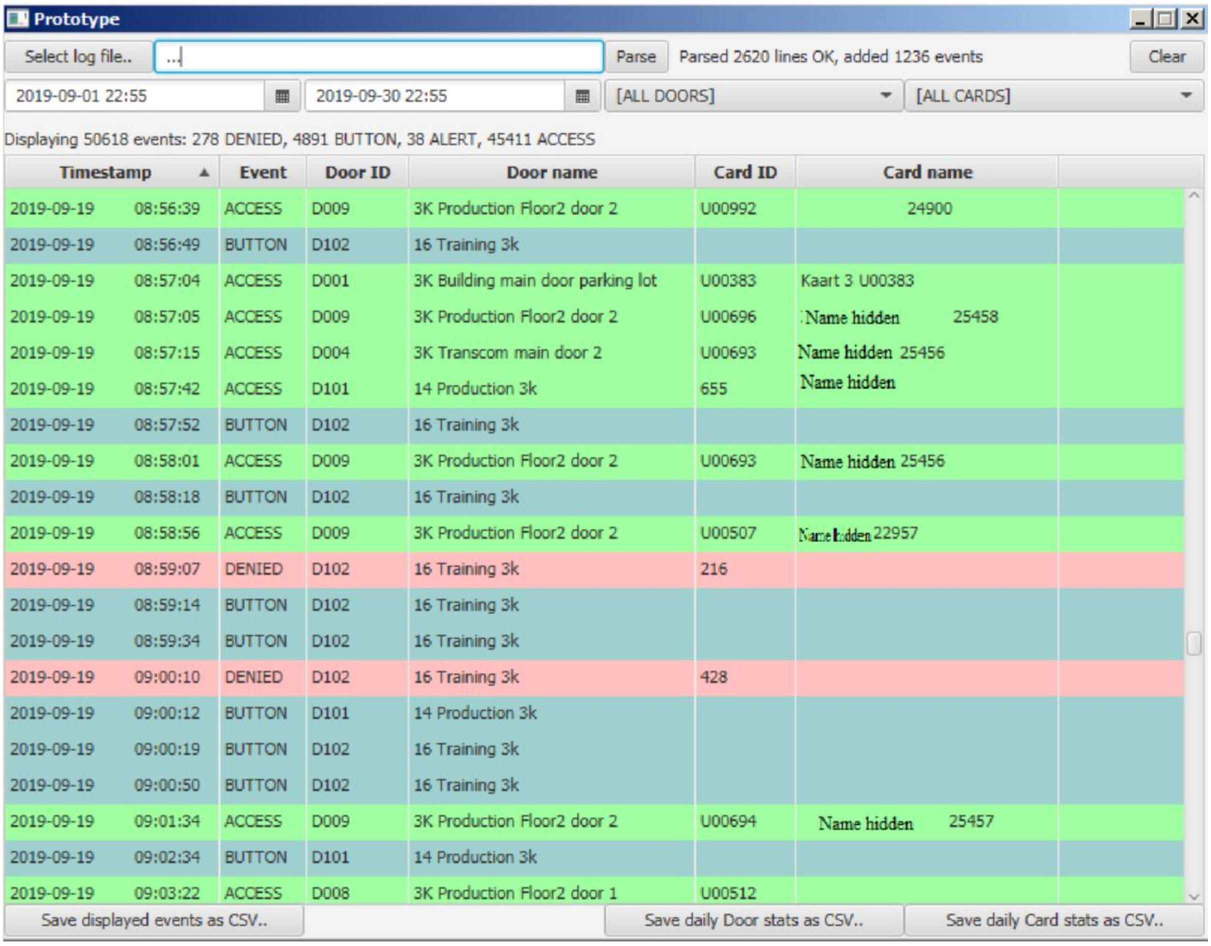


Figure 10. Class Diagram of the developed application.

7 Graphical user interface design

The graphical interface uses standard operating system components as driven by JavaFX, which makes it recognizable and comfortable to understand. The most important part for the analysis is the ability to colour code specific event types in order for the user to see them without having to read through the actual text details of the log. The number of interactive components was kept to a minimum in order not to overcomplicate the user interface - something that is clearly the problem with existing solutions. The application responds to all user input immediately once all of the data is loaded, although the initial parsing log files may take some time (the user is very well aware that parsing big log files takes some time).


Prototype
Select log file..
...
Parse
Clear
Save displayed events as CSV..
Save daily Door stats as CSV..
Save daily Card stats as CSV..

Parsed 2620 lines OK, added 1236 events

2019-09-01 22:55 | 2019-09-30 22:55 | [ALL DOORS] | [ALL CARDS]

Displaying 50618 events: 278 DENIED, 4891 BUTTON, 38 ALERT, 45411 ACCESS

Timestamp ▲	Event	Door ID	Door name	Card ID	Card name
2019-09-19 08:56:39	ACCESS	D009	3K Production Floor2 door 2	U00992	24900
2019-09-19 08:56:49	BUTTON	D102	16 Training 3k		
2019-09-19 08:57:04	ACCESS	D001	3K Building main door parking lot	U00383	Kaart 3 U00383
2019-09-19 08:57:05	ACCESS	D009	3K Production Floor2 door 2	U00696	Name hidden 25458
2019-09-19 08:57:15	ACCESS	D004	3K Transcom main door 2	U00693	Name hidden 25456
2019-09-19 08:57:42	ACCESS	D101	14 Production 3k	655	Name hidden
2019-09-19 08:57:52	BUTTON	D102	16 Training 3k		
2019-09-19 08:58:01	ACCESS	D009	3K Production Floor2 door 2	U00693	Name hidden 25456
2019-09-19 08:58:18	BUTTON	D102	16 Training 3k		
2019-09-19 08:58:56	ACCESS	D009	3K Production Floor2 door 2	U00507	Name hidden 22957
2019-09-19 08:59:07	DENIED	D102	16 Training 3k	216	
2019-09-19 08:59:14	BUTTON	D102	16 Training 3k		
2019-09-19 08:59:34	BUTTON	D102	16 Training 3k		
2019-09-19 09:00:10	DENIED	D102	16 Training 3k	428	
2019-09-19 09:00:12	BUTTON	D101	14 Production 3k		
2019-09-19 09:00:19	BUTTON	D102	16 Training 3k		
2019-09-19 09:00:50	BUTTON	D102	16 Training 3k		
2019-09-19 09:01:34	ACCESS	D009	3K Production Floor2 door 2	U00694	Name hidden 25457
2019-09-19 09:02:34	BUTTON	D101	14 Production 3k		
2019-09-19 09:03:22	ACCESS	D008	3K Production Floor2 door 1	U00512	

Figure 11. Design of the developed application.

8 Evaluation and analysis

Overall, the process of gathering requirements, analysing existing solutions and implementing a solution that would fit the client requirements was a success. However, the resulting application is only one part of the upcoming door usage analysis toolset that the client has ambitions for, so it is somewhat restricted in scope and functionality, while trying to stay as generic as possible to be useful to any user who is facing a similar problem.

8.1 Results

In the form of the result of this work, the author analysed the applications used by the company, also researched the developers web pages, researched the client requirements, created the UML diagram of the whole process, defined the functional and non-functional requirements, created the plan for action and development processes, developed the full-fledge computer application and design, tested acceptance tests, received positive feedback.

8.2 Client feedback

All acceptance tests were passed successfully. All wishes and requirements of the company were satisfied.

The user has acquainted with the interface of the developed application, used all the functionality of the application, uploaded several log files with different format and analysed them, saw how many doors and workers were contained in the log files, also filtered them by the date, alarms, time, door ID and other parameters, employee name, saved analysis to the CSV files and opened them in Excel, cleared the application analysis and started the new one without closing the application and etc.

The client is satisfied with the information which was received during the analysis of the log files, it is convenient to work with such an application. The developed application is supported on all office computers since Java runtime environment is installed on them, the new interface of the application did not pose any problems, if the client does not know how to use regular expressions, adding their own rules is not a problem.

The following is a summary of the client feedback: "The purpose of this job was to get an application as an alternative solution for door log parsing because the original system does not provide such a feature. The application was created and tested. All requested features are in place and working fine. There are no concerns about application or developer.

8.3 Overall performance

The application is available for use on any computer where is installed Java Runtime environment. In cases of the problems with the Internet connection, employees would be able to use this application, because this application refers to a stand-alone type of application that does not need an Internet connection. The application is convenient and easy in use.

8.4 Area for improvement and future work

If the company would decide that the application requires an update or additional functionality, developers can easily do it by exploring the documentation and code with developed application. The author recommends add the new filters that could improve analysis of the log files, also could be developed the algorithm, which counts the number of people in certain rooms in a certain period of time. Such algorithm would be very useful for people tracking in the rooms, but it could take a lot of time to implement it to the application and it is unlikely that it would be brought to perfection, since some doors could be opened using a button instead of a card, which would lead to misleading in the analysis logic, thus the calculations would not be accurate.

9 Summary

The aim of the work was to provide to the Transcom Eesti OÜ company an alternative solution which would improve the analysis of the log files. All the goals of the work have been achieved. Before starting to develop the application, the author conducted a survey and separate interviews with the IT department of the company during the development of the application, which helped to identify the main problems. All requirements for the developed application were created. The whole process was compiled in the form of the UML diagram. The applications which the company used before were analysed. The Java application types were researched. The functional requirements were described as use cases and non-functional requirements with the quality tables. The main problems of the existing applications were: security, interface, reliability, performance, portability, scalability, maintainability, reusability, flexibility. It was decided to develop a stand-alone application written in Java based on the experience of development, also it was decided to use the Iterative waterfall development model. With this choice of application development, the author was able to satisfy all client requests, as well as effectively analyse, design, implement, test and deploy the developed application, also with such development model the work plan was flexible, which helped to speed up the development process. In conclusion, the main objective of this thesis has been successfully accomplished. The application was successfully developed. The unit tests were written and the application was tested. User acceptance tests passed successfully and the application approved by the client.

References

[1] Java - Regular Expressions. [Online] https://www.tutorialspoint.com/java/java_regular_expressions.htm - (Accessed 10.10.2019)

[2] What is a Software Process Model? [Online] https://www.visual-paradigm.com/guide/software-development-process/what-is-a-softwareprocess-model/ - (Accessed 16.10.2019)

[3] Software development lifecycle models. [Online] https://www.researchgate.net/publication/220631422_Software_development_lifecycle_models - (Accessed 16.10.2019)

[4] Software Engineering | Classification of Software Requirements. [Online] https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/ - (Accessed 20.10.2019)

[5] ISO/IEC 25010. [Online] https://iso25000.com/index.php/en/iso-25000-standards/iso-25010 - (Accessed 21.10.2019)

[6] Design Patterns - MVC Pattern. [Online] https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm - (Accessed 27.10.2019)

[7] The Good and the Bad of Java Programming. [Online] https://www.altexsoft.com/blog/engineering/pros-and-cons-of-java-programming/ - (Accessed 28.10.2019)

[8] What Are the Advantages of a Standalone Computer? [Online] https://www.reference.com/world-view/advantages-standalone-computer-c30ab51db5ce2560 - (Accessed 28.10.2019)

[9] Applets: Advantages & Disadvantages. [Online] https://study.com/academy/lesson/applets-advantages-disadvantages.html - (Accessed 28.10.2019)

[10] Advantages of Java Servlets. [Online] https://medium.com/@javatutorialpoint0/advantages-of-java-servlets-8d6a2f09a08d - (Accessed 28.10.2019)

[11] Types of Java Programs. [Online] http://ecomputernotes.com/java/what-is-java-language/types-of-java-programs - (Accessed 28.03.2018)

[12] User Interface Design in Modern Web Applications. [Online] https://www.smashingmagazine.com/user-interface-design-in-modern-web-applications/ -

(Accessed 28.10.2019)

[13] User Interface Design For Programmers. [Online] https://www.joelonsoftware.com/2001/10/24/user-interface-design-for-programmers/ - (Accessed 10.11.2019)

[14] In Defense of Eye Candy. [Online] https://alistapart.com/article/indefenseofeyecandy/ - (Accessed 10.11.2019)

[15] Managing UI Complexity. [Online] http://www.brandonwalkin.com/blog/2009/08/10/managing-ui-complexity/ - (Accessed 10.11.2019)

[16] First Principles of Interaction Design (Revised & Expanded). [Online] https://asktog.com/atc/principles-of-interaction-design/ - (Accessed 12.11.2019)

[17] 10 Usability Heuristics for User Interface Design. [Online] https://www.nngroup.com/articles/ten-usability-heuristics/ - (Accessed 12.11.2019)

[18] How to Write Good Error Messages. [Online] https://uxplanet.org/how-to-write-good-error-messages-858e4551cd4 - (Accessed 12.11.2019)

[19] What Is User Acceptance Testing [Online] https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/ - (Accessed 19.11.2019)

[20] Pristis. [Online] https://www.pristis.ee/en/low-voltage-and-security-solutions - (Accessed 19.10.2019)

[21] NVP access control and alarm systems. [Online] https://www.nvp.ee/ - (Accessed 23.10.2019)

[22] FoxSec access control and alarm systems. [Online] https://www.foxsec.eu/et/tooted/foxsec-1850/ - (Accessed 17.10.2019) [23] Insight management software for concept 4000

Security/access Systems. [Online] http://assetcloud.innerrange.com/ProductFiles/4449-630053_Insight_V5_User_Manual.pdf - (Accessed 17.10.2019)

[24] Alistair Cockburn, Writing Effective Use Cases, 2000.

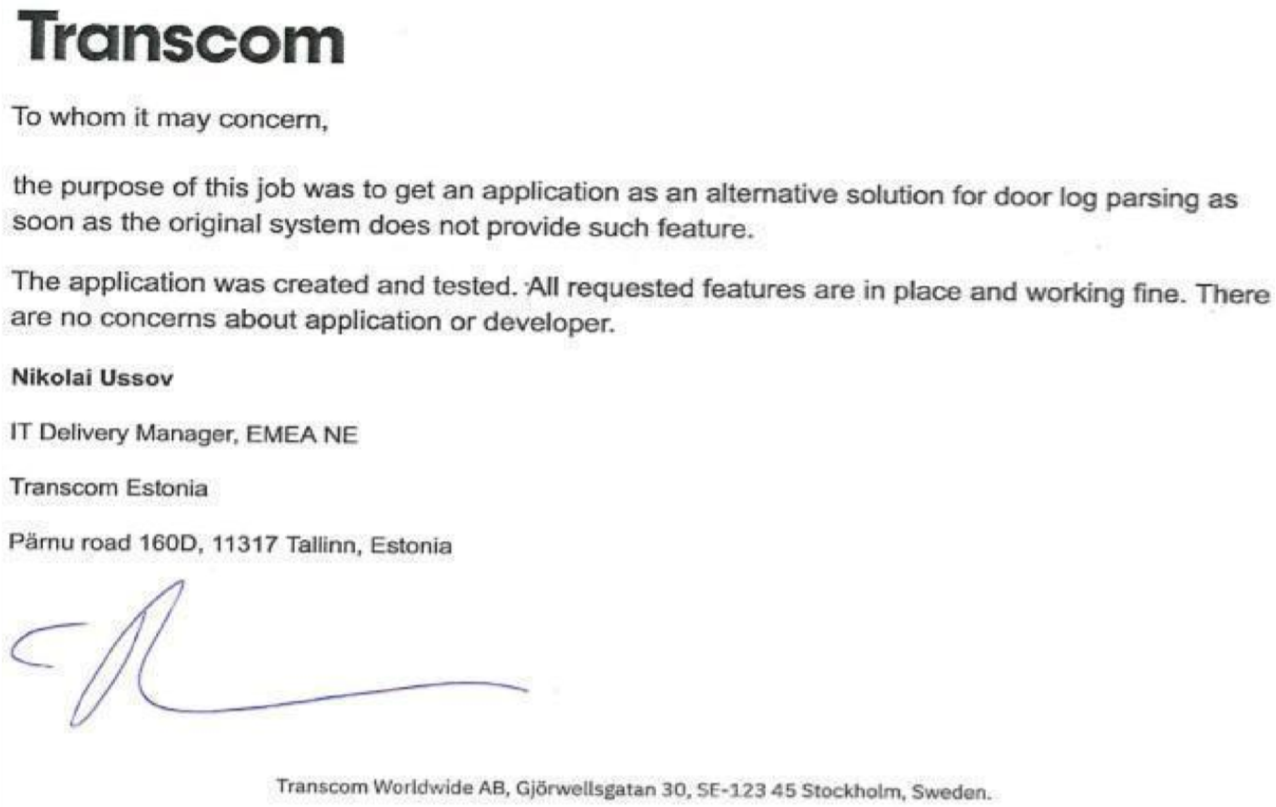
[25] What is Non-Functional Requirement? Types and Examples. [Online] https://www.guru99.com/non-functional-requirement-type-example.html - (Accessed 29.11.2019)

[26] Java™ Platform, Standard Edition 8 API Specification. [Online] https://docs.oracle.com/javase/8/docs/api/ - - (Accessed 23.10.2019)

[27] JUnit 4.13-beta-1 API. [Online] https://junit.org/junit4/javadoc/latest/index.html - (Accessed 25.11.2019)

[28] Why Bother to Unit Test? . [Online] https://www.qasystems.com/fileadmin/user_upload/resources/White_Papers/Why_Bother_to_Un it_Test.pdf - (Accessed 05.12.2019)

Appendix 1 – Client feedback.



Transcom

To whom it may concern,

the purpose of this job was to get an application as an alternative solution for door log parsing as soon as the original system does not provide such feature.

The application was created and tested. All requested features are in place and working fine. There are no concerns about application or developer.

Nikolai Ussov

IT Delivery Manager, EMEA NE

Transcom Estonia

Pärnu road 160D, 11317 Tallinn, Estonia

Transcom Worldwide AB, Gjörwellsgatan 30, SE-123 45 Stockholm, Sweden.

Appendix 1. Signed summary of client feedback.

Appendix 2 - Link to application files.

- Application files link on github: https://github.com/VadimKoop/Portfolio/tree/master/Application%20for%20door%20security%20log%20analysis%20(Java)