Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Computer Science

PRIVACY MANAGEMENT SYSTEM FOR SPECIFIC USER GROUPS

Alexandru-Victor Macocian

Supervisor: doc. Ing. Zdeněk Míkovec, Ph.D.

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Macocian Alexandru-Victor**	Personal ID number:	**473277**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Control Engineering**		
Study program:	**Open Informatics**		
Branch of study:	**Computer Engineering**		

II. Master’s thesis details

Master’s thesis title in English:

Privacy Management System for Specific User Groups Using ICT

Master’s thesis title in Czech:

Systém pro správu soukromí při používání ICT pro specifické cílové skupiny

Guidelines:

Conduct user research focused on privacy issues while using information and communication technologies (ICT) of specific user groups. Identify the user groups and specify their privacy concerns and potential threats connected to activities related to the usage of ICT. Based on this research create a concept of the future system helping specific user groups with privacy management and design possible solutions. By means of iterative user testing of low fidelity prototypes create the final design (follow the UCD methodology). Based on the final design implement a high-level prototype that will serve as a proof-of-concept by means of conducting research experiment.

Bibliography / sources:

[1] Shepherd, Lynsay A., and Jacqueline Archibald: "Security awareness and affective feedback: Categorical behaviour vs. reported behaviour." In Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA), 2017 International Conference On, pp. 1-6. IEEE, 2017.
[2] Goodman, Elizabeth, Mike Kuniavsky, and Andrea Moed: "Observing the user experience: A practitioner's guide to user research." IEEE Transactions on Professional Communication 56, no. 3 (2013): 260-261.
[3] Cooper, Alan, Robert Reimann, David Cronin, and Christopher Noessel: About face: the essentials of interaction design. John Wiley & Sons, 2014.

Name and workplace of master’s thesis supervisor:

doc. Ing. Zdeněk Míkovec, Ph.D., Department of Computer Graphics and Interaction

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **14.02.2019** Deadline for master's thesis submission: ___________

Assignment valid until:
by the end of summer semester 2019/2020

doc. Ing. Zdeněk Míkovec, Ph.D. Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt Student’s signature

DECLARATION OF AUTHENTICITY

I declare that I have completed this thesis independently and I have cited all used sources in accordance to Methodical instruction n. 1/2009 about ethical principles for academic thesis writing.

Prague, May 2019

…………………………………………….

ABSTRACT

Recent developments in internet and social media usage have shown that the time of browsing the internet while maintaining anonymity is slowly coming to an end. In special, the recent scandals in regards to how internet organizations maintain private identifiable data have shown that there's a need for some solutions to prevent users from being compromised while using a computer and using the services provided by the internet.

Special attention is put on older people and their accessibility to computers as studies have shown that older people are on average less experienced in using a computer and are subject to greater risks.

This work tries to provide a framework for further development that would eventually solve most vulnerabilities that our target group have. The main idea of this project is to provide an enclosed environment that emulates most of the features popular environment have while automating most processes that were not crucial to be handled by the person. The goal of this project is to have a working system that emulates popular desktop environment such as users feel familiar while keeping them as safe as possible from common vulnerabilities.

Contents

Table of figures

1 INTRODUCTION

According to [1], there are currently more than 4 billion people with access to the internet. These statistics show a general growth in the number of people with access to the internet, with no sign of stopping. Research done by the Pew Research Center shows that while almost all adults and teenagers already have access to the internet, the number of older people just recently started to rise and is expected to continue its upward trend [2].

Most of the people who have recently gained access and that have yet to gain access are not going to be proficient in computer usage. The OECD researchers have created a table to classify the proficiency levels of the adult population [3]. This document and related work are targeted towards the lower 40% of the population who can barely (if at all) use a computer for trivial tasks. Some elements of the developed application can be used by advanced users (or users that would be able to normally navigate the internet), but our main target group are the people in the lowest levels of computer usage proficiency.

The goal of this work is to implement the basis (or rather a framework) that will help people with low computer proficiency and people with disabilities that may impair their computer usage use a computer and have some basic access to the internet while maintaining their privacy and security as safe as possible without limiting too much their possibilities.

2 PROBLEM DESCRIPTION

In its conception, the computer and later the world wide web was developed by experienced people with strong academical backgrounds and they were designed as such. The internet provides many possibilities, but for many people, it is like an entirely different world with different rules. While the experienced people can enjoy the freedom and comfort it brings, for many people, the vast amount of information it provides and requires to be able to use properly is too big of a burden.

The purpose of this work is to provide a solution to the unexperienced people in our target group and to be used as an introductory tool to computer usage. As such, we will define the target group of this work as older people, with accessibility issues and unexperienced in using computers.

2.1 Computer usage in older adults

There is a significant number of older people who have access to computers. While there is no conclusive data or not generalized data, due to the sample size and the nature of the research, “57% of all respondents said that they had access to a computer” [4] from the respondents to the cited survey. Additionally, “43% of participants owned a computer” [4]. Some more data that supports our case that a significant amount of older people is not experienced with computers is that from the people surveyed in [4], only 28% managed to provide some answer, not necessarily an adequate one.

Regarding the actual usage of computers, “The most popular category was the Internet or some use of the Internet such as information access, research or shopping” [4], with close to 40% of the participants that use the computer, use it to access the internet “frequently” and almost 80% use it to access the internet “occasionally”.

While the difficulties that these people face when using computers may vary significantly, one common difficulty found amongst the people surveyed in [4] was the overly complex nature of computers and the large amount of information needed to be processed to do even the most basic of things.

Another thing to note is that amongst the older people and the data specified previously, most certainly that a significant portion of the people who use the computer “frequently” are already experienced and do not require additional help in order to fulfill their daily tasks. As such, they are not part of our target group.

Finally, as [4] concludes, our target group has “a need for simpler applications and documentation, designed with older people in mind, as well as for greater support” [4].

2.2 Main issues related to the target group

As of 2018, 64% of people with ages between 50 and 64 years old and 37% of people with ages 65 and above frequently use at least one social media site [5]. Also, according to [1], as of 2014, the percentage of older people that use social networks ranges in between 3% up to 9% of the total number of users of each social network. As the number of social network users has risen in the last 5 years, it is safe to assume that the number of older people using the social networks has also seen an improvement. As such, the actual number of people using social networks that fit into our target group makes up quite a significant number of people.

The largest issues that relate to our target group and that we aim to solve are the following: privacy, authorization and misuse. The privacy issue refers to the disclosure of too much personal information without understanding how that information will be handled. The authorization issue refers to how do the users authorize the usage of their information and how do the users understand the consequences of their authorization. The last issue is related to the misuse of computers and internet access, how can the users be taught to use computers and internet access, what can they expect from what they choose to use, how should they use these technologies as safely as possible.

The privacy issue is the largest one and the most popular one currently, as social networks (in particular Facebook) have recently faced numerous privacy violations and data misuse accusations. These issues present such a real threat to the users of social media that governments have been forced to act and provide solutions. As such, the General Data Protection Regulation has been enacted on 25 May 2018, proving that these concerns are real and that they must be tackled.

Another large issue more specific to our target group is accessibility. According to [6] 86.3% of blind people are older than 50 years old and 52.8% are older than 70 years. Also, according to [7], 20% of adults aged 60 or over suffer from some form of mental disorder. As such, a large percentage of our target group has a need for accessibility features and simplified interfaces.

2.3 Providing an approach to the problem

The approach to solving the issues presented in 2.2 - Main issues related to the target group is to provide a curated environment, where the target group can learn how to use a computer and how to browse the internet, while reducing the surface of attack in a number of ways.

By providing an encapsulated and curated environment, popular mistakes such as downloading and executing potentially dangerous applications as well as running unwanted and dangerous code on the user machine can be managed, and in some cases removed.

Also, using by using this environment, we can include tutorials and guide the user on their activity, training and teaching healthy and safe behaviors in computer and internet usage. The users can be taught how to understand when their information is requested, what to share and when to refuse sharing information.

A final part to the approach discussed in this document is including some accessibility features with the goal in aiding our target group in understanding and using the computer. These features should lead to lower fatigue from computer usage and make learning and understanding easier for the target group that is older and as a consequence, harder to teach.

3 STATE OF THE ART

This chapter focuses on the current solutions to the tackled problem, as well as suggestions and guidelines.

3.1 Popular accessibility features

3.1.1 TOUCH-SCREEN DEVICES

One current solution to accessibility problems is the usage of tablets and phones with big screens that support large font sizes and touch-screen methods of usage.

Figure 3.1 - Tablet with touch-screen support

This is currently the most popular solution as the touch-screen method of operation is very intuitive and doesn't require learning how to use additional supportive devices as well as having very good hand to eye coordination.

Big buttons with clearly visible text and large font sizes make it much easier for the user to utilize the interface.

3.1.2 TABLET MODE IN POPULAR OPERATING SYSTEMS

While previously we mentioned the hardware solution to the current problem, this sub-chapter presents solutions provided by the developers of the popular operating systems.

The latest versions of operating systems provide tablet mode interfaces with large icons, easily distinguishable text and fonts to aid visually-impaired users.

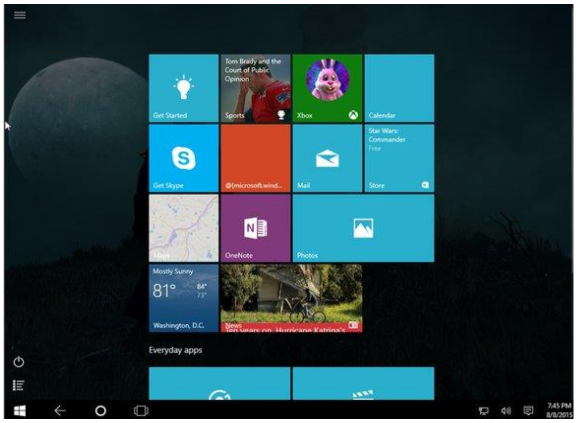


Figure 3.2 - Windows 10 Tablet Mode

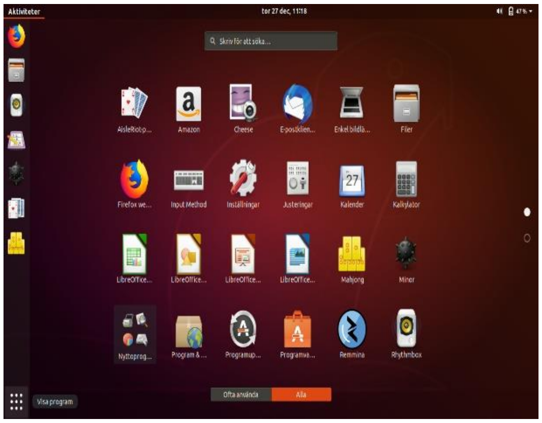
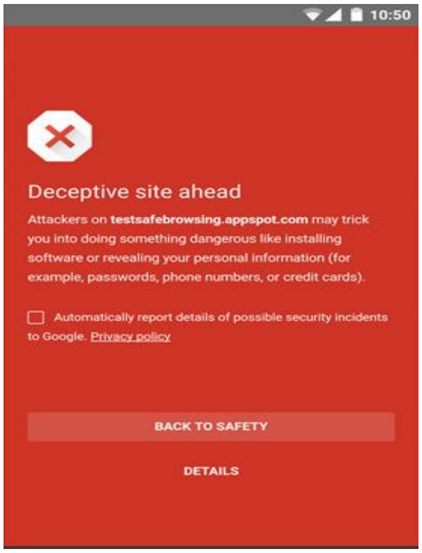


Figure 3.3 - Ubuntu with Gnome Desktop Environment

3.2 Safety and privacy features

3.2.1 GOOGLE SAFE BROWSING

To aid the people with lack of experience in computer usage, Google has developed an API designed to protect the users by showing warnings when they attempt to navigate to dangerous sites or downloading dangerous files.



10:50

Deceptive site ahead

Attackers on **testsafebrowsing.appspot.com** may trick you into doing something dangerous like installing software or revealing your personal information (for example, passwords, phone numbers, or credit cards).

□ Automatically report details of possible security incidents to Google. Privacy policy

BACK TO SAFETY

DETAILS

Figure 3.4 - Google Safe Browsing

3.2.2 GOOGLE PASSWORDS

For password storage, Google Chrome has embedded the API for Google Passwords, a cloud-based storage for personal information. Once a user has logged in onto a website, Chrome will ask the user to save the log in information onto their servers for easier re-use.

3.2.3 MESSAGING APPS WITH END-TO-END ENCRYPTION

A solution to safe messaging is to use an application which makes use of end-to-end encryption between the peers in a conversation. Applications like WhatsApp, Viber and Telegram encrypt the user communication while being transmitted over the internet, preventing malicious actors from gathering that communication and reading it. WhatsApp has become especially popular due to users having to authenticate using their real phone number and as a result, people were able to connect with their friends using the already stored contact information in their phones.

3.2.4 EMAIL FILTERS

Another issue that especially plagues inexperienced and older people is scam and phishing emails. Most usually, these emails are sent to a large amount of people with the goal that a few of them would be inexperienced enough to fall for the tricks and scams contained inside the email. One solution to this problem is using email filters that automatically detect and remove unwanted and potentially dangerous email from reaching the inbox of the user. Most email service providers, such as Google, Yahoo, Microsoft, etc., have automatic filters that are enabled by default.

3.2.5 AD BLOCKERS

One popular solution to the ever-present and potentially dangerous ads and popups are the ad blockers. They usually come as extensions or plugins to browsers with the sole purpose of blocking advertisement. While in description, this doesn’t sound like an actual privacy concern, a significant amount of these ads is designed to target inexperience and/or lack of knowledge and make users access them, leading to potentially dangerous websites that might employ phishing or scamming techniques on the visitors. As such, ad blockers are an important part of the privacy protection and browsing safety of users, particularly of those in the designated target group.

3.2.6 TWO-FACTOR AUTHENTICATION

Two-factor authentication is a method of authentication that’s been gaining popularity for quite a bit of time. Modern techniques involve using mobile devices as a part of the authentication procedure. The method requires that after the user has provided the correct credentials, they must further pass through some authentication steps where the current answers are provided to the user dynamically through their mobile devices, totally separated from the previous medium of authentication. These methods have been successfully used in critical systems such as banking systems where a potentially unwanted or malicious actor gaining access to restricted information could result in significant monetary losses.

Nowadays, most important online services providers such as Microsoft, Google, Yahoo, Apple, etc., provide the ability to use two-factor authentication in addition to the usual authentication method of username and password combination.

4 DESIGN

This chapter focuses on the design directions taken during development as well as the challenges faced during design phase of the prototype.

4.1 Use cases and scenarios

The application's main use case is simplifying computer use to reduce the complexity of actions necessary for our target group. The goal is keeping the security and accessibility to the highest levels and automated while users would be left to use and enjoy the computer without worrying about a large group of vulnerabilities.

The application is designed to eventually allow almost all casual activities one person can do on a computer. Currently, the plans involve browsing the internet, locally monitoring the browsing activity and notifying users about potential vulnerabilities while browsing as well as possible solutions to those vulnerabilities.

In order to be able to allow the previously mentioned activities, user information must be stored under the form of user profiles.

4.1.1 USER PROFILES

The user profiles are supposed to contain stored log-in information from previous browsing, personal identifiable information and frequently requested information by the online services. In order to simplify maintaining privacy while browsing while still being able to access the online services that require personal information, the user should be able to switch these profiles.

User profiles would store information that would be auto-filled when logging into online services and switching profiles would result in switching the personal information, practically switching the online persona of the user. Thus, one person having one main user profile and a few alternative profiles would be able to switch them during runtime and be able to avoid identification by online services on demand.

4.1.2 SCENARIOS

The currently designed framework is created with browsing the internet as the main activity. Extensibility and customizability are an important thing right now as to allow further development.

Scenario 1: The user wants to browse the internet using random pages.

Scenario 2: The user wants to connect to a social network and use the provided services.

Scenario 3: The user with reading impairment wants to be able to use the computer without significant effort.

4.1.3 USE CASES

UC1: For scenario 1, the user authenticates with the system. The authentication steps can be observed in 9.1 - Application showcase. After authentication, the user is given access to the browser control and is able to use it same as using any other browser.

UC2: For scenario 2, the user authenticates with the system. After authentication, the user opens the browser control. The user accesses the desired social network webpage. The framework is parsing the accessed webpage and is detecting a log-in prompt. The framework prompts the user with a warning stating that personal information is required. The user is allowed to switch between profiles and choose what related auto-fill data could be entered. The user logs in using the data generated by the framework and uses the social network.

UC3: For scenario 3, the user first authenticates with the system. Then, the user goes to the centralized control panel of the framework and customizes the look of the framework to fit their needs. The framework then redraws all the controls respecting the user's customizations. An example can be seen in Figure 9.9 - Accessibility features showcase where the user has changed the color of the windows to black and the size of the font to twice the normal size.

4.2 High level concept

4.2.1 SYSTEM OVERVIEW

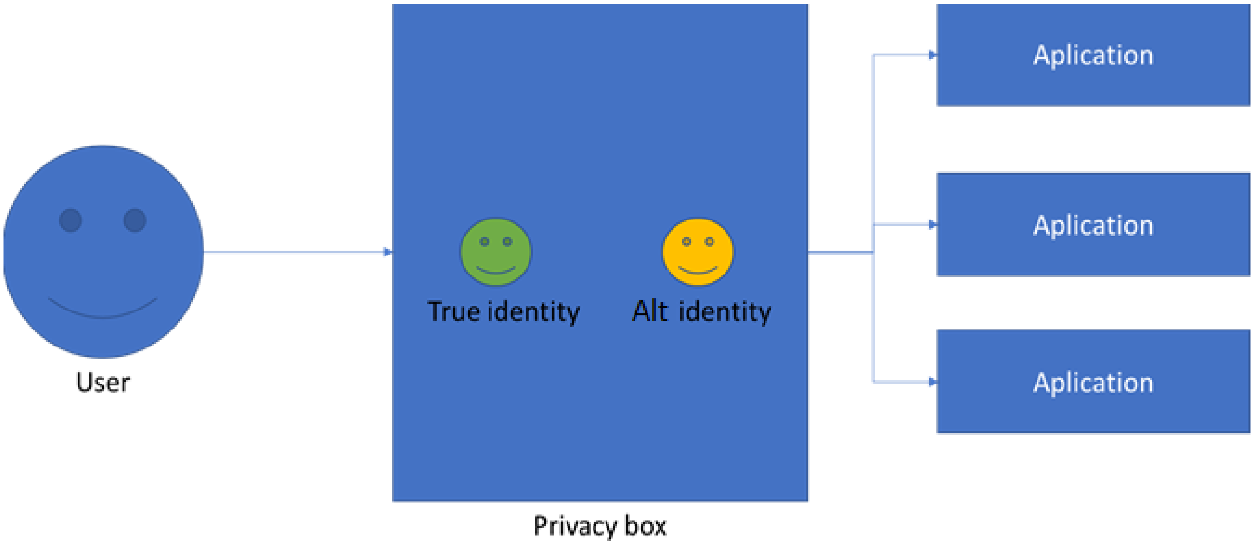


Figure 4.1 - High level concept

As a high-level concept, the application is supposed to be an intermediary between the user and the real-world applications. It serves as a man-in-the-middle, introducing several new concepts as true-identity and alternative-identity.

True-identity and alternative-identity are two concepts that play with the personal information of the user. As a large amount of applications require personal information in order to function properly, these concepts are supposed to allow masking the personal information of the user on demand, protecting the identity of the user while still maintaining accessibility to the desired applications.

4.2.2 KEY-BASED AUTHENTICATION

The system works based on USB flash drives authentication and storage. The whole user information and the settings are stored only locally on the USB flash drive and the system will only work if the user provides their flash drive.

Flash drive-based authentication introduces the “key metaphor”. The “key metaphor” draws a direct parallel between a raw physical key that could open the safe with their valuables and the flash stick that contains their information. Considering the experience and capability of the target group, using the “key metaphor” could make understanding the concept trivial for most people, as well as understanding the value and importance of their personal information. Having this “key” that can open their “safe” with information, the users will understand why protecting their privacy and data is an important factor in computer and internet usage.

4.2.3 USER INFORMATION MANAGEMENT

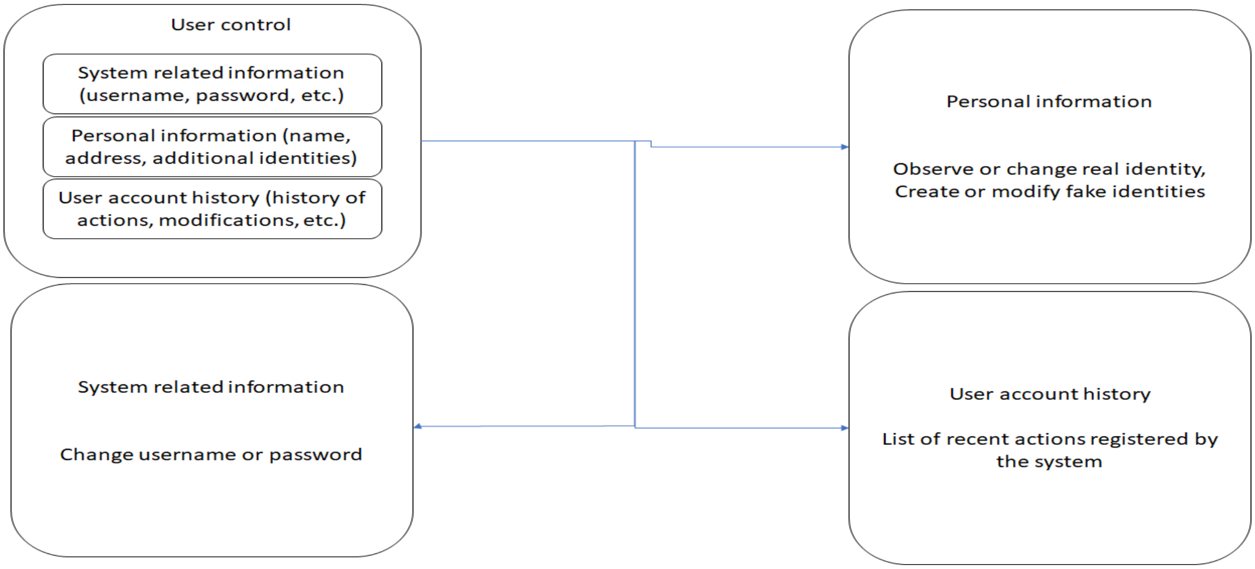

User control
System related information (username, password, etc.)
Personal information (name, address, additional identities)
User account history (history of actions, modifications, etc.)
Personal information
Observe or change real identity, Create or modify fake identities
System related information
Change username or password
User account history
List of recent actions registered by the system

Figure 4.2 – User control and management schema

The user is supposed to have absolute control of their information. Because of this, a centralized management system must be implemented, one that allows easy management of the application specific information and settings as well as data gathered on the user.

As the information is stored locally, any change should be reflected only for the current user and plugging the “key” into any computer running the developed system should reflect the changes and information contained by the management system.

4.3 Desktop environment design

4.3.1 DESKTOP DESIGN OVERVIEW

The desktop environment will contain two main areas. The main canvas area and the glyph area.

Due to developing the application for people with minimal computer experience, the goal was to emulate the existing desktop interfaces and to provide an experience as similar as possible.

The main canvas area serves as the main area for most of the operations. It contains all visual interfaces.

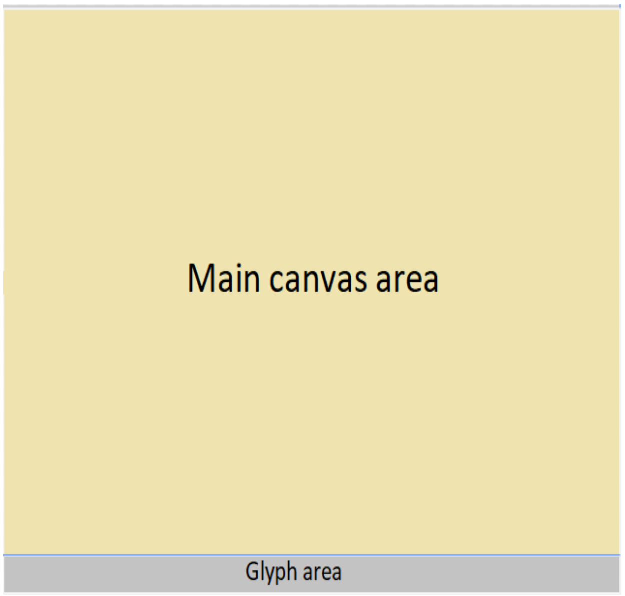


Figure 4.3 - Desktop main areas

The glyph area serves as the area for displaying the currently running applications' glyphs, making it easier to see what is currently running and making the action of switching in between windows easy.

4.3.2 GLYPH AREA DESIGN

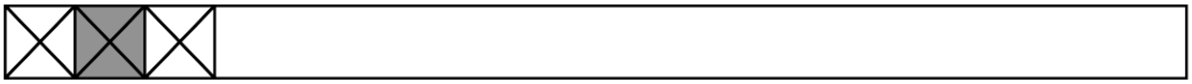


Figure 4.4 - Glyph area

To keep the interface simple and familiar, the following schematic is followed. The glyph area is made to display a series of glyphs, each glyph representing an application currently running. The differentiation between the which application is active and which is running in the background is done by overlaying the glyph associated with the currently running application with a semi-opaque overlay.

4.3.3 MAIN CANVAS AREA DESIGN

The main area is separated into a grid of dynamic size, where each application has its own glyph/icon to facilitate launching an application. The main canvas area also displays the canvases of the running applications and allows for common actions such as dragging and dropping or repositioning of the items on it.

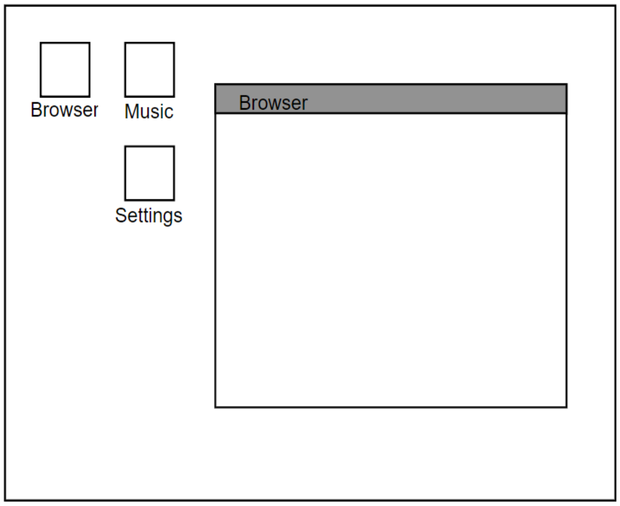


Figure 4.5 - Design of main canvas area

4.3.4 APPLICATION LAYOUT AND DESIGN

Each application has the displayed layout. The layout has a group of mandatory components and an area designated for the drawing and customization of the application-specific interface.

The grey area at the top is the title bar. This is a mandatory component. The title bar contains the following components: application icon, title textbox and 3 control buttons that allow minimizing, maximizing and closing of the application.

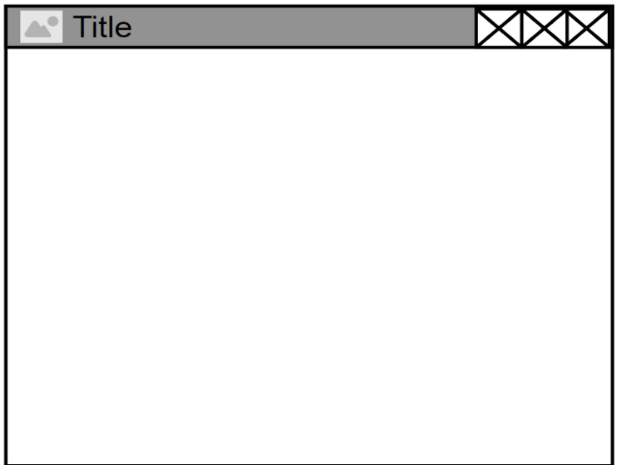


Figure 4.6 - Application layout design

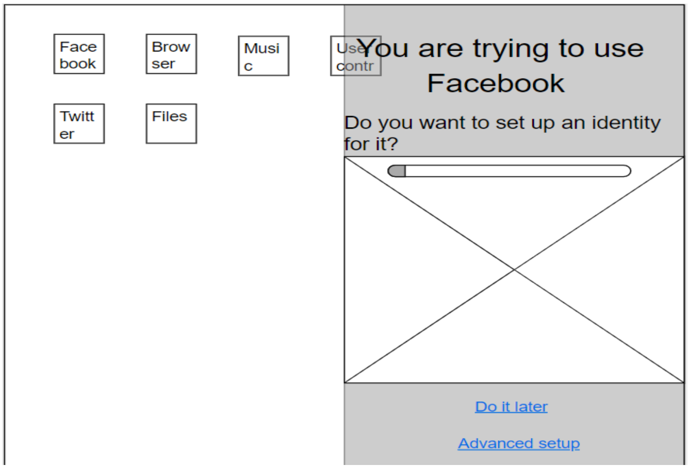
Another mandatory component of the application canvas is the border. It allows resizing by dragging on one of the sides of the border as well as helps distinguishing the application canvas from the background, aiding accessibility.

The white area is the application canvas which allows the drawing of the application-specific contents.

4.3.5 NOTIFICATION CENTER DESIGN

The notification center is supposed to notify the users of potential actions and offer future actions, making the usage of the framework an easier and smoother experience.

It displays as an overlay over the main canvas area and can be dismissed by simply clicking anywhere else.



You are trying to use Facebook

Do you want to set up an identity for it?

Figure 4.7 - Notification center design

4.3.6 DESIGN COHERENCE

One crucial task of the design is developing a standard and coherence that is supposed to be respected across the entire framework. As such, the following design choices have been made.

Regarding the color theme, the colors of the interface have been split into 5 categories: active, inactive, foreground, background, accent. Active color is the color of the interface in case an element is currently active (such as the Titlebar of an application currently in the foreground). Inactive color is the color of an element in case the element is currently inactive. Foreground is the color of glyphs and drawings while background is the color of the background of canvases. Accent color is the color used when overlaying elements.

In the case of the font colors, they will be adapted to the background color, always maintaining an acceptable contrast ratio.

One last important factor in the design of the interface is having a font scale factor. All the text in the application will be scaled by this factor and by modifying this value, all the interface can scale and aid people with visual impairment.

5 IMPLEMENTATION

This chapter focuses on the implementation of the current prototype, discussing the implementation of the design choices as well as successes and failures during the implementation phase.

5.1 System overview

The following paragraphs are explaining the UML diagram provided in Figure 5.1 - System overview UML diagram.

The system is in neutral state when no user is currently authenticated. When the user inserts a drive, the system will check for existing user configurations and profiles. If the files are available, the user is prompted to authenticate. If the files are missing, the user is prompted to create a new profile.

After the user is authenticated, the application opens the desktop environment and allows the user to perform their desired activities. During this state, the presence of the drive is constantly monitored as well as the actions performed by the user.

If the system detects an unsafe activity, the user is warned and prompted with solutions.

When the system detects that the drive is removed, the application closes all currently running actions and disposes of all the currently stored information. Then, it returns to the neutral state, awaiting another drive to be plugged in.

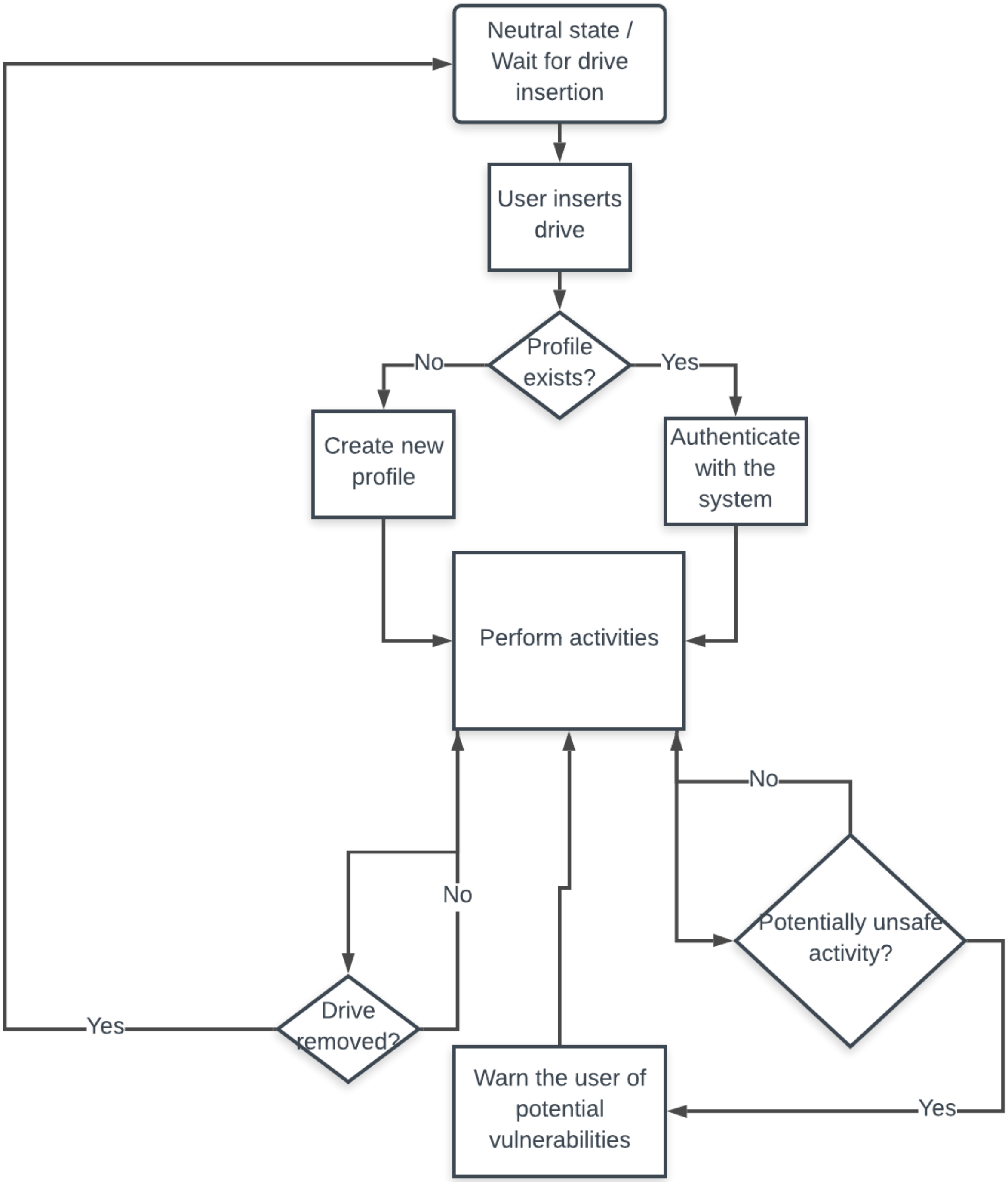

Neutral state / Wait for drive insertion
User inserts drive
Profile exists?
No
Yes
Create new profile
Authenticate with the system
Perform activities
No
Potentially unsafe activity?
No
Drive removed?
Yes
Yes
Warn the user of potential vulnerabilities

Figure 5.1 - System overview UML diagram

5.2 Choosing the operating system and development framework

According to most current statistics, the market share is still dominated by Windows and Android, with both of them hovering around 38% of the market share [8]. As such, we would try to target one of these platforms if possible. Also, these statistics align with our hypothesis that most users are running Windows OS on their PC, especially the less experienced people. One other thing to note is that while cross-platform development would be possible, it didn't rank high on the priority list as the highest priority was developing a prototype that could provide some results first.

Concluding the previous specified points, the choice has been made to mainly target the Windows OS platform, while still keeping in mind the consideration that eventually the finished product should be cross-platform, or at least portable to UNIX operating systems due to the advantages they possess.

After choosing the operating system, the next step was choosing the programming language and the framework in which to develop the prototype. The choice was made to use one of the two C++ or C# due to great performance compared to interpreted languages and due to the possibility of building a cross-platform application in the future which both languages allow.

The following frameworks were taken into consideration: MFC, WinForms, WTL, WPF. MFC can be easy to use and implement but doesn't have any easy way to customize the interface, such as styling buttons, which was crucial for our current work as we tried to emulate an operating system interface. WTL is more lightweight than MFC but is again too restricted for what this prototype would require. As such, we arrived at the final two choices, WinForms and WPF. Both could do what would be required, but in the end, due to WinForms support recently being ended, greater support for customization of the interface and the all-around more developed platform, the choice was Window Presentation Foundation.

The Windows Presentation Foundation is a graphical system used for rendering interfaces. The main advantage of this platform is that it allows the design of the interface using XAML, Extensible Application Markup Language, a declarative language based on XML. The logic of the application would be separately placed in files and written in C#. While developing the prototype in two languages could provide some hurdles, having the logic separated from the interface is a major advantage if at any point during development the choice will be made to switch to a cross-platform framework. Windows Presentation Foundation is also a mature framework with loads of information and examples which would ease the development cycle. One last big advantage of the framework is that it allows the design and creation of custom controls, which can then be reused in the application wherever the need, thus greatly reducing the complexity of the work.

5.3 Implementing inner windows controls

As shown in the design chapter, in Figure 4.5 - Design of main canvas area, the interface is supposed to emulate the look of most operating systems, with a desktop background, icons placed on the desktop that allow placement and double-clicking to open the associated application. As such, the inner windows controls had to have two major separated components, the icon glyph which would be placed onto the desktop and the actual window that would be displayed when an application is running.

5.3.1 IMPLEMENTING THE ICON GLYPH

The icon glyph implementation is rather straight-forward, with an image and a textbox all placed onto a canvas. The glyph will then be placed onto the desktop, positioned into a grid. When the user clicks on the glyph, it is displaced from the grid and allowed to be freely moved. When the user releases the left mouse button, the new position is then approximated onto the grid and the glyph is placed back into the grid.

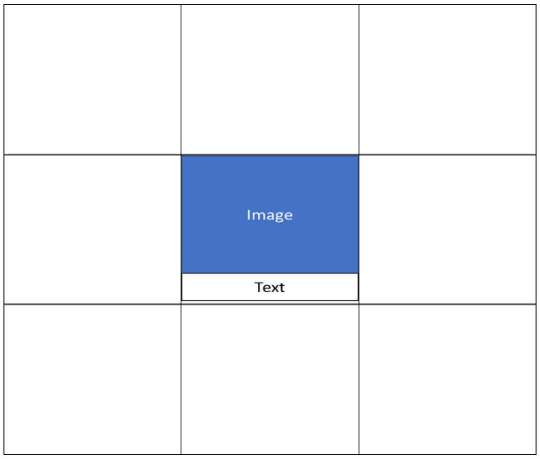


Figure 5.2 - Icon glyph design

For the actual window implementation, the process was not that simple. While there are lots of examples on how to use the current framework, nothing has been done to simulate an actual desktop interface, and as such, there were no easy ways to achieve the desired results. This process required implementing all the actions of the control, the resizing, repositioning and drawing of the contents from scratch.

The window was split into a couple of different controls, to make any eventual modifications easier than having to rewrite everything again. An inner window control will have a titlebar, a group of eight rectangles to facilitate resizing and a canvas that allows drawing of dynamic contents.

Please refer to Figure 9.6 - Desktop environment to observe the final look of the discussed features.

5.3.2 IMPLEMENTING THE TITLEBAR

The titlebar implementation was again pretty straight-forward, with a filled rectangle, 3 buttons placed on the right side to control the windows and an image control to display the application icon. The titlebar exposes the following events: ExitButtonClicked, ResizeButtonClicked, MinimizeButtonClicked. The events allow the integration with the rest of the controls of the window. The events are triggered when the related button is clicked. Also, the following properties are exposed, allowing the manipulation of the titlebar: Text, Icon, FontColor, Color.

5.3.3 IMPLEMENTING THE BORDERS

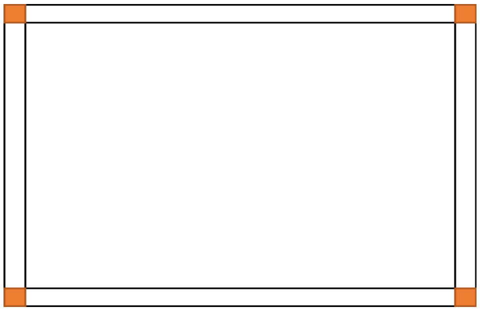


Figure 5.3 - Window borders sketch

The borders were implemented using eight rectangles that capture mouse actions. The rectangles were named as such as that their roles would be self-explanatory: LeftBorder, TopBorder, RightBorder, BottomBorder, TopLeftBorder, TopRightBorder, BottomLeftBorder, BottomRightBorder. When pressing the left mouse button on one of the rectangles, they would start capturing the mouse movement. On each tick of the application, a delta position would be calculated which would be the difference between the mouse position in the previous tick and the current position of the mouse, and based on this delta, a new position and size would be calculated for the window. Also, due to the choice of coordinate system with the origin in the top left corner, this calculated offset would have to be either applied to the position of the window or to the size, depending on which border was being manipulated.

5.3.4 IMPLEMENTING ADDITIONAL FUNCTIONALITY

Now that most of the functionality has been implemented, the remaining things were implementing the actions that would happen on clicking the buttons in the titlebars as well as capturing mouse movement onto the surface of the window.

When the titlebar is pressed, the window can be moved by capturing the position of the mouse each tick and obtaining an offset value from the current and previous position of the mouse. Thus, the position of the window is calculated based on this offset.

When the ExitButton on the titlebar is clicked, the window raises the ExitClicked event, which is handled by the desktop controller.

ResizeClick event is raised when the resizing button has been clicked. After that, the window would proceed with the resizing algorithm described in chapter 5.3.3-Implementing the borders.

MinimizeClick event is raised when a minimization would occur and Resized event would be raised once the resizing has been finished.

The window also exposes two colors, ActiveColor and InactiveColor that allow designing the window. The Icon property sets the image used as icon on the titlebar. The Active property sets or returns a Boolean value specifying if the window is active or inactive. WindowName property allows setting of the text on the titlebar and CurrentState property allows manipulating the state of the window, if maximized, minimized or in normal state.

To showcase the look and feel of the implementation, please refer to Figure 9.7 - Inner windows showcase.

5.4 Implementing accessibility methods

During the development of the prototype, accessibility was a priority and as such had to be taken into account into every step. To ease the complexity of adapting an entire dynamic interface to the parameters of accessibility, a solution has been developed. This solution has been described in 4.3.6-Design coherence. In this chapter, the discussion will be shifted on the actual implementation of the design discussed previously.

To have a unified implementation, the theme colors and font sizes were saved globally. When modifying one of the components of the theme (font sizes or colors), the following two events would be raised, depending on which component would be modified: ThemeChanged, FontSizeChanged.

To have fonts of different size but still maintain scalability over all the application, a scale factor for the fonts has been saved, as such, all fonts would scale with this font factor. When the event FontSizeChanged is called, all the components will adapt their font sizes based on the global factor.

In a similar fashion, when one of the colors is changed, ThemeChanged event is raised and all components change their colors based on the current color theme.

A showcase of the customizability of font sizes and colors can be observed in Figure 9.9 - Accessibility features showcase.

In the case of multi-language support, WPF provides a solution that fulfills the requirements of the prototype. All the text strings were saved in a resource file called language. When the current UI culture would be changed on the thread running the UI, WPF automatically uses the resource file with the extension that equates the current language of the UI. For example, when switching from English to Czech, when initializing a new control, all the text strings would be loaded from the resource file called language-cz.resx instead of language.resx. While this solution doesn't allow switching languages at runtime without reloading the entire interface, it was not deemed a priority to implement an entire new method from scratch to deal with the localization.

5.5 Implementing the authentication with the PC system

To detect when a new USB device is inserted in the PC, the application at launch implements a handler for the event where a new volume is found. Similarly, to detect USB removal, a handler is implemented for the event when a volume is removed.

When a new device is detected, the application checks for an existing user profile and if there's none, it prompts the user to create a new profile. If there exists a user profile, the application prompts the user to log into the application.

5.6 Implementing the log-in procedures.

Initially, the design choice has been made to store the user information as clear-text, because of the "key metaphor", that losing your key grants access to anybody to your storage. As such, there was no encryption algorithm and the log-in procedure were just to stop unexperienced users from accessing information using this application, but a malicious actor could just read the user profile file to gather whatever information they'd want.

Due to the previous choice and the lack of an encryption algorithm, a multitude of log-in authentications were possible. One of the most interesting ones will be discussed in the following chapter, as well as explaining why having to encrypt the user profile is at odds with this log-in procedure.

5.6.1 SIGNATURE BASED AUTHENTICATION

For the implementation of a signature-based authentication, the best option was using a trained system based on a Hidden Markov Model statistical system. The second option was an implementation using Support-Vector Machine. Both systems behave the same in respect to the conditions and requirements of the task and as such, this chapter will mostly discuss how the systems were used, not what they mean and how to implement them. As for the actual implementations, the library accord.net [9] has implementations for these systems.

To gather the actual signature, a component was developed that would start capturing and storing the position of the mouse related to the position of the control. As such, the actual shape of the signature would be captured and stored.

In order to actually identify the signature, the user would be requested to input their signature three times, to form the training data for the learning system. These three sets of input data would be used as learning data. The number of three tries has been chosen based on experimentation. After the learning data has been formed, the system would be offered the test input set and tasked with deciding if the input belongs to the class of the previous data or not. A positive reinforcement loop has been created in which, each positive recognition would result in the test data added to the learning data. Again, after experimentation, an acceptable number of hidden features were found that would result in the highest successful identifications while still having a respectable algorithm running time.

The problem arises when the users have requested that their information is stored encrypted onto a device. Creating a cypher based on the signature proved to be a very difficult problem and this current discussed method would require the learning data residing on the user device. Any person understanding the code of the application and the classification algorithm could take the input data and un-encrypt the personal data. There was no solution to keep this authentication method while providing a secure encryption algorithm for the data personal data.

In the end, after conducting the experiment, this authentication method has been discarded and an encryption algorithm has been selected to encrypt the user profile data.

5.6.2 PASSWORD AND PINCODE BASED AUTHENTICATION

Before the experiment, these two authentication methods were part of the three authentication methods implemented for the prototype. These two implementations differed mostly on the GUI level, while the authentication algorithms were the same. This is why they are paired together in this chapter and will be discussed at the same time.

When the user is prompted to introduce their password or pincode, the strings are stored into SecureString objects, which are a solution provided by the .NET framework that stores strings safely into memory. The choice to store the strings into SecureString objects is to avoid attacks based on memory scraping either at runtime, from pagefiles stored on the disk or from memory dumps.

When comparing, the SecureStrings are transformed into managed strings locally, compared and then the objects disposed.

Once the need for encryption has been discussed, it was easy to use these strings as a key for an encryption algorithm in order to encrypt the user profile.

5.7 User profile encryption

The user profile contains personal and eventually critical information of the user and as such, this information has to be protected. In order to protect this information, an encryption algorithm has been used. The algorithm steps are showed in figures Figure 5.5 - Decryption procedure and Figure 5.4 - Encryption procedure.

For the encryption procedure, the data and hashed user password is generated. For the user password, a 256bytes hash is generated. Then, the algorithm generates 256 bytes of random entropy for both IV and salt. A .NET implementation of the Rijndael encryption algorithm is configured using the hashed password as key, the generated salt and IV. The algorithm is then used to encrypt the provided data. The encryption is performed a pre-determined number of times.

For the decryption procedure, the IV and salt are extracted from the encrypted data. After, the same Rijndael algorithm implementation is used and configured with the extracted salt and IV to decrypt the data. The decryption is performed the same pre-determined number of times.

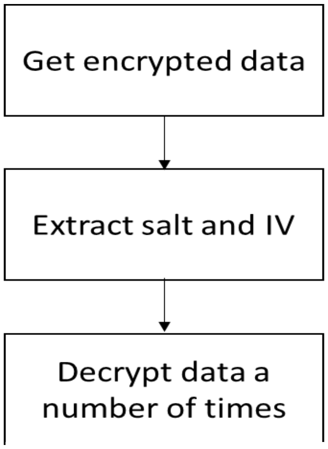


Figure 5.5 - Decryption procedure

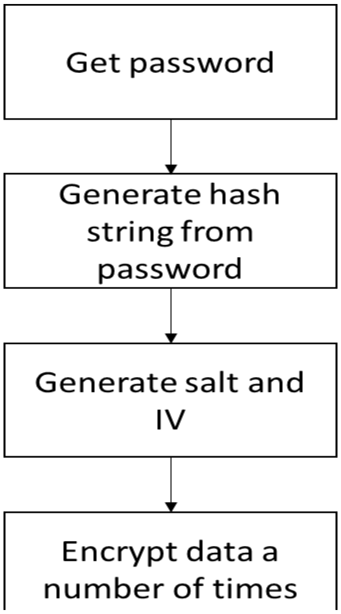

Get password
Generate hash string from password
Generate salt and IV
Encrypt data a number of times

Figure 5.4 - Encryption procedure

5.8 Implementing the browser control

In order to provide internet-surfing capabilities to the users, a browser control had to be implemented. Given the overall large support and acceptance of the users, as well as positive reviews, Chromium has been selected as the browser to be used in the prototype.

To embed Chromium into the application, a library has been used which provides bindings from the C++ source code of the browser to C# [10].

Initially, the WPF implementation of the browser has been used in the prototype. The disadvantages of the WPF became apparent when the double-buffering of the interface resulted in poor framerates.

To solve this issue, the browser was opened as a normal browser and the interface was later drew into the interface, avoiding the double-buffering of the control and obtaining a great improvement of almost double the number of frames displayed per second.

Although this solution fixed the framerate issues, this resulted in the problem that displaying contents directly, they would not be sorted by the index during displaying, resulting in the browser being displayed on top of everything in the application.

To solve the display issue, a work-around solution has been implemented. When the browser control would be put into the back, behind another control, the browser buffer is captured and an image is generated. Then, the browser is hidden and the image is displayed as an image control which performs the double-buffering techniques of WPF and allows sorting by Z-Index resulting in the image being positioned behind the content in front of it.

Besides the previous problems, the browser control had to have a number of controls to manage the browsing experience, such as forward and back on the webpages.

Another challenge was implementing the tabs of the browser control, to simulate the functionality of currently developed browsers. In the end, the choice was made to open new Chromium instances for each tab. While this may use additional resources, it separates the multiple processes, avoiding shared memory vulnerabilities.

The last challenging implementation was the implementation of the address bar. In order to provide functionality similar to browser currently in development, the string introduced in the address bar had to be parsed. The input string is first checked if it contains “http” or “https” at the beginning. If the string then can be processed into an URI (uniform resource identifier), then it is a valid link and it is given as an address to the browser. If the string cannot be processed into an URI, the string is given as a parameter to a google search, effectively searching for the string using the google search engine.

The look of the browser control can be observed in Figure 9.7 - Inner windows showcase.

5.9 Implementing login detection

One priority of this prototype was to develop some algorithm that detects login forms, effectively allowing the application to monitor when the user is asked to log in, in order to raise the user’s attention.

There is no known or easy way to do this action and as such, the current algorithm is only work in progress and will not work on all the websites. Instead, it does work on the most common social network websites such as Facebook, Google, etc.

The algorithm starts by checking the URL for keywords such as “signin”, “login”, “log-in” and “sign in”. This step is necessary to detect the sign in procedure from Google, which doesn’t use forms. Then, if those keywords are not detected, the HTML of the page is loaded and parsed for forms. Then, each form is checked for its id containing “login”, “log-in”, “log_in”, or for the action of the form being named “login”, “log-in”, “sign-in”, “signin”. If any of these are detected, then the browser alerts the user that their log in information is currently detected.

6 EXPERIMENT

This chapter focuses on the experimentation and testing part of the prototype. In order to achieve a 95% confidence to observe problems affecting 50% of the users, the following formula: $nr_{participants} = \frac{ln(1-conf)}{ln(1-usr_a)}$, where $conf = 0.95$ is the degree of confidence and $usr_a = 0.5$ is the percentage of users affected. Thus, five people from the target group has been deemed as sufficient to test the current prototype. Additionally, a sixth person has been added to the group, to act as a reference, as this sixth person is capable in using a computer and his behavior would be used only to further test the performance and capabilities of people using the provided framework without any guidance.

6.1 Motivation

The main motivation of running this experiment has been to obtain valuable insights into the usability of the developed framework. Due to the nature of the target group (people with accessibility issues as well as low-level computer usage capabilities), there is a necessity to test the prototype on participants from the target group as their input and opinions might not only be crucial, but also completely unexpected and might require big changes to the final solution.

6.2 Experiment setup

For the conduction of the experiment, a polished and restricted version of the solution has been prepared. The six participants were provided with access to a computer running the currently developed prototype. They were given a set of tasks that they were supposed to complete while being evaluated on their actions. The

6.2.1 PREPARATION

The first step was the preparation of the experiment and the setting up the goals. The main point of this step was to define a clear path that we wanted the subjects to follow, the amount of information and help that we should provide up-front for the subjects before the beginning of the experiment, the expected behavior as well as preparing for the most obvious mistakes and deviations from the initial plan.

As a path, we defined the exact steps that the participants are expected (and would be guided) to follow through the application. In our case, these steps involved the following tasks: Understanding the system, inserting the key into the system, managing to follow the user creation step, managing to set up an authentication method, understanding the desktop environment, managing to use basic application features, disconnecting from the system, reconnecting to the system and managing to reauthenticate with the existing user. We will expand each of these procedures in the chapter 6.2.3 Experiment .

The next step of the preparation focused on the preparation of the application. As the application is in currently a prototype, it cannot accurately represent the goals and features that are supposed to be present in the finished product. To ensure that the test would run without problems as well as that the participants would not be confused, we had to develop an experimental build that would hide or remove all the features we were not interested in testing as well as pushing to have the tested features as complete as possible. Due to the time constraint, we decided to polish the basic features of the application such as authentication and basic accessibility and customization. We ended up removing some unfinished features and came up with a simple build that masked all features that were incomplete, or we deemed might be too confusing for the participants.

The following step involved choosing the participants. We had to decide on some characteristics that we expected the participants to have as well as provide a bit of variety to the group. Initially, we chose five participants, their characteristics being described in a later chapter. During the moments before the actual experiment, we decided to add a sixth participant, one that was far more experimented and capable than the others, with the goal to test the application to its limits.

The last step was finding the location and determining how the experiment will take place. We decided that each participant would be tested separately alone in a room with the reviewer, so that no participant could learn from previous ones some parts of the application. Doing this, we obtained untampered insights into their behaviors.

6.2.2 EXPERIMENT GOAL

The goal of the experiment was to obtain as much information as possible into how the participants interact with the system. We wished to observe how the system makes the users behave as well as to listen to feedback on the experience of the participants.

One extra goal was to ask the participants about possible features that they would like to be present in the finished product.

6.2.3 EXPERIMENT TASKS

In the case of understanding the system, the reviewer would provide the participants with as much information as possible (without stressing or tiring the participants), so that the participants could understand the goal of the system and how to utilize it. During this first part, the participants would be introduced to the "key metaphor". The participants would be told how all the information from the developed system would only be saved locally onto their flash sticks and that they should keep it the same as they would with any key. After, the participants would be instructed on how to insert the stick into the USB port of the computer and presented with a demonstration of the authentication methods. They would be told that they can choose any of the methods provided based on how comfortable they feel with each one.

During the part of inserting the key into the system, the participants would be monitored as they try to insert the USB stick into the USB port of the computer to determine how accustomed they are to it.

For the part that focused on the procedure of creating a new user profile, the participants would be asked to follow the instructions on the screen in order to create their new user profile. The main goal of this part would be monitoring how the users react to the instructions and how well can they follow the provided instructions.

Once the users are to be introduced to the system, they have to choose an authentication method for further uses of the system. During this part, the reviewer is supposed to observe if the participants manage to properly set up an authentication method.

Following the authentication, the users would be presented with a default environment that they could further customize to their needs. During this part, the participants are to be told what this environment is capable of and are to be instructed to test some of the functionality of the system, mainly navigating through the desktop environment and opening a few windows and manipulating them.

Following the previous part, the participants are to be allowed to toy with the system, to try and discover as much as possible and customize the system to their desire. During this part there should be a minimal level of interaction between the reviewer and the participants as this step’s main goal is to observe how intuitive the system is and how easy it is for the participants to discover the features provided.

Once the discovering phase is over, the participants are to be asked to disconnect from the system. They are to be instructed on how to do so (by removing the USB stick from the computer). This part should provide some insight for the users on how to use the system.

After disconnecting from the system, the participants would be asked to reconnect to the system. During this part, the reviewer should take again an observational stance. The participants should follow the instructions displayed by the system. Finishing this part would conclude the experiment.

6.3 Participants

The experiment has been conducted on a number of six participants, out of which, five were part of our target group and one was part of a potential future target group.

Participant ID	Age	Gender	Willingness to learn	Experience
1	53	Female	6	4
2	58	Male	8	6
3	82	Male	6	2
4	80	Female	3	1
5	76	Male	7	3
6	21	Male	10	10

Table 6.1 - Table of participants

6.3.1 EXPLANATIONS

The “Willingness to learn” category provides a grading from one to ten. It measures the willingness of the participants to learn about the provided framework as well as their willingness to use the computers in general.

The “Experience” category grades the participants on a scale from one to ten in regard to their experience in computer usage and internet usage. It provides a universal value that should be associated with the overall proficiency of the participant in regard to computers.

The values provided in the above-mentioned categories are not to be taken for more than face value as their purpose is to mostly provide the reader of this document with some additional information in regard to each participant.

6.3.2 REMARKS

This section will provide some additional remarks about the participants as well as some topics of further expansion. The observations provided in the following paragraph are not to be taken as any conclusive data, as due to the low number of participants, the following statements could be purely coincidental.

From observing the provided table, we can see some potential connections between the gender and age of the participants and their proficiency. Older people were in general less proficient, with the addition that female participants had lower proficiency levels than the males in their age group. The willingness to learn was also distributed in a similar fashion. In addition to the previous trend, we can also observe that willingness to learn is usually directly proportional to the proficiency of the participant.

6.4 Summary of the experiment

In this part of the document, we will focus on the experiment, how it evolved and how it was concluded on a participant basis. The participants will be referred to by their assigned ids, according to the table provided previously (Table 6.1 - Table of participants). The summary is split into the steps described in chapter 6.2.3-Experiment .

6.4.1 UNDERSTANDING THE SYSTEM

During this part of the experiment, all participants understood unremarkably how the system is supposed to function as well as what to expect from it. One notable exception was participant 3 who seemed particularly accepting and excited to test the system. All the participants understood the “key metaphor”. They agreed that this metaphor fits the situation. Participants 1, 2, 3 and 5 expressed that using the metaphor makes them better understand the value of their privacy and information and that they should protect their information to the best of their abilities.

6.4.2 INSERTING THE KEY INTO THE SYSTEM

This part of the experiment was eventually fulfilled without major complications. Participant 4 was the only one who had to be guided on how to insert the key. This provided the information that even the most basic users know what an USB device is and how to plug it into the computer.

6.4.3 CREATING A NEW USER PROFILE

Participants were monitored (in some cases guided) while they created their initial user account. They had to follow the provided instructions. Participants 3 and 4 had trouble orientating onto the screen, but eventually did manage to perform the task.

6.4.4 CHOOSING AN AUTHENTICATION METHOD

When prompted by the system, the users had to choose an authentication method. The participants could ask what each method meant. All participants barring participant 6 chose the authentication method by signature. Participant 6 chose authentication by password.

6.4.5 UNDERSTANDING THE DESKTOP ENVIRONMENT

The participants were presented with the environment inside the application and were allowed to play with it. Participants 3, 4 and 5 were particularly interested in the behavior of the windows, how to resize them and how to open additional windows. Participants 1,2 and 6, being previously more accustomed with similar environments, instead tried to discover as many features and settings as possible.

6.4.6 USING BASIC FEATURES

This part of the experiment was, to a degree, blended into the previous one. Participants 1, 2 and 6 already discovered most of the features before being guided in doing so. Participants 3, 4 and 5 were guided onto the basic features and demonstrated some of the capabilities of the framework.

6.4.7 DISCONNECTING FROM THE SYSTEM

Due to understanding how to connect to the system in the previous part, this part evolved without any extraordinary event. All participants successfully removed the USB device from the computer and in doing so, managed to successfully disconnect.

6.4.8 RECONNECTING TO THE SYSTEM

Similarly to disconnecting, the participants remembered the previous steps and without any help, they managed to reconnect their device to the computer.

6.4.9 REAUTHENTICATING WITH THE SYSTEM

Once reconnected, the system asked the participants to authenticate in order to use their previously created user profiles. Participants 1 and 6 managed to do it without any guidance, while participants 2, 3, 4 and 5 had to be guided to some degree. The main stumble was that the participants hadn't understood that they should use the same authentication method as before. Participant 4 had to be helped in particular to also remember the authentication method chosen before.

6.5 Conclusion of the experiment

After all the steps have been performed, the experiment was concluded. The participants were asked to rate their experience and provide their insights and opinions. This step proved particularly useful, as will be explained in the following paragraphs.

The users were asked if they'd use the framework in its current state. Participants 1 and 3 answered yes. Participants 2, 5 and 6 answered no, justifying their answer as the lack of desired features. From this answer, we have concluded that participants would require additional usability and social features added. One such reoccurring desire was the presence of a file manager as well as text editor. Participant 4 answered that they would not use the system as they have no desire to interact with computers on a regular basis and trying this platform didn’t change their mind.

During the discussion about features, one important question presented by participant 6 was regarding the safety of the data in case of a lost flash stick. Prior to this experiment, all data was saved unencrypted, which in case of a loss would provide a malevolent actor with a lot of confidential or personal information. When we presented this eventuality with the other participants, they all agreed that it is unacceptable. When told that they could sacrifice authentication methods in order to for their information to be stored safely (read encrypted) onto the device, all participants chose the to give up on usability in exchange for safety. Particularly, in this case, the choice was to give up on signature authentication, to which they all agreed. The reason why signature authentication is not possible with data encryption is provided in chapter 5-Implementation.

7 CONCLUSION

Due to the recent explosion of social media usage as well as recent scandals involving privacy violations, a solution is necessary that would tackle the privacy concerns of individuals. The developed framework attempts to provide a solution to the security and privacy concerns, designed for especially vulnerable people, as those part of the designated target group.

The developed framework aims to make the computer experience an easier one for people still learning and for people with lower attention span. Older people especially are included into the target group by providing accessibility methods to enhance usability.

While currently the framework only supports web browsing and basic accessibility methods, the goal is to provide a complete solution that would cover all basic and casual computer uses, maintain the privacy goals and provide a safe experience while both surfing the web and while performing basic activities such as listening to music, document editing, playing videos, etc.

The results of the tests show that the project is heading into the right direction. The target group is especially in need of some software like the currently developed one and the advantages it provides (usability, accessibility, safety) are greater than the disadvantages (restricted activities, less customizability, less access and control on the computer). The target group is especially interested in the security and accessibility aspects of it.

7.1 Future implementations

The framework is currently a work in progress. As such, in its current state, this software is only a skeleton, developed with a solid foundation and ready for implementations and extensions to extend the usability of it.

The current prototype serves as a solid foundation for further expansion. In this case, the developed framework currently features only a control panel and a fully implemented browser. The following features were part of the original design but are not yet implemented and would be added during a future development.

- Implement webpage scanning and parsing to detect requests for additional data, other than log-in forms.
- Implement a file manager that restricts access to the user drive.
- Implement media features such as music player and video player.
- Implement document readers and editors for various document formats.
- Implement warnings and suggestions during use, such as warning users when doing potentially dangerous activities and providing alternatives or suggestions in the current context. This part would be particularly complicated as there would be a lot of different cases that solutions would have to be developed separately.

Besides the previously mentioned implementations, an addition was under design but was scrapped due to time limitations. This addition was an implementation of a decentralized social media implementation, running in an application. The main ideas of this design were that users would be able to give and revoke access to resources from other users, as well as all the resources would only reside locally on the user's device (of course it would temporarily reside on the device of the people accessing it) and removing the access to the resource or the resource altogether would be completely up to the user, not to some organization hosting data in cloud networks in multiple copies. This feature is extensive and would require a lot of research and effort to be implemented properly, that being the main reason it was removed from the current project and postponed to eventual future development.

8 BIBLIOGRAPHY

[1] I. W. Stats, "Internet Usage Statistics - World Internet Users and Population Stats," , . [Online]. Available: http://www.internetworldstats.com/stats.htm. [Accessed 5 5 2019].

[2] "Pew Research Center: Internet, Science & Technology," [Online]. Available: http://pewinternet.org/chart/internet-use-by-age/. [Accessed 5 5 2019].

[3] OECD, Skills Matter: Further Results from the Survey of Adult Skills, Paris: OECD Publishing, 2016.

[4] J. G. R. E. Audrey Syme, *Older adults' use of computers: a survey,* 2014.

[5] P. R. Center, "Demographics of Social Media Users and Adoption in the United States," 5 February 2018. [Online]. Available: https://www.pewinternet.org/fact-sheet/social-media/. [Accessed 12 May 2019].

[6] S. R. F. T. B. M. V. C. A. D. J. B. J. J. K. J. H. K. J. L. H. L. a. o. Rupert RA Bourne, "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis," *The Lancet Global Health 5,* vol. 5, no. 9, pp. e888--e897, 2017.

[7] W. H. Organization, "Mental health of older adults," 12 December 2017. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/mental-health-of-older-adults. [Accessed 22 May 2019].

[8] StatCounter, "Operating System Market Share Worldwide," StatCounter, 2019. [Online]. Available: http://gs.statcounter.com/os-market-share. [Accessed 12 April 2019].

[9] C. Souza, "Machine learning, computer vision, statistics and general scientific computing for .NET," 2014. [Online]. Available: https://github.com/accord-net/framework. [Accessed 11 December 2018].

[10] ".NET (WPF and Windows Forms) bindings for the Chromium Embedded Framework," 2018. [Online]. Available: https://github.com/cefsharp/CefSharp. [Accessed 22 February 2019].

[11] J. Nielsen, "How many test users in usability study?," 4 June 2012. [Online]. Available: https://www.nngroup.com/articles/how-many-test-users/. [Accessed 10 April 2019].

[12] GlobalWebIndex, "Active social network and active app users, excluding China," GlobalWebIndex, 2014.

9 APPENDIX

9.1 Application showcase

This additional chapter focuses on showcasing some of the functionality of the developed framework. The pictures showcase the steps taken by the framework when the user is trying to authenticate and use the application.

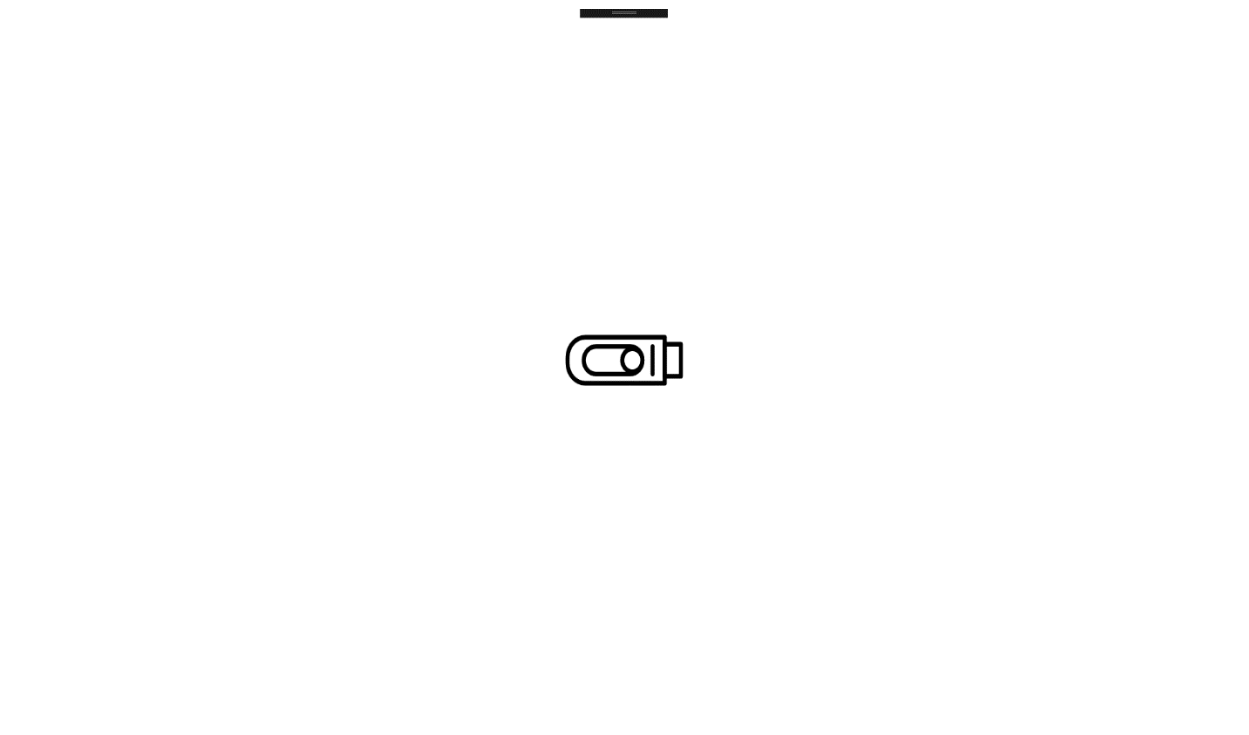
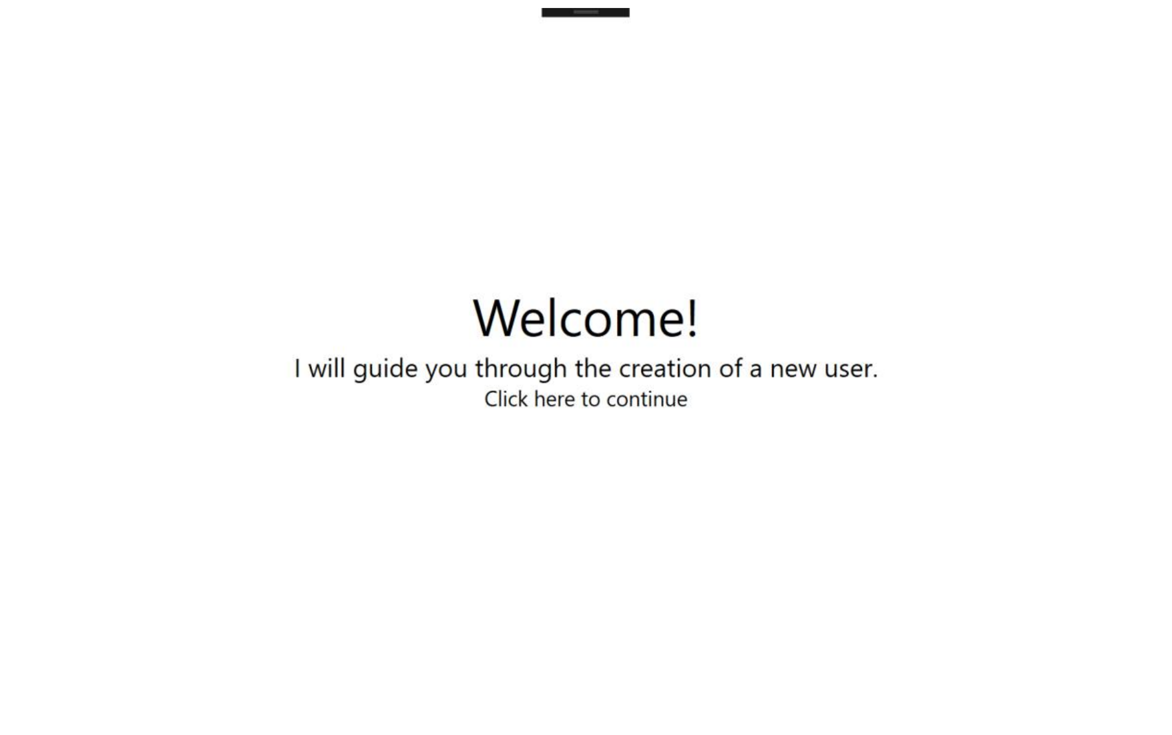


Figure 9.1 – Screensaver

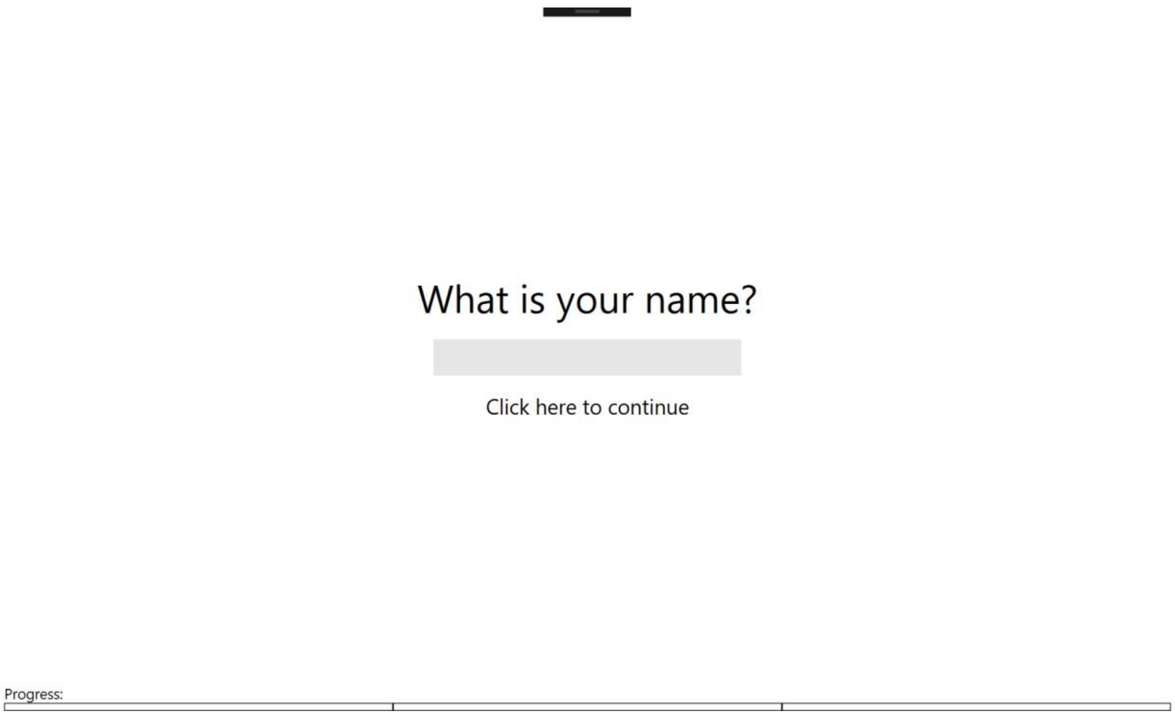


Welcome!

I will guide you through the creation of a new user.

Click here to continue

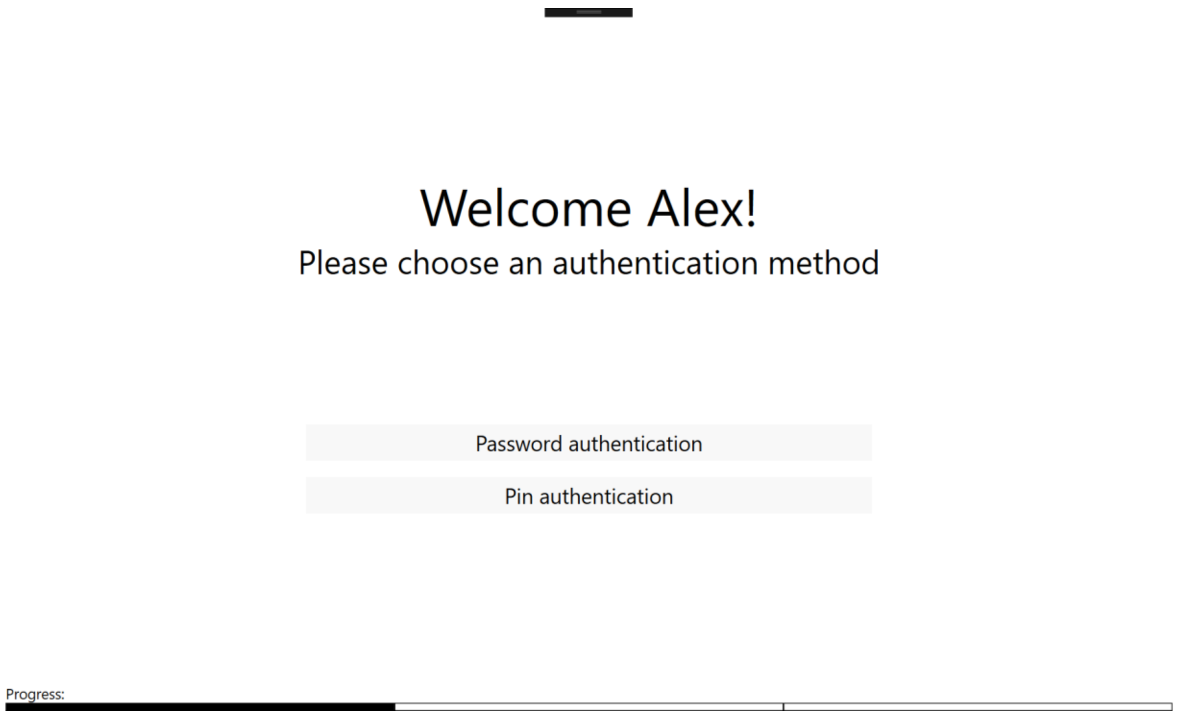
Figure 9.2 - Welcome screen



What is your name?

Click here to continue

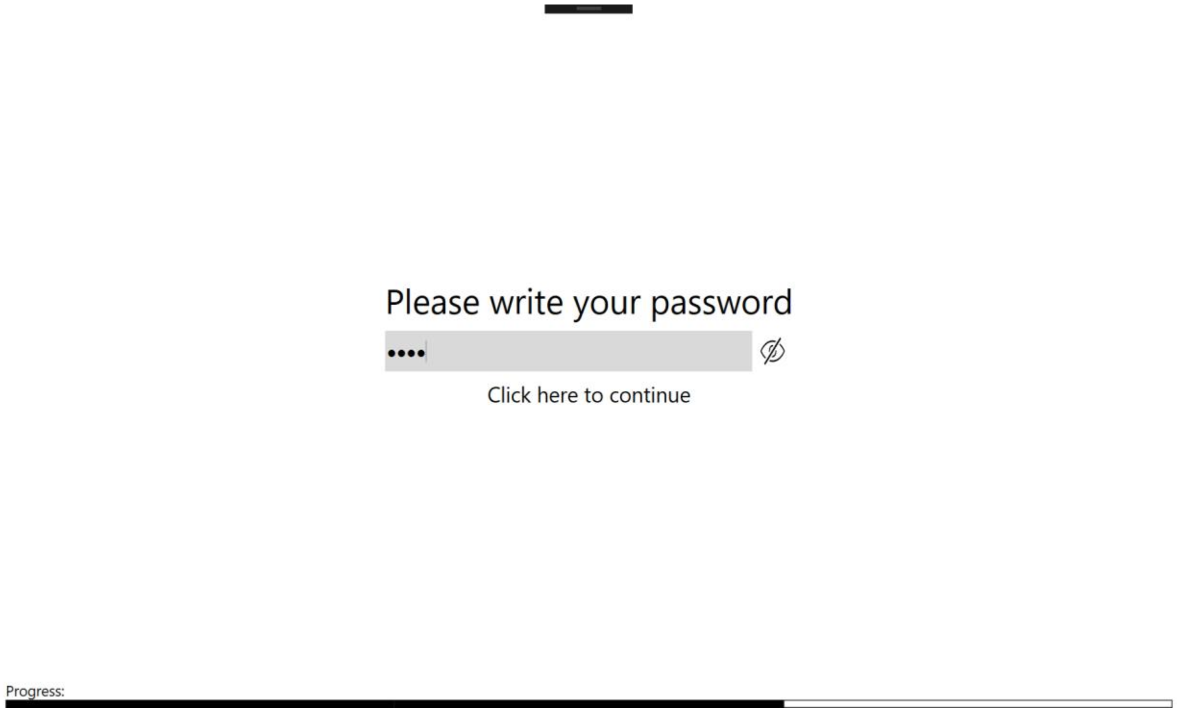
Figure 9.3 - User creation first prompt



Welcome Alex!

Please choose an authentication method

Figure 9.4 - User creation second prompt



Please write your password

Figure 9.5 - User creation last prompt

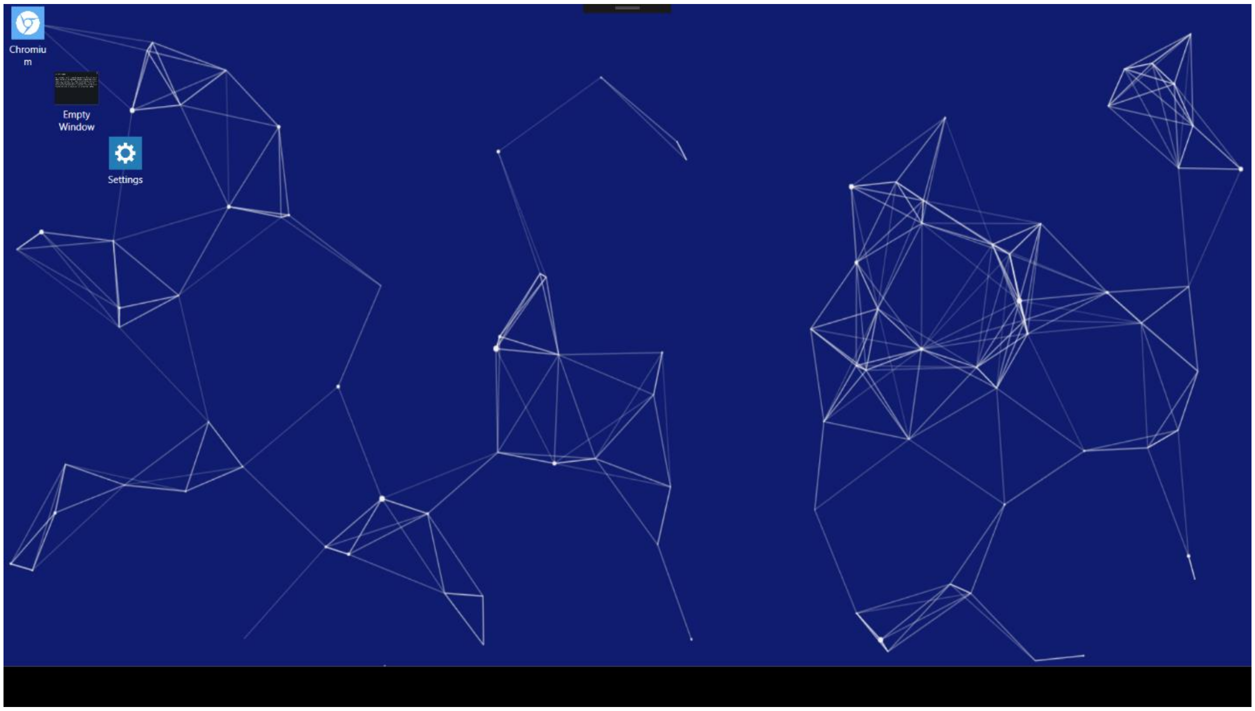


Figure 9.6 - Desktop environment

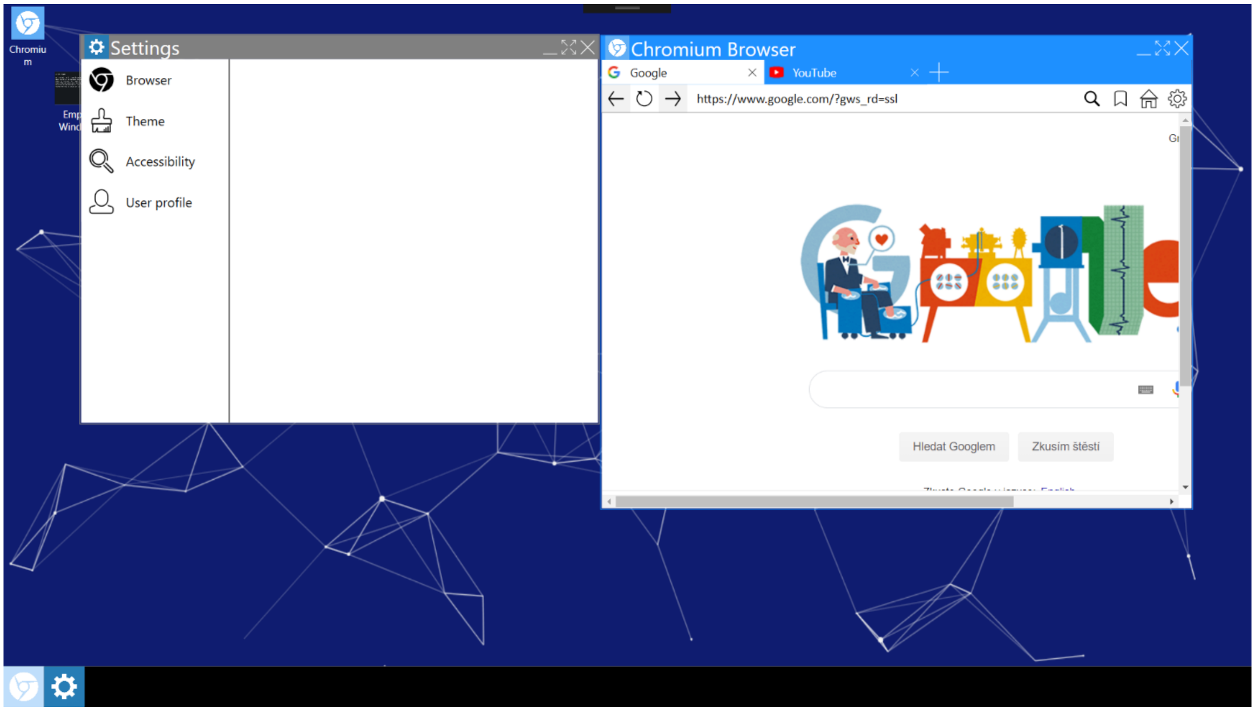
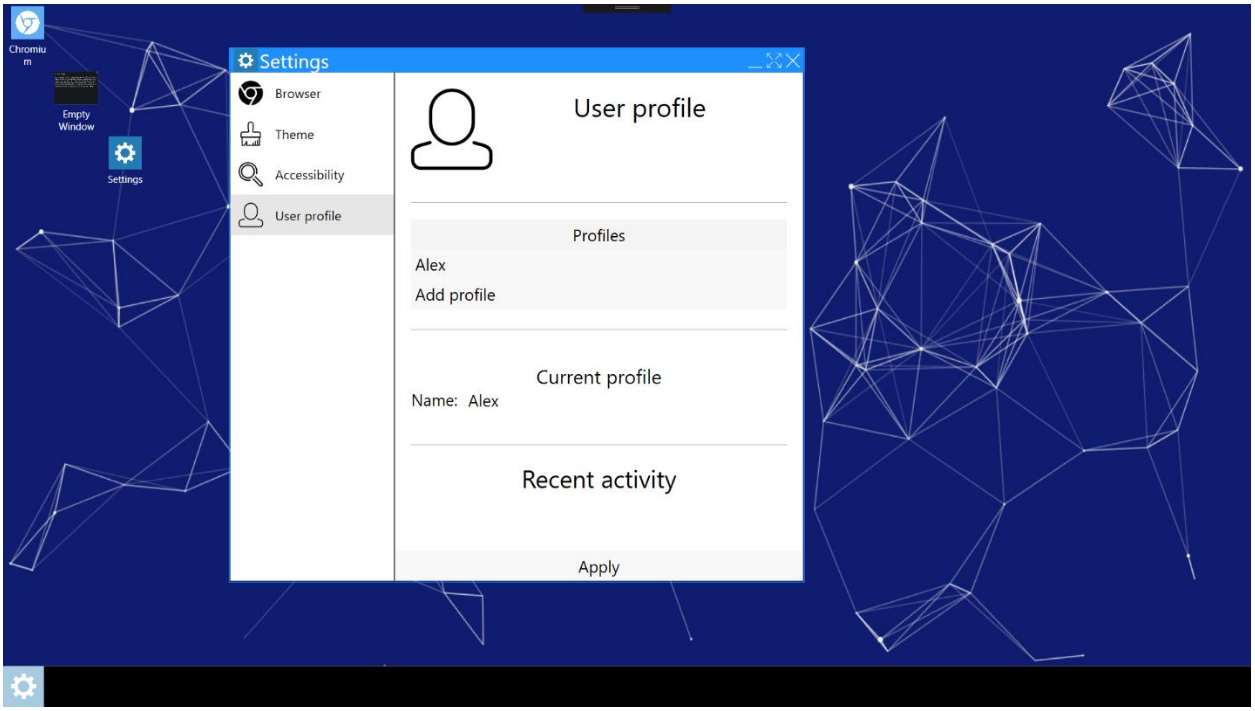
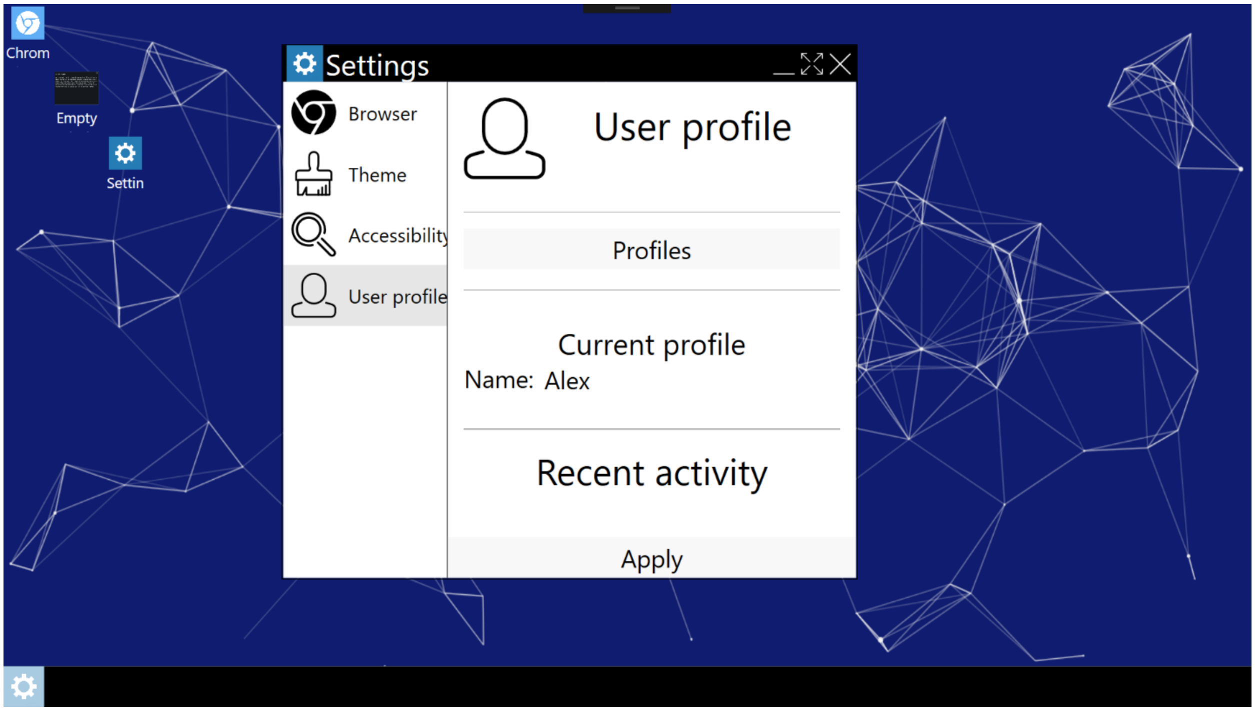


Figure 9.7 - Inner windows showcase



Name: Alex

Figure 9.8 - Control panel showcase



Name: Alex

Figure 9.9 - Accessibility features showcase