Re-design User-interface of Packaged Web Applications using Ext-JS Framework

Henock Melisse
celineboy@yahoo.com

Mesfin Belachew
Ministry of Communication and IT, Addis Ababa, Ethiopia
mesfinbelachew@gmail.com

Abstract

Different companies use software systems to automate their work process. Companies prefer package applications because non-package applications are more expensive and take longer time to develop. But some package solutions are not compliant with the current solutions on usability and other characteristics. One of the usability issues is the user-interface design.

Developers design the user-interface of package applications in order to make it easy for the end-users. During the development process, challenges such as picking the right user-interface designing tool, understanding the user's need and measuring the performances of user-interface designing tools occur.

The approach used to come to the solution of the problems takes the form to explore the user-interface technology of water billing system package application which is developed for Addis Ababa Water and Sewerage Authority.

In order to justify this, an interview with companies with the expertise of developing package applications has been held. Moreover, an experiment of two user-interface designing tools in order to measure the performance advantage one takes over the other has been done.

After the experimentation process, the result clearly shows that Ext-JS framework is better in performance and usage of the application than ASP.net web forms. The result from the development of the prototype shows that Ext-JS framework made the user-interface of the application easy.

After the experimentation and implementation on the water billing system prototype, it was found out that most of the enlisted problems were indeed tackled by the suggested solution.

Keywords: User Interface Design; Package Solution; Ext-JS framework; ASP.net

1. Introduction

The nature of the business world these days forced institutions to facilitate their work process with a computer system. Time and money are precious resources in such competitive environments. The choice for institutions to automate their work process is more biased on purchasing off-the-shelf package applications. Factors such as time and money will limit the company to narrow down the automation options to only getting off-the-shelf package applications. Since it is expensive and time consuming to make an arrangement with software developing companies to develop software application from scratch, institutions prefer to buy package applications.

It would be more attractive for institutions to use package applications if the usability problems such as user-interface are solved. As technology grows rapidly, the techniques to develop software also grow.

In Ethiopia, the habit of using package applications is increasing. At the same time, local and foreign software developing companies have also continuously grown into becoming the supplier of package applications. But most of these package applications are criticized by their user-interface performance and other usability issues. In order to solve the user-interface problems, the user-interface designing tool that we are choosing to develop a user-interface plays a vital role.

As the programming world widens, different vendors introduce technologies that are useful to develop software.

Programmers have grown into becoming users of different technologies. This increasing demand brings about a strong focus on the efficient development of applications. Since the Internet becomes very important issue in our lives, the demand of software in the form of web-application is also becoming important.

The main purpose of this work is to solve the difficulties and challenges in re-designing user-interface of packaged web applications; which in this case is the proper choice of the user-interface designing tool and the system architecture that best fits the application at hand.

After buying package applications, institutions end up without using the software and start to complain on the friendliness and performance of the user-interface. We can increase the performance and usability of a software by choosing a suitable user-interface designing tool. The user-interface designing tools that we use to develop package applications need to be very easy and powerful at the same time. There are many tools available to design user-interfaces. Examples would be Ext-JS, JQUERY, PHP, Silverlight, and many more, all having their own advantages and disadvantages.

2. Background

The demand for software systems in the form of web applications is increasing widely. This increasing demand brings about a strong focus on the efficient development of applications. As technology grows rapidly, the technique to develop software is also increasing. Developers find it easy to write code due to the introduction of different technologies. But the development of applications doesn't come easily. There are many challenges that developers face in the development process. Some of the challenges and reasons for the initiation of this work are:

- User-interfaces of package applications are not being developed in a way people understand, need, and want.
- Re-designing the existing architecture of package applications is difficult.
- User-interfaces of package applications have performance problems. These include performance issues such as CPU usage and memory allocation.
- Re-designing the user-interface of package applications is a challenge.

This paper presents a mechanism to solve the above mentioned problems that are encountered to re-design the user-interface of package web applications.

3. The Proposed Solution

The general objective of this paper is to re-design the user-interface of AAWSA's water billing system using a user-interface developing tool called Ext-JS framework.

The specific objectives of this paper are to:

- gather requirements form Addis Ababa Water and Sewerage Authority (AAWSA) to analyze the user-interface gap,
- re-design the user-interface of the water billing system using Ext-JS framework,
- clearly show the advantages of using Ext-JS framework to re-design user-interface, and
- introduce appropriate technology to re-design user-interfaces of package applications by improving performance and usability.

4. Experimentation (Prototype)

Before suggesting a technology, we always have to compare it with other similar technologies in order to show the preference. For this work, the experimentation is done by comparing Ext-JS framework with that of ASP.net web form. We have considered other user-interface designing tools such as JQUERY. But both JQUERY and Ext-JS are JavaScript technologies. Comparing the performance of identical frameworks may not give us the performance comparison that we are looking for. So this led us to concentrate on non-JavaScript technologies such as ASP.net web form. Moreover, during the research, we found out that many of the packaged web applications in Ethiopia are being developed using ASP.net web form.

The technique we used to compare the user-interface designing tools is by developing identical

software developed with the same technology, except their user-interface. We used SQL server 2005 for the database, Entity data framework as the data access technology, and Microsoft Visual Studio 2010 to develop both applications. We have only changed the user interface designing tool. One of the application's user-interface was designed using Ext-JS framework and the other was designed using ASP.net web form.

The following scenarios were identified for the experiment.

- Performance of CPU sampling
- Performance of instrumentation profiling
- Performance of memory allocation (sampling)

The experimentation test result shows that Ext-JS framework uses low CPU, low instrumentation profiling, and has high memory allocation, whereas ASP.net web form uses high CPU, high instrumentation profiling, and low memory allocation.

As a sample, Figures 1 and 2 show the CPU sampling which is one of the test scenarios for both user-interface designing tools. Figure 1 shows the CPU usage result of Ext-JS framework. Figure 2 shows the CPU usage result of ASP.net web form.

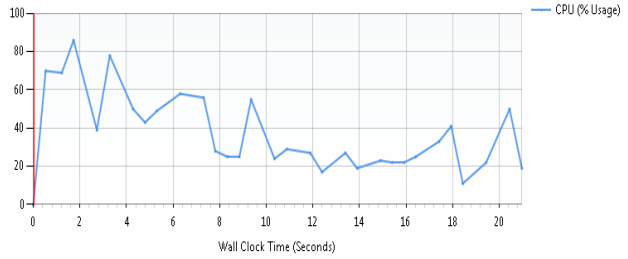

Figure 1: CPU usage result of Ext-JS framework

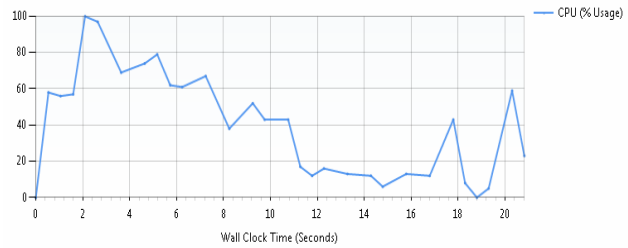

Figure 2: CPU usage result of ASP.net web form

The prototype chosen for re-designing the user-interface of package web application is the water billing system of Addis Ababa Water and Sewerage Authority. We followed the layered architecture to re-design the user interface. The main reason for choosing this architecture is because there is no need to make change on the business and application layers of the existing system; rather it is convenient to concentrate on the presentation layer since we are re-designing only the user-interface.

Layered architecture is best suited for this kind of work because it breaks down the components of the system into business layer, application layer, and presentation layer [1]. If there is a need to change only the user-interface, it can be accomplished by only taking out the presentation layer, re-designing it and putting it back without affecting the other layers.

5. Related Work

Even though it was hard to find resources that concentrate on Ext-JS framework, the research has been done by the information and useful facts gained from very few books and forums. The reason for this is that the framework is young and at the growing stage. Only few books are available about the framework; the rest is a documentation that is released by the community developers' site called Sencha library.

Ext-JS framework made the post back of data from the application to the web server very fast [2, 3, 4]. Ext-JS has an html page, called API, which is only created at run time. The API handles the post back process for Ext.

It is also mentioned that Ajax development is very crucial for web applications and that Ext-JS framework simplifies Ajax development through the use of reusable objects and widgets. Ext-JS framework was introduced by providing an overview of the object-oriented JavaScript design concepts behind it and it is shown how to use the Ext-JS framework for rich Internet application user-interface elements.

As explained in [6], Ext-JS framework is one of the user-interface development tools with a rich set of widgets and layout management tools that give full control of an application development. It is also mentioned that Ext-JS framework has six main features.

When we come to application package software, it is mentioned in [5] that early packages that focus on a single function within a corporate or

government hierarchy, such as accounting, were built using older software languages and programming techniques. This had caused application packages to become inflexible and increasingly hard to fine-tune to customer requirements. The article also mentioned that application packages must be integrated into in-house application and data architectures. As a result, in-house programming teams have had to modify application packages and this has resulted in difficulties in reintegrating vendor upgrades back into the package software. Falling behind current releases of a given package increases the challenge of upgrading packages exponentially.

6. Conclusion

This paper tries to show the challenges that exist in the effort of developing the user-interface of packaged web applications. The main purpose of the paper is to identify what the challenges were and providing a technology that solves them. The methodologies to collect data and come to conclusion included interviews, questionnaire, literature review, experimentation, and prototyping. The experimentation done shows the advantages of Ext-JS framework over ASP.net form in the aspects of both performance and usability. Water billing system was the web application selected for the case study. The user-interface of the water billing system was designed as a prototype to introduce Ext-JS framework and to show how we can easily manage it with the selected architecture of the layered architecture.

At the end of the experimentation and prototype implementation, the findings clearly show that the process of experimentation and prototyping of the web application successfully solved the problems.

As a future work, problems for user-interface re-design of package applications will be compared with other latest user-interface developing tools to come up with better results. Moreover, issues with Ext-JS framework or localization can be added as a feature for the language. Applying Ext-JS framework in mobile applications is also another concept by itself. It is mentioned that Ext-JS framework can be implemented in mobile phones through Sencha library. Since water billing system is an application used by the public, there may be a time where the application needs to be used in mobile phones.

References

[1] http://fewagainstmany.com/blog/introduction-to-layered-architecture-part-one, May 12, 2013.

[2] Shea Frederick, Colin Ramsay, Steve ‘Cutter’ Blades, “Learning EXT-JS”, 2008.

[3] Shea Frederick, Colin Ramsay, Steve ‘Cutter’ Blades, Nigel White, “Learning Ext-JS 3.2”, 2010.

[4] Jorge Ramon, “Ext-JS 3.0 Cookbook”, 2009.

[5] Cutter, “Application Package Software: The Promise vs Reality”, Vol. 6, No. 9, September 2006.

[6] Jesus Garcia, “Ext-JS in Action”, 2009.