AskFedora UX/UI and Functionality Overhaul

The AskFedora Redesign project

K.A.Welivita
Department of Computer Science and Engineering, University of Moratuwa. Sri Lanka
anuradha.12@cse.mrt.ac.lk

Abstract— **AskFedora is a well known community knowledge base and a support forum, which provides community support for Fedora users. It is powered by Askbot, a Python and DJango based web application. With time, the user interface and user experience of AskFedora had come to a point of having a comprehensive overhaul. This is to have some uniformity with the other Fedora websites and have improved usability features. The AskFedora redesign project was aimed at achieving the goal of providing better user experience for the users. The project was conducted in three main phases, which were user experience analysis and mock-ups development, coding and integration with Askbot. The designs done are mainly based on user experience research and the styles used follow concepts of responsiveness and cross browser compatibility.**

Keywords—User Experience (UX); User Interface (UI); Graphical User Interface (GUI)

I. Introduction

AskFedora is a popular knowledge base providing knowledge and support for Fedora users all over the world [1]. It is also a well known question and answer forum maintained by The Fedora Project open source community, where people can ask questions to be answered by the community. AskFedora is the primary place for community support in Fedora and it is powered by Askbot, which is a Python and Django based web application. Currently over 11,000 questions have been asked and there are over 12,500 contributors on the website [1]. As the original website used the default layout and styles provided by the generic web application Askbot, it was in need for some UX and UI overhaul to be done in order to provide better experience for the users.

AskFedora being the major support forum for the Fedora users, it required some uniformity with the current Fedora websites and needed to have a better look and provide better user experience. This has been an issue prevailing in the Fedora community for more than four years and no proper involvement in a complete redesign effort of AskFedora was to be seen. This fact has been the motivation behind this redesign project.

Today, the period where the programmers considered that only the back end of the application needs to be perfect is over. Currently numerous types of research activities are being carried out in the fields of UX/UI and Human Computer Interaction to identify means of providing better user experience. In such an era the need to provide a better user experience for AskFedora users was critical. Numerous drawbacks have been identified with the previous system, some of which are poor use of color schemes, lack of consistency with the other Fedora websites, lack of attractiveness and poor responsiveness when it scales down to mobile devices. This project aimed at providing a complete overhaul for the website so that all the above issues are being addressed and the users have a better experience with the website.

The project has provided a complete UX and UI overhaul for the website following an approach, which consisted of three phases namely UX analysis and mockups development, coding and integration with Askbot. This was also followed by a phase of testing and bug fixing. The complete overhaul followed the Rational Unified Process software development methodology and this has resulted in a redesigned version of the AskFedora website which now provides an uncluttered, responsive interface with a lot of dynamic elements for the users to interact with. It is at the same time consistent with the other Fedora websites. It also addresses the above mentioned issues and provides better and consistent user experience for the Fedora users.

II. Literature Review

Ask Fedora is a community edited knowledge base and a support forum of the Fedora community [1]. It is a community maintained website powered by Askbot which is a Python and DJango based web application where the Fedora users get themselves registered to ask and answer questions related to Fedora matters. AskFedora website is ranked 8,857th in the world (amongst the 30 million domains), which means it gets a relatively lesser number of visitors when compared to other Q&A websites like StackOverFlow and AskUbuntu. This site is relatively popular among users in India and gets 32.9% of its traffic from India. This site is estimated to be worth $59,267 and this has a good pagerank(5/10) [2].

AskUbuntu is another parallel website which provides community support for Ubuntu Users. It is a question and answer website for Ubuntu users and developers. It is built and run by the Ubuntu community as part of the Stack Exchange network of Q&A sites [3]. With the help of the community, they are working to build a library of detailed answers to every question about Ubuntu. AskUbuntu is ranked 2,070th in the world (amongst the 30 million domains). This site is relatively popular among users in the United States. It gets 16.4% of its traffic from United States and this site is estimated to be worth $1,843,244 and has an excellent pagerank(6/10) [4].

By comparing the statistics we can see that AskUbuntu is relatively more popular than AskFedora among the community. One of the main reasons behind that is AskUbuntu has many usability features and provides better user experience for the users than AskFedora. AskUbuntu has a consistent look with the other Ubuntu websites and it is built following the same color scheme as their other websites which are based on GNOME Human Interface Guidelines 2.2.2 [5]. It is cross browser compatible across many different web browsers such as Google Chrome, Firefox, Safari, Opera and Internet Explorer, yet it lacks responsiveness when it scales down to mobile devices with smaller screen width.

By applying favourable usability features which can be seen in websites like AskUbuntu, the UI and UX for AskFedora could be subjected to an overhaul to give it some uniformity with the current Fedora websites. There may also be changes to be done in Askbot itself and have possibility of them being integrated upstream. The Fedora community itself aimed at achieving results similar to what AskUbuntu has achieved [6]. However as AskUbuntu is not based on Askbot, similar theming techniques could not be applied and whatever the theming techniques that could be applied had to adhere to the Fedora user interface guidelines and colors.

Responsiveness is a major usability feature which can be seen in most of the new websites today. Responsiveness can be achieved either by using frameworks such as Bootstrap or customizable layout engines such as Susy [7]. The advantage of using customizable layout engines than using fixed column grid layouts provided by frameworks is that the number of columns, the gutter size, the padding etc. can be customized according to the way we want it to be, rather than letting ourselves constrained to the given grid layout. Also for styling, it is advantageous to use CSS pre-processors like Sass instead of traditional CSS because they support maintainability across stylesheets by providing features such as declaring variables, mixins and nesting elements [8].

Considering the technologies which can be used to achieve targets related to user experience, the Fedora community believed with some UX related research the website could be subjected to an overhaul for achieving the aim of providing better user experience for its users.

III. SYSTEM MODELS

A. *System Requirements*

Requirements of the system included mainly requirements related to GUI which were namely the need of a better web layout, need of being consistent with other Fedora websites, need of including dynamic and interactive features in the website and need of making the website cross browser compatible across different web browsers and responsive when it scales down to mobile devices. When it came to usability, the system required an uncluttered interface where the users can find their way through the website without any difficulty and an easy to use, user friendly website where the users require minimum training to completely understand how the website works.

The non-functional requirements of the website included availability, where the users need to access the website throughout the day without facing interruptions, minimum Mean Time Between Failures (MTBF) and minimum Mean Time To Repair (MTTR). And when it came to dynamic and interactive elements they needed to respond quickly within seconds after a user performed an action. The bugs and the defects rate needed to be maintained at a minimum level. The system also needed to be maintainable in order to adapt to any changes at the request of users.

Fig. 1. shows a graphical view of the actors in the system, their use cases and the interactions between them. It shows the main features which are required by the users of AskFedora. It has been identified three main user profiles among users who access the website. Firstly the seekers who are having specific questions to ask and will land directly on the Main page or on the Q/A page of AskFedora, secondly the Contributors who want to answer user questions and are mostly using “Profile” and “Tags” pages in order to find areas which they want to answer, and thirdly the Surfers who land directly on the “Questions” page as a result of querying via a search engine like Google and who may not want to sign up or login. The different types of users are having different use cases as depicted in the Fig. 1. and in the redesign process the web pages which each of these users use, needed to be classified and the feedback received by these different type of users needed to be incorporated when designing the pages.

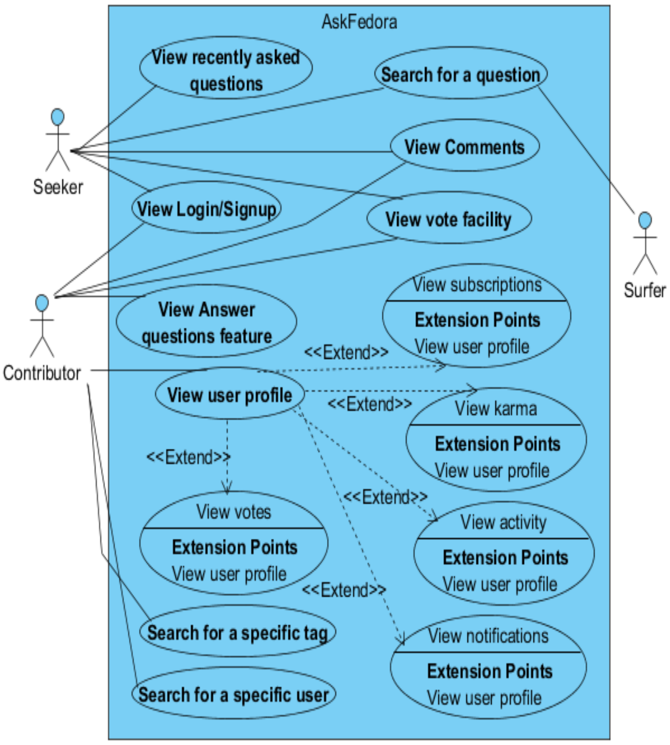


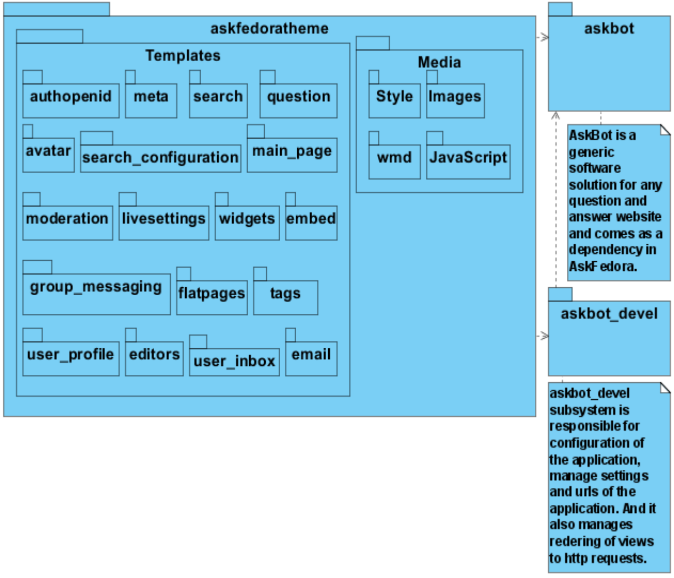
Fig. 1. Use case diagram of the system

B. *System Design*

1) *Main architecture used in the system*

The system consists of three main subsystems. First one is '*Askbot*' which comes as a dependency under the AskFedora instance and is a separate package that offers default functionality of the website. Secondly '*Askbot_devel*' which is a package consisting of python classes like settings.py, url.py, manage.py etc. which handles the control of the web application specifying what pages should be rendered for specific url requests. Finally '*askfedoratheme*'is a package, which all the AskFedora templates and stylesheets are housed. The template package is divided into several other packages as depicted in Fig. 2. This is to generalize the rendering of the templates and to make the system more reusable and easy to maintain.

Fig. 2. shows the sub systems and packages associated with the redesign of AskFedora. It also shows the dependencies between the packages that make up the model.



AskBot is a generic software solution for any question and answer website and comes as a dependency in AskF edora.

askbot_devel subsystem is responsible for configuration of the application, manage settings and urls of the application. And it also manages redering of views to http requests.

Fig. 2. Package diagram for AskFedora Redesign

2) *Layered Architectural View of the system*

The system has a multi layered architecture where the system is divided into three layers. These are the physical layer, the logical layer and the data access layer. The bottom data access layer communicates with the underlying SQLite database and obtains data to be displayed in the web browser, bypassing them all the way up to the physical layer through interactions between the model and view classes of the logical layer. The templates which are rendered by the view classes ultimately contain information fetched from the underlying database and are displayed for the user. This is the system architecture of AskFedora and mainly the redesign process revolved around overriding the templates and styles components of the system. The default database used by the third party Askbot software is used as the underlying database.

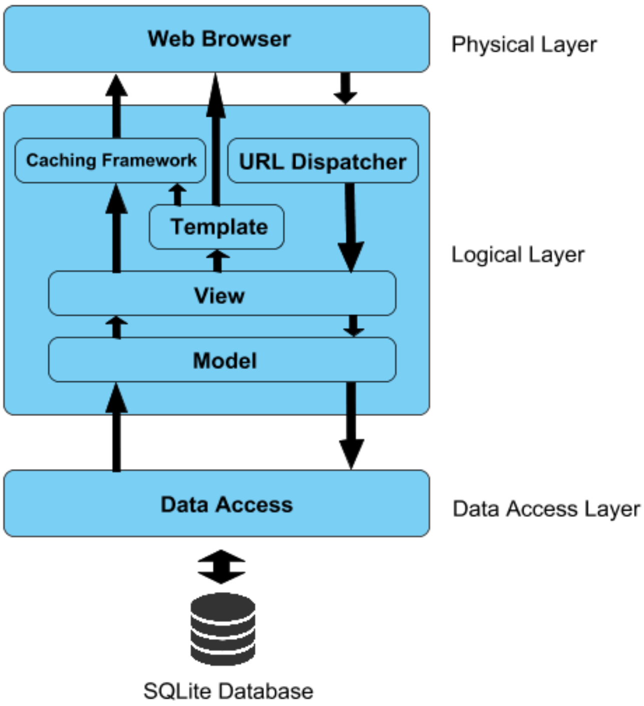


Fig. 3. Layered architectural viw of the system

IV. SYSTEM IMPLEMENTATION

A. *Implementation procedure*

The redesigned version of the website consists of customized templates and styles as well as dynamic and interactive components. For the styles, the system uses Sass, which is one of the CSS preprocessors. Mostly styles are coded using traditional CSS but when the stylesheets become large it is difficult to maintain them further. CSS preprocessing can be used here which helps to have more maintainable and extendable code. Sass has many features similar to those offered in languages like Java and C. Some of such features which are incorporated in the system are variables, nesting, mixins and inheritance. These features have made the system more maintainable and easier to debug.

The system makes use of various mixins provided by Compass version 1.0.1 which is an open source CSS authoring framework. Compass uses Sass, and comes with a lot of Sass mixins and functions that are organized into modules all of which are thoroughly documented. Compass provides a framework for working with Sass stylesheets. The structure of a Compass project includes a Sass file directory, CSS stylesheets directory, a Sass cache directory and a config.rb file. All the configuration details are included in the config.rb file while the rest of the code only needs to be included as Sass files inside the Sass directory. Those Sass files are then automatically compiled to CSS.

For building up a customizable web layout, Susy, a Compass responsive grid plugin is used. It is utilized to produce responsive web pages following a bottom-up approach (Start developing the interface for mobile devices first and moving into larger screens). It has many advantages over the commonly used Bootstrap grid system due to the fact that Bootstrap only allows a 12 column grid system where the number of columns, the column width, the gutter size and the padding are fixed, whereas in Susy all such properties can be customized just by specifying them in the configuration file.

The entire system is powered by Askbot, which is a Python and DJango based web application. The default styles and templates in Askbot are customized and a completely new skin is developed on top of it.

The Rational Unified Process Software development methodology was used when developing the system. The process consisted phases of Inception, Elaboration, Construction and Transition. Major project activities related to UX analysis and mockups development, coding, integration with Askbot and testing were carried out during these phases. For mockups development, Inkscape graphics editor has been utilized. The UX research was conducted based on these mockups, by showing them to the Fedora community and getting them reviewed and accepted. The project had two major milestones. Milestone one was integration of the completed designs of Main and the Q/A pages with Askbot. This was met at the end of first development iteration. Milestone two was integration of the rest of the pages with Askbot. And that was met at the end of second development iteration.

Throughout the project timeline, weekly updates of the project were discussed in Fedora weekly infrastructure meetings where the project progress was reviewed by the Fedora developers community. The feedback received from them were continuously integrated into the design and they were documented in a weekly basis.

B. Materials

The backend of the system is powered by Askbot [9] which is a Python and DJango based generic web application that is used to build any question and answer forum. It follows the MVC architecture. The tool provides features such as efficient question and answer knowledge management, focused on-topic discussions, showing the best answers first, tags and categorization and features related to follow-up and re-posting. It is also used in popular websites such as Investment Q&A, SHAZAM Community and askSAGE [9].

The system is built on top of the existing Askbot database which is an SQLite database. This consists of several user access levels such as admin, moderator, regular, suspended and blocked [10]. In order to backup the database if it is not blank, it requires to type in the commands of ‘python manage.py syncdb’ and ‘python manage.py migrate’. Then it syncs the existing data with the database. In order to add a new table for the existing database nothing related to the core of Askbot needs to be changed but it can be added by including the format of the new table in models/_init_.py file and recompiling. Some of the existing question and answer data which is in the current website of AskFedora was used for demonstration purposes.

The new color palette used for the website is the default Fedora color palette [11]. That is to make the website consistent with the existing Fedora websites. The logo used is the Fedora logo [12] and the website follows all the usage guidelines of logos and colors indicated in the Fedora wiki [13].

C. The Algorithm

Displaying contributor information is one of the most important parts of the source code. When originally posting a question or answer on the website or else if the question or answer is updated, the information of the user who did the original posting or the update needs to be properly displayed in the interface. In order to achieve it, the code given by Fig. 4, is included in the macros.html file. As the code relevant to this functionality is required from multiple places it is included as a function.

```
1. post_contributor_info(revision, contributor_type)
2. if contributor_type = original_author
3.  if revision → post_type = question
4.      action_label = WORDS.ASKED
5.  else if revision → post_type = answer
6.      action_label = WORDS.ANSWERED
7.  else
8.      action_label = posted
9.  else
10.action_label = updated
11. if revision → post_type = question | answer
12.original_author = revision → author
13. else
14.original_author → none
15. inside <p> tag {
16.if revision → post_type = question
17.  href = urlquestion_revisions revision → id
18.else
19.       href = urlanswer_revisions revision → id
20.    }
21. show_user_info =(contributor_type =
original_author | last_updater&original_author !=
revision → author)
22. if show_user_info
23. post_contributor_avatar_and_credential(revision)
```

Fig. 4. Psedocode for updateing questions and answers

According to the pseudocode, if it is a question which is originally posted by a user, then the wording displayed shows ‘asked’. If the post type is an answer, the wording displayed shows ‘answered’. Else if it is a comment which does not belong to either a question or an answer, it displays just the wording ‘posted’. If it is a post which is updated, then the wording relevant to update of posts is displayed on the interface (line 1 - 14).

The latter part of the pseudo code from line 15 – 20, indicates how the URLs relevant to questions and answers are displayed. And lastly if the show_user_info boolean attribute is true, then the method ‘post_contributor_avatar_and_credential(revision)’ is triggered which displays the relevant user information along with the user profile and other user credentials (line 21 - 23).

D. Main User Interfaces

The main page of the website displays the most recent questions asked by the Fedora users. The new design of the website displays the questions in the form of question cards so that it will have an uncluttered user friendly web interface. when a user clicks on a question card, he/ she is directed to the Q/A page that show the question and the answers in detail and the user can answer questions and make comments. The user profile page contains profile information of the registered users in the contexts of Network, Karma, Followed Questions, Activity, Notifications, Votes and Subscriptions.

Tags, People, Badges are the pages with medium interactions. The Tags page has all the tags which the AskFedora users have included in when asking questions. The new system includes features for the users to search a particular tag by typing the tag name in a text field whereas in the old system the user did not have such a search mechanism. The People page has all the profiles of AskFedora contributors and of all the people who have registered themselves on the website. In the new system, the People page provides a search mechanism to search a person by his or her username. The Badges page has details of the rewards which are offered to active AskFedora users.

About, FAQ, Help, Privacy Policy, Feedback are the pages with low interactions. Links to these pages are included in the footer of the website and are rarely accessed by the users. These contain static content and hence will follow the same layout as of the other important pages discussed above.

E. *Software Interfaces*

The system makes use of various mixins provided by Compass version 1.0.1, an open source CSS authoring framework. Compass uses Sass, and comes with a lot of Sass mixins and functions that are organized into modules, all of which are thoroughly documented.

Susy, a Compass responsive grid plugin is used to produce responsive web pages by following a bottom-up approach (Doing the interfacing of the mobile devices first and moving into larger screens) in designing the web interfaces. Askbot, which is a Python and DJango based web application, is used in AskFedora and the styles and the templates used in it are customized and a completely new skin is developed.

F. *Hardware Interfaces*

The minimum requirements the hardware interfaces should have are described in Table 1.

TABLE II. HARDWARE INTERFACE MINIMUM REQUIREMENTS

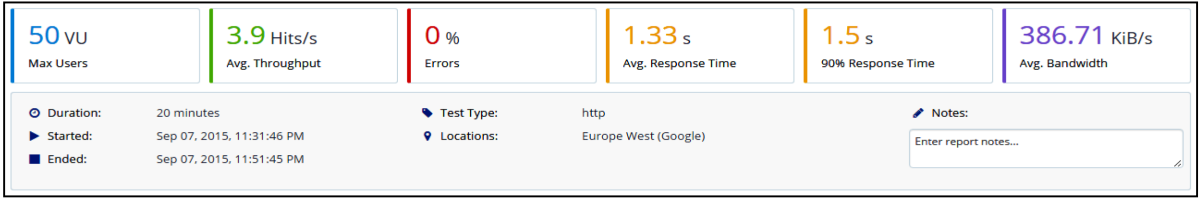
Client Side			
Web Browser	***Processor***	***RAM***	***Disk Space***
Internet Explorer-8	Intel Pentium III or AMD-800 MHz	128 MB	100 MB

G. *Communication Interfaces*

The client needs to communicate with the server in which the website is deployed by means of HTTP/HTTPS protocols and access the website through any kind of web browser.

V. SYSTEM TESTING AND ANALYSIS

The overall system was tested under categories of functionality, UI, performance and security. For functionality testing a methodical testing strategy was utilized where the system was divided into small units. On each unit a specific pre-defined set of tests were performed to verify that the system fulfills all the major user requirements. Selenium IDE [14], Selenium WebDriver [15] and Junit [16] are the tools used to run functional testing.



50 VU	3.9 Hits/s	0 %	1.33 s	1.5 s	386.71 KiB/s
Max Users	Avg. Throughput	Errors	Avg. Response Time	90% Response Time	Avg. Bandwidth

Duration:	20 minutes	Test Type:	http	Notes:	Enter report notes...
Started:	Sep 07, 2015, 11:31:46 PM	Locations:	Europe West (Google)		
Ended:	Sep 07, 2015, 11:51:45 PM				

Fig. 5. Performance test results by Blazemeter

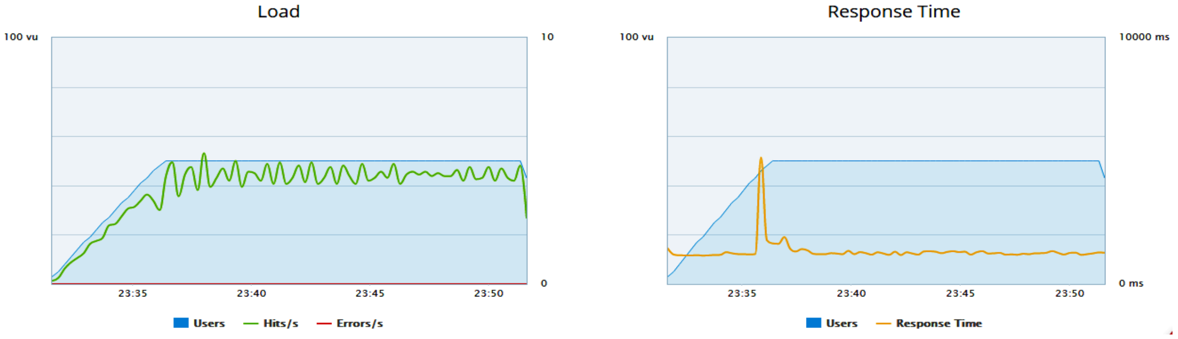


Fig. 6. Graphical illustration of load and response time of the website

For UI testing a regressive-averse testing strategy was applied where all the tests related to user interface testing were automated using BrowserStack [17]. Thus, when a change is occurred the set of automated tests were run repeatedly to ensure that nothing related to the UI is broken and responsiveness and cross browser compatibility of the website is properly maintained.

Performance of the website was tested simulating from 10 to over 1 million concurrent users with ease by running the website in a virtual sandbox environment. Blazemeter [18] tool was used for this purpose which can be used to launch tests from multiple locations in the USA, Europe and Asia Pacific for a more accurate user representation. Fig. 5, and Fig. 6, show the throughout, average response time, average bandwidth etc. It can be seen that the website has reasonable performance when keep on increasing the number of users.

In terms of security, the web application was subjected to penetration testing to find the vulnerabilities that an attacker could exploit. The penetration tests were automated. It gathers the information about the target before the test, identifies possible entry points and attempts to break in to the website. The findings of the test are reported so that it can determine security weaknesses present in the website and can be used to further enhance the security of the system. For this Xelenium security testing tools [19] were utilized which runs on top of Selenium Web Driver [15].

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a redesign approach for the AskFedora community knowledgebase. The design and development of this project are mainly based on user experience analysis, mock-ups development, coding and integration with Askbot. The development has improved the responsiveness of the system that facilitated with scaling down to mobile devices with smaller screen width, consistency with the other Fedora websites and cross browser compatibility. This would provide an enhanced user experience for the Fedora users and would in turn increase the popularity of the website.

This work can be extended to improve the GUI of the website in terms of removing some of the pop-up menus and including them as option buttons, which will directly catch the user’s eye without navigating through a menu. Also searching facilities can be improved using JavaScript base search features. The usability of the website can be improved by introducing a set of chat rooms to discuss user issues with the online contributors. Further, the financial aspects of the website can be improved by allowing advertising opportunities for career centers so that the users could find some opportunities related to their areas of interest and expertise.

ACKNOWLEDGEMENT

This research project was partially supported by The Fedora Project open source community; The Fedora Design team and The Fedora Infrastructure team.

REFERENCES

[1] Ask.fedoraproject.org, 'About Ask Fedora - Ask Fedora: Community Knowledge Base and Support Forum', 2015. [Online]. Available: https://ask.fedoraproject.org/en/about. [Accessed: 18- Sep- 2015].

[2] Webstatsdomain, 'Ask.fedoraproject.org', 2015. [Online]. Available: http://ask.fedoraproject.org.webstatsdomain.org. [Accessed: 18- Sep- 2015].

[3] Askubuntu.com, 'Tour', 2015. [Online]. Available: http://askubuntu.com/tour. [Accessed: 18- Sep- 2015].

[4] [4] Webstatsdomain, 'Ubuntuforums.org', 2015. [Online]. Available: http://ubuntuforums.org.webstatsdomain.org. [Accessed: 18- Sep- 2015].

[5] Developer.gnome.org, 'GNOME Human Interface Guidelines 2.2.2', 2015. [Online]. Available: https://developer.gnome.org/hig-book/3.4. [Accessed: 18- Sep- 2015].

[6] Fedoraproject.org, 'Summer coding ideas for 2015 - FedoraProject', 2015. [Online]. Available: https://fedoraproject.org/wiki/Summer_coding_ideas_for_2015#AskFedora_UX.2FUI_.26_Functionality_Overhaul. [Accessed: 18- Sep- 2015].

[7] Susydocs.oddbird.net, 'Susy 2.2.6 — Susy 2.2.6 documentation', 2015. [Online]. Available: http://susydocs.oddbird.net/en/latest. [Accessed: 18- Sep - 2015].

[8] Sass-lang.com, 'Sass: Sass Basics', 2015. [Online]. Available: http://sass-lang.com/guide. [Accessed: 18- Sep- 2015].

[9] Askbot.com, 'Create your own Q&A powered by Askbot in a minute.', 2015. [Online]. Available: https://askbot.com. [Accessed: 18- Sep- 2015].

[10] Pypi.python.org, 'askbot0.7.53 : Python Package Index', 2015. [Online]. Available: https://pypi.python.org/pypi/askbot. [Accessed: 18- Sep- 2015].

[11] Fedoraproject.org, 'Logo/UsageGuidelines - FedoraProject', 2015. [Online]. Available: https://fedoraproject.org/wiki/Logo/UsageGuidelines#Colors. [Accessed: 18- Sep- 2015].

[12] Fedoraproject.org, 'Logo/UsageGuidelines - FedoraProject', 2015. [Online]. Available: https://fedoraproject.org/wiki/Logo/UsageGuidelines#Permitted_logo_variants. [Accessed: 18- Sep- 2015].

[13] Fedoraproject.org, 'Logo/UsageGuidelines - FedoraProject', 2015. [Online]. Available: https://fedoraproject.org/wiki/Logo/UsageGuidelines. [Accessed: 18- Sep- 2015].

[14] Seleniumhq.org, 'Selenium IDE Plugins', 2015. [Online]. Available: http://www.seleniumhq.org/projects/ide/. [Accessed: 18- Sep- 2015].

[15] Seleniumhq.org, 'Selenium WebDriver', 2015. [Online]. Available: http://www.seleniumhq.org/projects/webdriver/. [Accessed: 18- Sep- 2015].

[16] Junit.org, 'JUnit - About', 2015. [Online]. Available: http://junit.org/. [Accessed: 18- Sep- 2015].

[17] Browserstack.com, 'Cross Browser Testing Tool. 300+ Browsers, Mobile, Real IE.', 2015. [Online]. Available: https://www.browserstack.com/. [Accessed: 18- Sep- 2015].

[18] Continuous Performance Testing for DevOps | BlazeMeter, 'Continuous Performance Testing for DevOps', 2015. [Online]. Available: http://blazemeter.com/. [Accessed: 18- Sep- 2015].

[19] Owasp.org, 'OWASP Xelenium Project - OWASP', 2015. [Online]. Available: https://www.owasp.org/index.php/OWASP_Xelenium_Project. [Accessed: 18- Sep- 2015].