

The Cost of Designing User-Friendly Web Applications

Boris Bankov

Abstract

In recent years web applications have become more and more complex on the inside in order to better suit the needs of their users. Thanks to the increase in quality of available hardware systems, the performance aspect of computer software has been put a bit in the background, while code clarity, usability, flexibility and robustness are discussed frequently. In this paper the cost of designing user-friendly web applications is analyzed based on database complexity, server load, business logic security and browser performance. Several problems of balancing design approaches aimed at cleaner user interface are discussed. The study is conducted as a part of a scientific project to gather statistical data about academic activities, implemented in University of Economics – Varna.

Keywords: web applications, design patterns, user-friendly, server performance

C88 Other Computer Software

Introduction

Designing and developing user-friendly web applications does not always mean writing robust and quality code. Even though absolutely every big project is thoroughly tested and reiterated numerous times to try and fit necessary requirements, meet deadlines and have a smooth launch, it is nevertheless a great challenge to follow programming standards and please users at the same time. Nowadays a viable strategy is to leverage the computing power of the devices that run those projects and build a functionally-rich and easy to use applications. Evaluating software and determining the price of programming has been a field of study for a long time (Farr & Nanus, 1964), (Boydston, 1984), (Zhang & Zhang, 2009). In this paper however we will look at the cost of designing a web application in terms of resources and performance. The discussion presented here is a result of our work on a university-based project, the goal of which is to create and maintain a web application that academic staff can use to publish reports and create plans about their scientific activities during each school year. Following the trends emerging from creating different social platforms and the retrieval of statistical data from them has given us enough reason to modernize the way we collect information about academic activities within the University of Economics – Varna¹. The main objective while designing and implanting the aforementioned system is to make it as easy to use as possible, while maintaining efficiency. Furthermore because this is a university intellectual property it is hosted on a private server alongside other such software applications, so we are taking into consideration server resource sharing with other projects and systems.

1. Project details, expectations and plan of action

To best explain the design decisions that we took in order to build a fluid and user-friendly web application it is necessary to talk about the project itself – how data has been handled in the past and what purpose will the system serve.

On a national level in Bulgaria, academic staff in universities is required to submit reports about their research and activities each year. The information is used multiple times for different procedures – university ranking, career development and promotions, accreditation of bachelor, masters and doctorate degree programs, etc. In order to motivate university lecturers and lab assistants to continue writing papers and participating in scientific projects in University of Economics - Varna they are also required to present a plan for their activities throughout the next academic school year.

Reports are split into 8 different subject areas, while plans are divided into 5. The sections are presented in Table 1 and Table 2.

¹ Todoranova, L. The public sector in a “cloud”?. Scientific works of Russe’s University, 54, 2015, p. 139.

Table 1. Report sections

№	Section	Description
1	Publications	It is required of academic staff to submit full description of all their scientific papers published within during the last school year.
2	Citations	It is required of academic staff to submit any new citations of their published works, discovered within the past school year and not reported for previous periods.
3	Editorial boards and reviews	Staff that participate in editorial boards as well as write reviews outside of editorial boards for different purposes need to submit information about the journals and works they review.
4	Projects	Staff that participate in projects with international, national or university funding are required to submit information about their activities, project duration, scope.
5	Consulting	Staff that offer consulting services are required to submit information about the organizations they work with and specifics about the services.
6	Events	Staff that participate in international, national or university held events are required to submit information about the topic, place and date of the events.
7	Academic career	In this section staff submit any qualification improvement classes that they take, any specializations, PhD students that they supervise, successful career developments, e.g. becoming a Professor or finishing a dissertation, etc.
8	Activities	In this section staff submit any positions won after an election, e.g. head of faculty, head of department, etc. and specific tasks issued with an order from the University's Rector.

Table 2. Report sections

№	Section	Description
1	Publications	Staff are invited to submit a rough estimate of how many papers they plan to write and publish during the upcoming period.
2	Projects	Staff are invited to submit if they plan to participate in any projects for the upcoming period
3	Events	Staff are invited to submit any events they plan to visit during the upcoming period.
4	Qualifications	Staff are invited to submit any qualifications courses they plan to take for the upcoming period.
5	Specializations	Staff are invited to submit any specializations they plan to do for the upcoming period.

In the past years reports and plans were being submitted via physical paper copies based on a specific form template. This method had some flaws. One common issue was when the information supplied was more than the prepared blank spaces in the form and academic staff had to go outside the line to write the full text of the reported activity. Another issue would occur when the written information could not be read due to bad calligraphy. The biggest problem was, however, that the supplied data after being submitted had to be brought into a digital environment to be used in spreadsheets to quickly sort, filter, count and draw subtotals for different purposes, create inquiries, run statistical analysis, etc.

For the previous period of 2016/2017 reports and plans were submitted online via Google Forms. This procedure was a trial example in preparation for the development of a more sophisticated and specialized software that will be used in the future. Roughly 250 (252 to be precise) individuals from the academic staff of University of Economics - Varna submitted data that due to the template and limitations of Google Forms ended up in a single spreadsheet. Afterwards the document was filtered based on department and faculty and then sent to the Heads of each department and Deans of faculties. Some issues were resolved however the main concern at that time that data supplied on some occasions would not be in the correct format, e.g. field that should contain numbers would also have words or abbreviations; in others instead of names and titles which should be exclusively available were left blank or skipped. Due to the fact that Google Forms does not offer additional functionality in terms of hints, tooltips and advice when filling the reports, most users did not manage to fix any mistakes before the data was reviewed. When trying to optimize the data that could be retrieved from the end inquiry with all reports it became clear that on their own users tend to write names of places or institutions in a few but not too many different ways. This would cause unnecessary problems when trying to filter and sort the document by that specific table column. So, with that in mind our project had to be designed in such a way that would:

- 1) enforce strong data format on certain fields, so sorting would work properly;
- 2) enforce all required fields to be filled;
- 3) allow data to be edited, with enough hints towards trial and error;
- 4) help users with autosuggesting relevant tooltips from already submitted similar titles, names or other data in the same field;
- 5) give everyone the freedom and tools to make a quick inquiry about different report sections with the necessary information in a proper spreadsheet.

Focusing on those points as well as designing and building a user-friendly web application becomes a difficult task. Some design decisions are based on our work on another project is used currently in University of Economics – Varna: a registry where all academic staff members submit full descriptions of their published scientific works. Our general strategy was to keep the user interface as similar as possible between the two systems, which undoubtedly will help build a positive experience². The restrictions and limitations on one hand are aimed at producing better end results that will help with statistical analysis. At the same time if the platform is not intuitive enough and presents users with not enough control over their reports, its capabilities will quickly be distrusted. Balancing these requirements towards the web application is going to come at a cost – performance wise both for the server and the client (user`s) side.

2. Design philosophy

The concept “design” has to be interpreted in the context of a process of building a comprehensive concept of a product³. When it comes down to collecting data for statistical and

² Nacheva, R., Sulova, S. Approach to exploring users' expectations of digital services' functionality. *Ekonomiczne Problemy Usług*, Szczecin: Wydawnictwo Naukowe Uniwersytetu Szczecińskiego, 2, 2018, p. 138.

³ Nacheva, R. Principles of User Interface Design: Important Rules that Every Designer Should Follow. *Izvestia, Journal of the Union of Scientists - Varna. Economic Sciences*, 2015, p. 140.

business analysis, forms are usually how those tasks are completed. Developing the aforementioned web application revolves mainly around constructing and presenting forms in the best possible way. Translating a physical paper form into a digital online one gives us a number of things to consider when designing the user interface. The aspects of our system that we find most viable to tweak in order to give the best user experience are:

- a) database complexity;
- b) business logic security;
- c) browser performance;
- d) server load.

Database complexity refers to the use of additional tables and primary-foreign key relations beyond the general requirements for storing all data that comes through the web forms. Those additional tables contain lists of items that can otherwise be kept statically inside a template web page file. An example of one such table is “phd_programmes” which has a list of different doctorate programmes that University of Economics – Varna offers to PhD students. We chose to do this in the event of needing to expand the details that are required while filling out the report forms. So instead of querying one table to retrieve the records our system would join two tables (See Fig. 1). Nevertheless a good example of set principles to follow when designing a database is always to normalize it, make it so it scales up when the application is improved up and simplify data management.

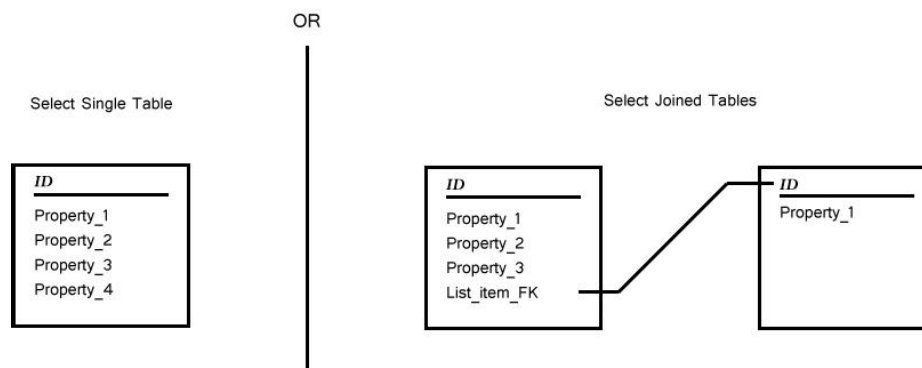


Fig. 1. Querying unnormalized vs normalized tables

The question here is: **does table normalization affect user experience?** In general users prefer choosing items from a list rather than typing them manually. Lists can also be translated into filter options later when the data is stored and retrieved back or exported into a spreadsheet. Static lists would have to be changed everywhere in the code if an item is edited, removed or if a new one added.

Business logic security is a subset of the system security. However due to our project being locked behind a log-in form and all data is only available to a selected few from the administration staff and the heads of the departments and deans of faculties, the web application is only vulnerable from within. We label all data submitted through the aforementioned system as private. Private/confidential data can have moderate adverse effect on a University’s reputation, resources, services or individuals. Our focus is on trying to prevent anyone from accidentally or purposely attempting to change someone else’s data⁴. This is the case due to the fact that every user is only held accountable for his report and plan, but also represents the institution or University of Economics – Varna. However some information submitted by academic staff is used as autosuggest content on specific fields in forms in different sections. This means that the longer our system is used, the better the user experience will be due to the fact that users will have more options to choose from and spend less time manually inputting titles, names, etc. The use of autosuggested

⁴ Kuyumdzhev, I. Controls Mitigating the Risk of Confidential Information Disclosure by Facebook: Essential Concern in Auditing Information Security. TEM Journal, 3, 2014, 2, p. 115.

content in our project also comes at a cost.

Sometimes a mistake can happen, spelling error or a wrong number like a date can enter our database. To prevent bad input from spreading across multiple users' reports and plan we had to design the application to properly deal with such errors. The solution: **every time a user submits data, new or old, through create or edit, any previous records of his in the same scope (section) are deleted (see Fig. 2.).**

Currently we use one view or one web page to display the form from a single section in both the states *create* and *edit*. After a user submits for a first time their report for a specific section e.g. Events, the same view is shown. Other users can use a dropdown autosuggest that will list any Events submitted by other users.

If someone wants to edit a specific section in his report, the system will take all his previous records from the same section and try to delete them before storing them anew. Deletes won't happen cases when records are shared, e.g. many lecturers opted-in and chose the same Event, which they in fact attended. In this scenario, only personal details would be recreated, while general information will remain unchanged.

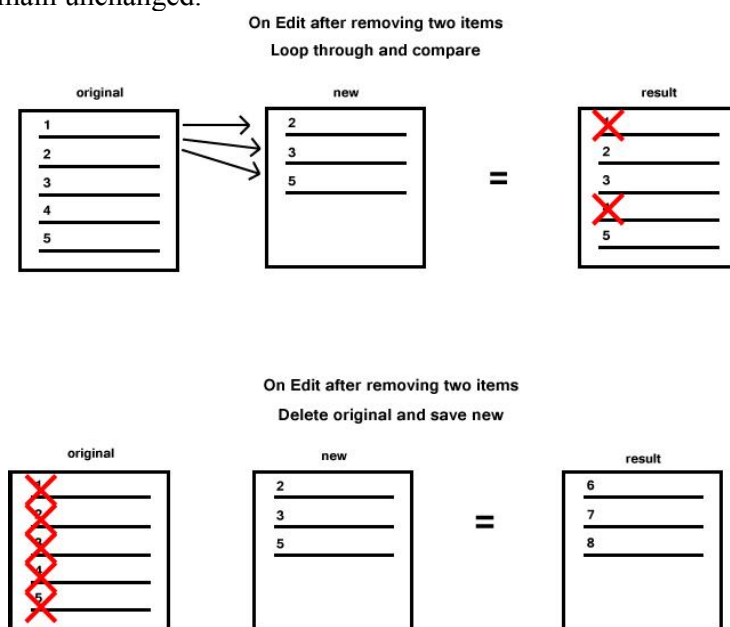


Fig. 2. Comparing session stored model of a table versus post data and mass delete and create approach

How does this translate to system operations? In the above scenario we execute one delete query and one create query on the database. On the business logic layer we do not check if the user has submitted data that exists in the database, has edited and/or deleted anything from the last state his data was in. This of course could be done with fewer actions performed on database level. However to accomplish that one will need to have records stored in memory (session) and then compare against the data that the user submits, which adds more complexity to the business logic level. Here we chose to leverage the power of MySQL database engine and ignore old data states.

Browser performance in the context of the developed web applications refers to how business logic is handled on the client's side and on the server. Here we have two different problems to balance. On one hand we feel user experience is smoothest if web pages refresh rarely when creating, adding, editing or removing items from the forms. While normally posting data would cause a page reload, we moved towards using AJAX⁵ for the majority of operations, which will keep the state of the page unchanged on saving. Pages are refreshed only after completely

⁵ Asynchronous JavaScript and XML

resetting a section or switching to another section. Very old computer configurations can have a moderately harder time when trying to do asynchronous requests, due to the fact that JavaScript is executed on the client's side (in the user's browser). AJAX can have a bigger effect when autocompleting and autosuggesting happens, because more data is injected and loaded on run-time⁶. Earlier in this paper we mentioned that in certain fields users will be able to get an autosuggestion about a title of a publication, name of a place, event, a PhD student, etc. Apart from helping academic staff members fill out their reports faster, autosuggesting also saves up database space, as less duplicate records are being created.

In some sections e.g. Citations, we chose not to implement autosuggesting due to the sample size. Currently in our database we keep information about over six thousand publications. In other report sections, publications can be autosuggested, but they are sampled from the user's own list. Citations however come from publications written by other academic staff, which means when querying the database, if a broader term is used, then a larger subset would be returned to the user's browser. In our experience this slows significantly the whole process in the cases when a user has to report a large number of citations of his works.

Server load in some sense depends on all three previous aspects we mentioned so far and it can be explained as the amount of operations added on top of the general infrastructure of our system. In the majority of sections, forms have the option to iteratively add new fields to allow the submission of multiple items at once, e.g. submitting data for two Events instead of one per page or per create action. This theoretically means that if a user wants he can create a form specifying that he participated in an infinite number of Events, he can do so. To allow such behavior from our forms, we are loading asynchronously additional fields from a different template for each section. As a result, every new set of input fields is another request sent to the server.

Server load also comes into play when we are exporting the data to MS Excel files. Currently we have employed a regulation, where reports and plans can be exported for departments and faculties per section in different spreadsheets and not as one whole file. Although this can be fixed by having the system hosted on a more powerful server, we also believe that not all data that the application generates is going to be needed at the same time for multiple purposes and thus rendering the process ineffective – why wait for a larger file to be generated and then downloaded, when you need only a specific set of information. At this moment we can agree that both the user and the server that runs our application are favorable of less downtime for processing requests, such as exporting stored data.

3. Results from a three week exploitation run.

The system was launched on 27th of September and now after 3 weeks we can present some data about the usage. The deadline was 20th of October. Here are some details in numbers:

- In the three week period 259 new publications were added to the database, with 114 of them being submitted in the last three days.
- A total of 198 academic staff members have created their reports and plans.
- A total of 685 publications have been published for the period of 2017/2018.
- Citations amount to 695 reported.
- Academic staff members participated in 95 editorial boards over the period of 2017/2018 and have written 96 reviews outside of editorial boards.
- Users have participated in 149 unique projects. 204 unique participations have been submitted. This means 55 of those participations are filled out using autosuggest.
- Consulting services amount to 126.
- Academic staff members have participated in 292 unique events. 399 participations in events have been submitted. This means 107 are filled out using autosuggest.

⁶ Petrov, P. Methodological problems in the creation of web applications running in real time, Izvestia, Journal of the Union of Scientists - Varna. Economic Sciences, 2015, p.100

- 116 courses for improving academic staff members' qualifications were taken, while 16 specializations were held abroad.
- Lecturers have reported working with 107 PhD students for the period of 2017/2018.
- 16 academic staff members reported successfully completing a doctorate and earning PhD title.
- 381 publications are planned for the upcoming period.
- Academic staff members plan to participate in 115 projects.

Over the course of these 3 weeks there haven't been any major inconsistencies and problems encountered by the academic community of University of Economics – Varna, while working with the system. There are some suggestions about wording or phrasing of some helper texts on a few of the sections, which will be discussed and fixed.

Conclusion

Deploying a web application nowadays can be done in many ways. Different methodologies exist to give programmers strict guidelines to follow and implement systems effortlessly. On some occasions specific aspects of software development must be taken into consideration. When it comes to building a web application around the notion that user-experience is key to the success of a project, some interesting design decisions must be taken. They can be broken down to database complexity, business logic security, browser performance and server load. From a user's point of view an example of a system is an absolute and cannot be compared to properly to different iteration of the same functional scheme. Handling the client and server side infrastructure has to be done in such a way that users do not experience any design flaws. The web application we build for University of Economics – Varna is likely to be extended and build upon to provide even better service at a minimum cost for its users performance-wise.

References

1. Boydston, Robert E. "Programming cost estimate: Is it reasonable?." Proceedings of the 7th international conference on Software engineering. IEEE Press, 1984.
2. Farr, Leonard, and Burt Nanus. Factors that affect the cost of computer programming. No. TM-1447/000/02. SYSTEM DEVELOPMENT CORP SANTA MONICA CA, 1964.
3. Kuyumdzhiev, I. Controls Mitigating the Risk of Confidential Information Disclosure by Facebook: Essential Concern in Auditing Information Security. TEM Journal, 3, 2014, 2, p. 113-119.
4. Nacheva, R. Principles of User Interface Design: Important Rules that Every Designer Should Follow. Izvestia, Journal of the Union of Scientists - Varna. Economic Sciences, 2015, p. 140 - 149.
5. Nacheva, R., Sulova, S. Approach to exploring users' expectations of digital services' functionality. Ekonomiczne Problemy Usług, Szczecin: Wydawnictwo Naukowe Uniwersytetu Szczecińskiego, 2, 2018, p. 137 - 145.
6. Petrov, P. Methodological problems in the creation of web applications running in real time, Izvestia, Journal of the Union of Scientists - Varna. Economic Sciences, 2015, p. 97 - 104.
7. Todoranova, L. The public sector in a "cloud"? Scientific works of Russe's University, 54, 2015, p. 136 - 40.
8. Zhang, B., Zhang, R. Evaluation model of software cost estimation methods based on fuzzy-grey theory. Internet Computing for Science and Engineering (ICICSE), 2009.

Contacts

Boris Bankov, PhD
University of Economics - Varna
boris.bankov@ue-varna.bg