

USER INTERFACE AGILE DEVELOPMENT AND EVALUATION OF THE IKNOW STUDENT SERVICES SYSTEM

Ivan Chorbev

Marjan Gusev

Dejan Gjorgjevikj

Sasko Ristov

FCSE

FCSE

FCSE

FCSE

Skopje, R. of Macedonia

Skopje, R. of Macedonia

Skopje, R. of Macedonia

Skopje, R. of Macedonia

ABSTRACT

This paper describes the agile user interface design and development of the iKnow student services system. The evaluation of the system was thoroughly performed by using a detailed questionnaire given to multiple users. The results of the evaluation are presented.

I. INTRODUCTION

Electronic student services systems are complex information systems. They encompass multitude of functionalities provided to numerous users with different access privileges. Also, the users have different levels of familiarity to the system and the underlying processes. The user interface and its performance is essential for the system's adoption as much as the systems reliability and scope of functionalities.[5],[6] Agile development of a user interface is necessary because of multiple reasons.

- because of changing requirements [4]
- because of user satisfaction
 - o constructive remarks by experienced users
 - o habits gained by using legacy applications
 - o trial period experiences
 - o upgrade of technologies during development

II. USER REQUIREMENTS

Users demanded several features important for them:

- grouping of functionalities on fewer forms
 - o easy access
 - o decreased need for navigation (Fig.1)
- Screen size and resolution (conflicting) [2], [3]
- Impossible to make an error [1]
- Automation of processes
- Simplicity

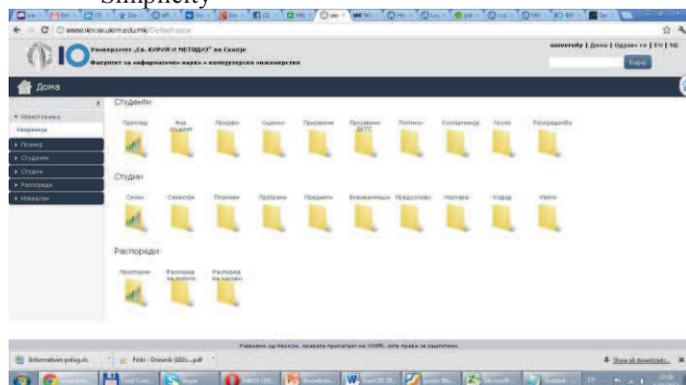


Figure 1: Homepage – navigation through standard top and side menus or folder like structure in the middle of the screen

Users tend to avoid leaving the current page that they are working on, due to time-consuming navigation. Therefore the forms were heavily enriched with popups that could present additional information or provide means for inserting/updating data [7]. Standard web page popups were avoided as deprecated, and instead, hidden forms existing on the same web page were used (Fig. 3). Although effective in terms of work efficiency and user satisfaction, such forms tend to exponentially grow in complexity and size. The side-effects of such policies are complex forms prone to higher number of errors and extreme difficulty to test and validate. Server side preparation of such forms presents a serious load on both the application and database server. Additionally, the size of the content itself is a burden to the internet link generating significant traffic. Since the demands are conflicting, a balance had to be reached for effective, but still lightweight forms that could be easily navigable. The advantages of AJAX and caching were heavily used to achieve the desired goals. Fig. 2 shows a form in which students choose the courses they will enrol during the semester. Multiple calculations are performed in real time concerning the allowed courses that the student should take, the courses the student must repeat, the ECTS credit limitations in the semester, the financial implications of the selections, etc.

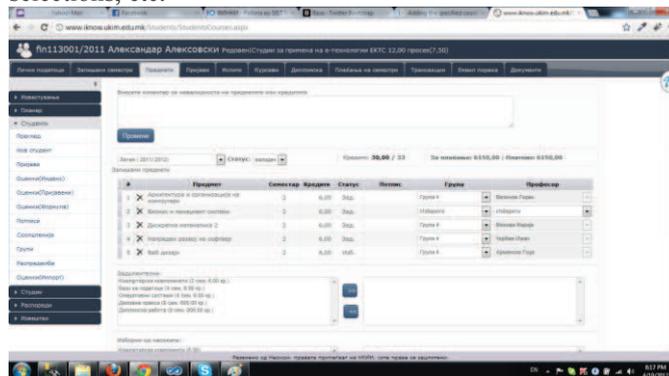


Figure 2: Courses selection per semester, a backend complex form rendered easy to use

The screen size of the monitors used by the employees in the student services department varied starting from 15” up to 22”. Therefore an adaptive interface had to be developed that is capable of using the benefits of large screens and resolutions, in the same time avoiding rendering small monitors useless. The content had to be visible and the interface usable in all screen sizes. Additionally, management used pads mainly for reporting, adding another layer of

complexity, making the interface workable on touch screens and appropriate screen resolutions.

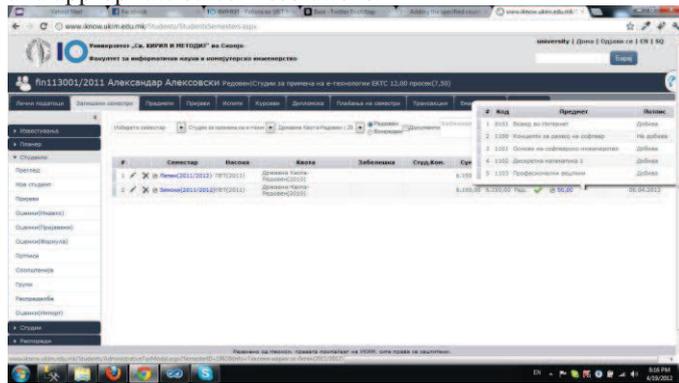


Figure 3: Overview of student semesters, expandable as needed with modal popups with additional data for SMS payments, courses enrolled in the semester, etc..

Although training is always thorough and unavoidable when deploying a complex information system such as this one, users tend to learn the system by “trial and error” and intuition. The user interface had to provide clear and short labels and messages, avoid easy deletion, and lead the user through the business processes modelled. Simplicity is a key issue (Fig. 4).



Figure 4: Modal popup for SMS payment automation

Efficient use of the workforce in the student services department is only possible if the processes are as automated as possible (Fig. 5).

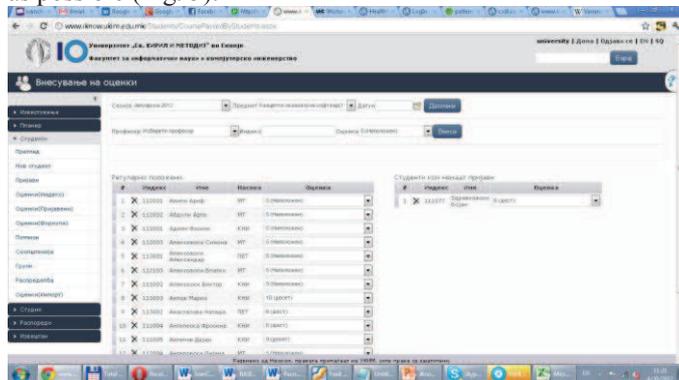


Figure 5: Bulk importing and fast input of grades for exams passed.

III. QUESTIONNAIRE

The questionnaire consisted of 33 questions. Each question was answered by the users with a grade of 1-5. The user could answer with a 0 if he/she had no opinion on the matter. The questions were as follows:

Accessibility

1. How do you judge the information about the launch and the training you received?
2. How do you access the process of registration?
3. Does your browser display all information correctly?
4. Is site load time appropriate to content and response?

Layout

5. Text-to-background contrast
6. Is the font size and style easy to read?
7. Does the site have a consistent look and feel?
8. If you have a disability regarding your eyesight: Is the content readable?
9. Is the label location and format consistent?

Navigation

10. Are the major parts/menus of the site directly accessible from the main page
11. Are the navigation labels clear and descriptive?
12. Is the workflow navigation consistent and easy to identify?
13. Is the respective location within the process (site) transparent?
14. Is the site search easy to access?
15. Is the exit point clear on each page?
16. Does it require minimal steps in sequential menu selection

Exception and status handling

17. Are the messages regarding status clear and descriptive?
18. Are the messages regarding exceptions/errors clear and descriptive?
19. Position of messages on screen is good

User guidelines and online help

20. Is the site designed to require minimal help and instructions?
21. Is the help and instruction information easily accessible?
22. Is there an easy channel available to communicate with an administrator?

Learning

23. Easy to learn to operate the system
24. Easy to explore new features by trial and error
25. Easy to remember names and use of commands

Quality and structure of information

Content and Efficiency (refers to all information i.e. field explanations, order of fields)

26. Is the content understandable?
27. Is the content well-structured and correlates to your requirements?

- 28. How do you evaluate the support of the system?
- 29. Do you observe an increase in efficiency?
- 30. Does the system provide a sufficient number and quality of reports?
- 31. It is easy to use.
- 32. What is your overall evaluation of the system?
- 33. Do you have any further comments?

IV. EVALUATION RESULTS

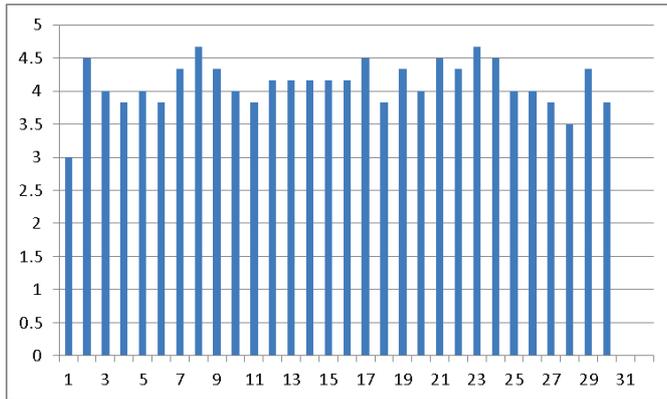


Figure 6: Average grades given by the users in the questionnaire.

Fig. 6 presents the average grade awarded to each question by the users, while Fig. 7 presents the variance of the average grades.

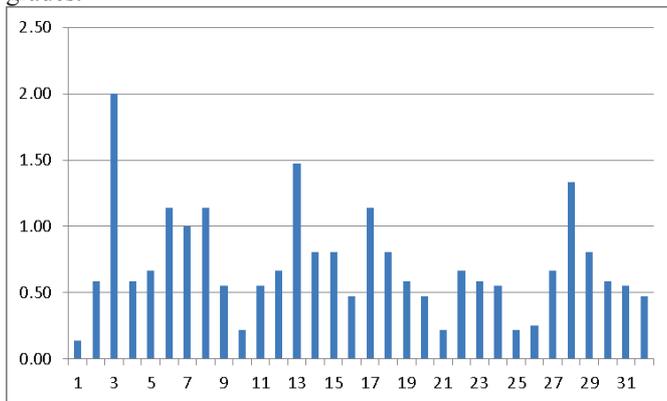


Figure 7: Variance of average grades given by the users in the questionnaire.

The user satisfaction is evidently high, due to their participation in the user interface fine tuning and the efficient interface that resulted.

V. CONCLUSION

Making a complex student e-services information system is a challenge by itself. Making its user interface simple to use and learn, as well as efficient, adds another equally important layer of difficulty. In this particular project, due to the careful approach, detailed and flexible initial projects specification and adequate agile development, the conflicting business and user demands were achieved in great part. There is always

room for corrections and upgrades and it is expected that they will be carefully implemented. This is especially true in systems like this where requirements constantly change.

REFERENCES

- [1] Anthony Vance, Braden Molyneux, Paul Benjamin Lowry, "Reducing Unauthorized Access by Insiders through User Interface Design: Making End Users Accountable", Hawaii International Conference on System Sciences, January 2012, pp. 4623-4632
- [2] Jörg H. Mayer, Timm Weitzel, "Appropriate Interface Designs for Mobile End-User Devices--Up Close and Personalized Executive Information Systems as an Example", Hawaii International Conference on System Sciences, January 2012, pp. 1677-1686
- [3] Gang Huang, Daimeng Wang, "Adapting user interface of service-oriented rich client to mobile phones", Service-Oriented System Engineering, IEEE International Workshop on December 2011, pp. 140-145
- [4] Lu Xudong, Wan Jiancheng, "User Interface Design Model", Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Eighth ACIS International Conference on, July 30 2007-Aug. 1 2007, Volume: 3, Page(s): 538 - 543
- [4] Massila Kamalrudin, John Grundy, "Generating essential user interface prototypes to validate requirements", Automated Software Engineering, International Conference on, November 2011, pp. 564-567
- [5] Shinichi Inenaga, Kaoru Sugita, Tetsushi Oka, Masao Yokota, "Performance Evaluation of User Interfaces According to User Computer Skill and Computer Specifications", Intelligent Networking and Collaborative Systems, International Conference on, December 2011, pp. 446-449
- [6] Shinichi Inenaga, Kaoru Sugita, Tetsushi Oka, Masao Yokota, "A Preliminary Evaluation for User Interfaces According to User Computer Skill and Computer Specifications", P2P, Parallel, Grid, Cloud, and Internet Computing, International Conference on, October 2011, pp. 295-298
- [7] David R. Karger, "Creating user interfaces that entice people to manage better information", Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM '11), October 2011, pp. 1-2