T-110.5241 - User Interface Construction

Report for the group assignment

Anonymous substance abuse counseling e-service

February 1, 2013

Introduction

The group is formed by Maciek Ślifirski, Olivier Cléro and Arthur Templé.

We chose topic number 6: **Anonymous substance abuse counseling e-service**.

An anonymous substance abuse counseling e-service

The service is meant to provide information to people in difficulty regarding drug abuse. The aim is to give:

1. proper answers to people's drug problems
2. easy access to common services (like the website, emergency calls, ...)

For this purpose, four different user interfaces have been developped and are provided:

- Web
- Mobile
- Desktop
- Command Line

Furthermore, a placeholder backend server is also incldued, which provides PDF files according to users' requests. Basically, it searches in the request (which is a text string) for drug names, and crafts a proper PDF file accordingly to the identified drugs the user may have typed in the request.

Web Interface

Overview

Web User Interface version of this project may be considered as alternative and widely available version of the Desktop Interface. It contains almost the same functionality as Desktop Interface, except search history. Main functionality is based on command line input, where users can describe their problems and as a result receive PDF ïňĄle from server. File can be stored in user default location or displayed in new tab using default PDF reader - it depends on browser settings.

Advantages and issues from Web User Interface

Main target was to create interface, which is really easy to use, due to target group of users. Drug addiction can touch people regardless their age and cause health problems (mental or physical), which often decrease their ability to use advanced functions. Moreover, interface is also designed for people looking for information about help for their friends or family. The only action that user is expected to do is describe nature of the problem - drug types are automatically recognized from the input. There are no hidden functions or options and all actions are visible for the user (e.g. when PDF file is ready to download). Also to receive answer for the request, no contact information are needed. User can retrieve file

directly from server and in addition helpline numbers are displayed. There is no interaction with user's file system - all files can be downloaded only if users will accept it. Colors of the layout are toned to give the impression of serenity - no sharp or drawing attention ones. Our idea was to create website that looks like quiet, private room, where user can ask for help without spectators. Moreover website does not contain any special graphical effects, transitions or even advertisements in order to minimize response time. All visual effects are using browser's native functions, which also makes it compatible with every computer.

Technical choices

Web User Interface have been developed in Bluefish editor with HTML5, JavaScript, jQuery and CSS3. These languages used together are powerful tool for creating websites that can act like applications. The reason behind using this technology is compatibility with all browsers and all operating system, so with our application we can gain huge group of users. Moreover, developing application in this environment is quick and easy, because all necessary knowledge is available in W3Schools tutorials. Core of the website is based on HTML5 DOM. JavaScript with jQuery library handles displaying DOM objects on the website and process request to server. Layout is written in CSS3, which enables developers to change it, whenever they want - e.g. colors or fonts may depend on the season. Another reason of this choice is no need to use external plugins to display interface. It decreases loading time and amount of used resources.

Outcomes

Web Interface is designed for users that don't want to install any application or want to use it in any place. Also using browser may provide more anonymity, especially using incognito windows - in that case there is high probability that nobody will find out information about user problems. Web Interface is equivalent of desktop interface. It also provides simple interaction with minimum demanded steps to receive answer. Main target was to create website that is simple, clear and easy to use for everyone.

Mobile Interface

Overview

Mobile User Interface version of this project may be considered as alternative version of the Web User Interface, designed for small devices like mobile phones, netbooks or tablets. It contains the same functionality as Web User Interface and is created according to RWD recommendations. Main functionality is based on command line input, where users can describe their problems and as a result receive PDF ïňĄle from server. File can be stored in user default location or displayed in new tab using default PDF reader - it depends on browser settings.

Advantages and issues from Mobile User Interface

Main target was to create interface, which is clear, easy to use and scalable due to target group of devices. In current times almost everybody has mobile device that can be connected to the Internet. Because of usually they are personal, user can enter the mobile website and search for information anonymously. Moreover, interface is also designed for people looking for information about help for their friends or family. The only action that user is expected to do is describe nature of the problem - drug types are automatically recognized from the input. There are no hidden functions or options and all actions are visible for the user (e.g. when PDF file is ready to download). Also to receive answer for the request, no contact information are needed. User can retrieve file directly from server and in addition helpline numbers are displayed. There is no interaction with user's file system - all files can be downloaded only if users will accept it. Our

idea was to create mobile version of the website that contains the same functionality and similar layout. Used colors of the mobile interface are the same as web interface - they are toned to give the impression of serenity - no sharp or drawing attention ones. Moreover website does not contain any special graphical effects, transitions or even advertisements in order to minimize response time and avoid possibility of lack of support. All visual effects are using browser's native functions, which also makes it compatible with every mobile device. All width and height values in Mobile Interface are given using percentages, which makes website scalable for all types of mobile devices. User experience should be the same on all types of screens and platforms.

Technical choices

Mobile User Interface have been developed in Bluefish editor with HTML5, JavaScript, jQuery and CSS3. These languages used together are powerful tool for creating websites that can act like native applications. The reason behind using this technology is compatibility with great amount of mobile devices and all mobile platforms, so with our application we can gain huge group of users. Moreover, developing application in this environment is quick and easy, and decreases time to release to market, because there is no need to implement native apps. This solution is also the cheapest one (considering costs of development) and fastest, because there is no need to place application in app store. Also it gives good opportunity to make quick updates, which are visible for everyone (in contrast to the app stores, where updates must be posted, accepted and installed by the users). Core of the website is based on HTML5 DOM. JavaScript with jQuery library handles displaying DOM objects on the website and process request to server. Layout is written in CSS3, which enables developers to change it, whenever they want - e.g. colors or fonts may depend on the device or operating system. Another reason of this choice is no need to use external plugins to display interface. It decreases loading time and amount of used resources.

Outcomes

Mobile Interface is designed for users that don't want to install any application and want to use it in any place on their small devices. Mobile Interface is equivalent of the web interface. It also provides simple interaction with minimum demanded steps to receive answer (it may be helpful in emergency situations). Main target was to create responsive website that is simple, clear and easy to use for everyone, and almost on any mobile device.

Desktop Interface

Overview

The desktop interface version of this project may be considered as an extended version of the Command Line Interface interface (see page 5). Efforts have been made on a smooth and responsive graphical interface in order to provide a significantly improved experience over the use of a CLI. The design has been made relying on simplicity, and even if it looks like a Windows 8 application, it can be run in every Windows.

Advantages and issues from the Desktop interface

The point where the desktop interface takes advantage on a CLI interface is obviously the graphical appearance. We tried to keep the design very humble and simple, in order to show the seriousness of the application. The users don't need to read detailed instructions prior to dealing with our application since the simplicity and readability guarantee understandability.

Since Windows 8 is about to take off the PC market in a few years, we decided to spoof the Windows 8 UI design. This design is typography-based, excludes superfluous graphics and instead relies on the actual

content to also function as the main UI. Elements are flat, colored live tile, which is not fancy at all and works well in a fragilized user context. We tried to respect Windows UX Design Principles[4].

For the sake of simplicity, we thought useful to link the application to others Windows applications. Downloaded PDF files are opened with Adobe PDF Reader, and phone calls to Drug Help Centers can be done with Skype.

Since users may be fragilized by the use of drugs and our desktop application behaves a bit like a website (previous button, pages), we decided good to follow WCAG 2.0 Guidelines in order to gave the best accessibility[5]. This application is compatible with current Windows and also integrate well in Windows 8 design. User navigation is simple and can be done with keyboard. Big font sizes provide understandable and readable text. User can still change options at any time if he thought he made a mistake. Also, text can be traduced to any language if needed.

We thought also that big buttons and big font size are particularly adapted to tablets. Recent Microsoft Surface tablets are consequently able to display our application without losing accessibility and usability.

Shneiderman [1], Nielsen [2] and Tog's [3] heuristics are respected. The design is constant, feedback is given to user (visually or textually, sometime both). Errors are handled and detailed to the user, who cannot be lost inside the application thanks to displayed application status and previous page button.

Technical choices

The desktop application have been developed in Visual Studio 2010, with C# and XAML, within the WPF framework (Windows Presentation Framework), thus is only for Microsoft Windows platforms. The reason behind this choice is basically aesthetic. A development in Java could have been made to target more platforms, but WPF provides perfect integration in Windows and development time is faster if Windows is the only target platform. Moreover, Windows is still the most used operating system on the market, and we can consider Linux users to have healthier behaviors and to prefer a CLI interface.

The WPF framework is a great way to separate the View and the Model for a program, because both are in separated files, respectively `.xaml` files[1] and `.cs` files. This let the development team focus on interface while no worrying about the code behind, still providing a MVC-like structure[2]. We used both Microsoft Visual Studio 2010 and Microsoft Expression Blend 4 to design the UI, the first being more a IDE and the second being more a UIDE[3].

We chose to download and save files in a directory the user cannot choose. The reason of this choice is providing a history of the previous downloaded files, and still providing access to these files. We can suggest a fragilized user may not have his hard drive tidy, and is often subject to losing files. Keeping the downloaded files in a repertory within the application allow the user to avoid this. We, of course, let the user choose if he wants to keep the files. The kept files are encrypted, to provide more security. Also, the download is done in a separate thread, and progress is indicated by a circular progress bar. This gives the possibility of still using the interface while downloading.

Outcomes

The Desktop Interface has been made while thinking to more casual users than a CLI. The Desktop application integrates well in

[1] XAML is a Markup Language

[2] Called MVVM (Model-View-ViewModel) for WPF applications

[3] Development environments:
IDE: Integrated development environment
UIDE: User Interface Development Environment

Windows user interface, and provides also links to others Windows applications. This makes the flow very smooth for a drugged user, who may need to be guided and to have simple and clear information.

Command Line Interface

Overview

The Command Line Interface (CLI) version of this project may be considered a reduced port of the Desktop Interface version for CLI, towards users not desiring to use any graphical environment. Functionalities are stripped down to essential. User is able to open a browser to the web interface, to an information website and to send a request to the server and get the informative PDF file he requested. Furthermore, the CLI allows opening this PDF file with default PDF reader.

Advantages and issues from the Command Line Interface

A important point in the design of this interface, especially regarding the very nature of targeted users, is ease-of-use. Every action has to be handed out in a simple way, with no hassle. For the sake of ease-of-use, every interaction the user may have to do is unequivocally written on screen. There is no hidden command, and the state of the program is visible (for instance when a PDF file has been downloaded).

Regarding control, which is somewhat considered important for CLI users, the principle of total clarity is once again put forward. Every interaction with the File System or the system itself must be displayed on screen. As a matter of fact, address of the download directory is explicited. Moreover, every interaction with an external software (PDF reader, web browser) is announced by the program.

Multi-tasking is not usual on CLI, thus multi-tasking operations here are handled by external software. As explained previously, every departure from the terminal is explicited, so there should be no surprise when using the interface.

Of course, speed is an inner quality of this interface as no graphical element is displayed, nor are fancy features. The bare minimum of functionalities of this piece of software grants the system all freedom to deliver full speed to the interface, and resource consumption is made anecdotal.

However, remote access to the application, which could be expected from a CLI program, is not implemented here. The reason behind is quite straightforward: most results from this software are to be displayed on graphical interfaces. Thus some technologies like X forwarding or tunneling would come into play, which would put the application out of its objective of simplicity towards fragilized users.

Technical choices

The CLI interface for the Anonymous substance abuse counseling e-service is completely written in Java. The major reason behind this choice is that the development team shows great proficiency in this language and that only a small portion of development time has to be dedicated to this interface. Furthermore, deployment and testing of this interface are really made easy by the multi-platform aspect of the Java language.

Nevertheless, a drawback from the use of Java is that there is no real way to "refresh" the screen, that is to say to empty existing lines of text and replace them with other ones. The hack to give a similar feeling is to write some empty lines in order to make former text disappear - CLI being a drop-down text interface. This is quite unclean, of course, however it also offers a kind of *history* functionality as a user may scroll up in order to see old messages and states from the program.

The ability to choose in which folder PDF downloads should come may have been offered to the users. This required a dash of reflection from the development team, but it was even-

tually decided that the user should not have this choice in the CLI interface. This in order to get the simplest interface possible, with no fancy feature and no extra.

Finally, possibility is left for enthusiast programmers to embed this interface into their own Java software, may it be graphical or not. Indeed, input and output streams are specified programmatically when the CLI interface is launched. As a result, fellow software developers only have to specify which streams they want the interface to interact to and from.

Outcomes

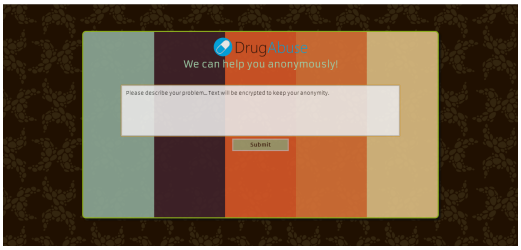
The CLI offers a drastically simpler way to consider the interface of the Anonymous substance abuse counseling e-service. The really small amount of features provided as well as the inherent speed of the interface makes a drastically peculiar interface, reserved for specific needs and full-speed connoisseurs.

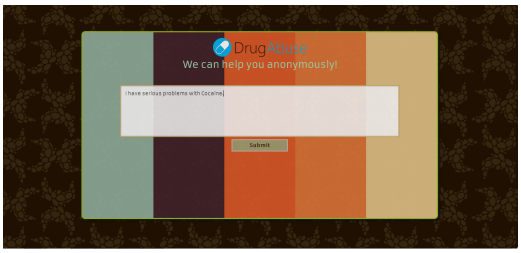
Conclusion

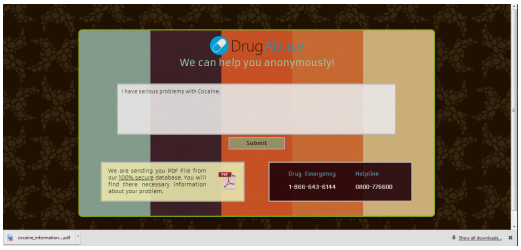
For an application of this kind - very simple and anonymous - there is no need to fancy functionalities. That is why a desktop application is not really appropriate. Eventually, the website would be the optimal way to present a drug abuse information service, because it can work everywhere on every device, and its responsiveness is adapted for smaller screens as well as bigger screens.

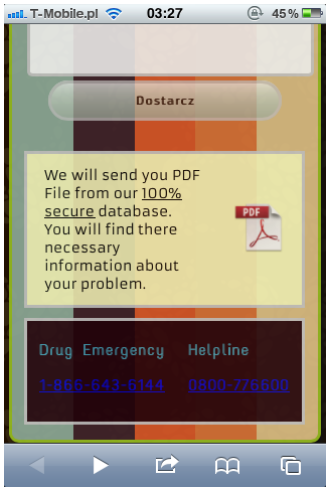
Appendices

A Web interface screenshots

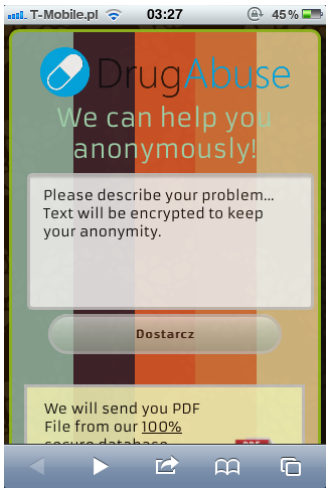




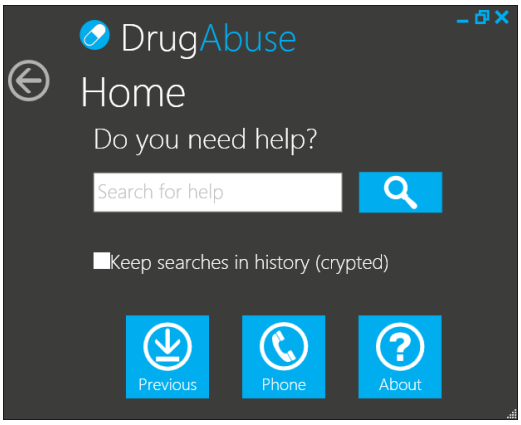


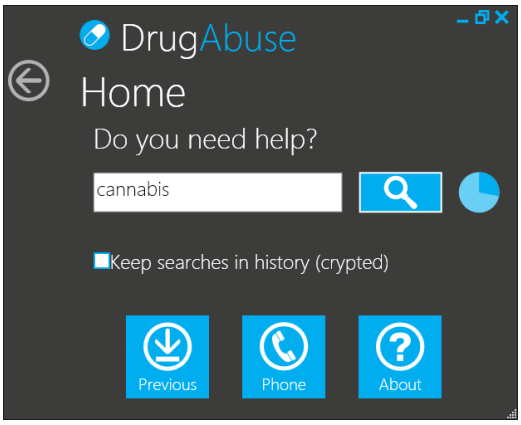


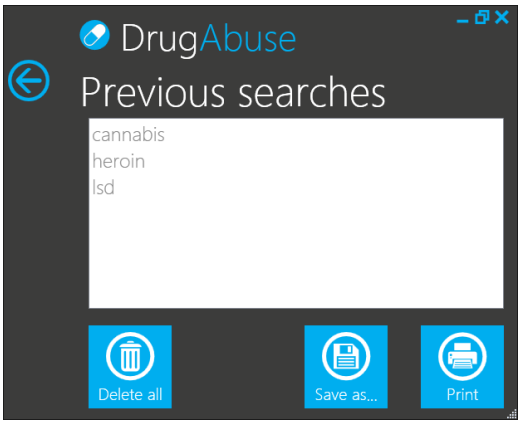
B Mobile interface screenshots

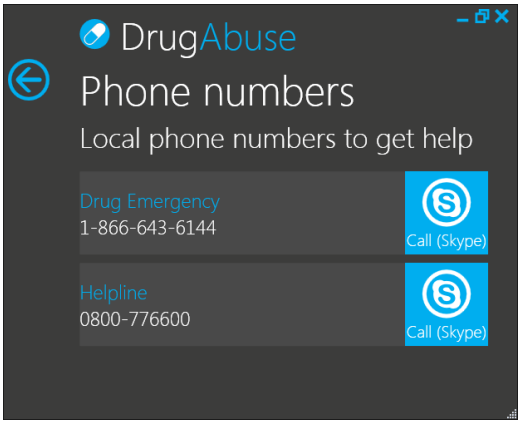


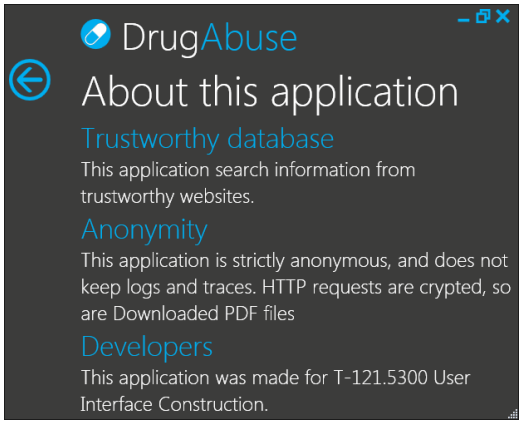
C Desktop interface screenshots











D CLI screenshots

```
                Anonymous substance abuse counseling e-service
                        Command-line interface

        What do you want to do?
[1] Write a written request to Anonymous substance abuse e-service
[2] Open browser to Anonymous substance abuse e-service website
[3] Open browser to Centers of Disease Control and Prevention
[4] Exit

        Press a number key between 1 and 4
1
Please write your request as you feel it. There is no special keyword, however
important to specify which drug is causing you trouble.
cannabis



        A file concerning your problems has been successfully downloaded to
TEMP/cannabis_information.pdf
[1] Open file with PDF reader
[2] Go back to menu
[3] Exit

        Press a number key between 1 and 3
1
Please wait while the downloaded file is opening...
```

References

[1] Shneiderman, S. B., & Plaisant, C. (2005). *Designing the user interface 4 th edition.* Pearson Addison Wesley, USA.

[2] Nielsen, J. (1994). *Heuristic evaluation.* In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY

[3] Tognazzini, B. (2003). First principles of interaction design. *Interaction design solutions for the real world, AskTog.*

[4] Microsoft Windows UX Design Principles. Available on the Web: http://msdn.microsoft.com/en-us/library/dd834141%28v=MSDN.10%29.aspx

[5] Web Content Accessibility Guidelines, 2.0. Available on the Web: http://www.w3.org/WAI/intro/wcag20