Reporting common street problem to relevant local authority:

“FixEmUp” – a software solution for Tornio city

Zahid Choudhury

Bachelor’s Thesis of the Degree Program in Business Information Technology
Bachelor of Business Administration

TORNIO 2012
ABSTRACT


The objective of the thesis is to describe the “FixEmUp” software and elaborate the working mechanism as well as the architecture of the system. To accomplish the objective of the thesis, constructive research method has been used to gather information and to design “FixEmUp”. Literature review and detailed documentation analysis were supplementary research techniques to get more familiar with the “FixEmUp” system.

The choice of using constructive methodology is based on the fact that the thesis can mitigate real life problem. The thesis also provides practical answers regarding the software’s different aspects. The thesis also includes literature analysis of every using materials regarding software.

The main goal of this thesis is to enable opportunity for further development as well as research about “FixEmUp” web application. As the software technically opens a new way of advance communication between the citizens and authority, the thesis only describes the things theoretically.

The analysis of documentation and literature on “FixEmUp” architecture as well as the design and demo implementation is summarized in the result in the discussion and conclusion chapter with further research suggestions.

Keywords: FixEmUp, open source, PHP, MySQL, Google Map, Database, HTML, complain
# ABBREVIATIONS LIST

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>PHP</td>
<td>Hypertext Preprocessor</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>XSS</td>
<td>Cross-Site Scripting</td>
</tr>
<tr>
<td>XSL</td>
<td>eXtensible Style Language</td>
</tr>
<tr>
<td>ER Diagram</td>
<td>Entry Relationship Diagram</td>
</tr>
<tr>
<td>Software V&amp;V</td>
<td>Software Validation and Verification</td>
</tr>
</tbody>
</table>
ABSTRACT

ABBREVIATIONS LIST

FIGURES

CONTENTS

1 INTRODUCTION ........................................................................................................... 7
  1.1 Background and orientation ................................................................................. 7
  1.2 Objectives and methods ...................................................................................... 8
  1.3 Structure of the thesis ......................................................................................... 10

2 RESEARCH QUESTIONS AND METHODOLOGY ............................................... 12
  2.1 Research questions ............................................................................................ 12
  2.2 Methodology ....................................................................................................... 14

3 THE CONCEPT OF SOFTWARE FOR REPORTING SOCIAL PROBLEM “FixEmUp” ............................................................................................................. 16

4 REQUIREMENTS ANALYSIS FOR “FixEmUp” ..................................................... 23
  4.1 Functional requirements ...................................................................................... 23
  4.2 Non-functional requirements ............................................................................... 26

5 “FixEmUp” TOOLS .................................................................................................. 28
  5.1 HTML .................................................................................................................. 28
  5.2 Style Sheet Technologies .................................................................................... 29
  5.3 PHP ..................................................................................................................... 30
  5.4 MySQL ............................................................................................................... 33
  5.5 Google map ......................................................................................................... 37
  5.6 PostGIS ............................................................................................................... 38
  5.7 DJango ............................................................................................................... 39

6 ARCHITECTURAL DESIGN ................................................................................. 42
  6.1 Analysis of components ..................................................................................... 43
  6.2 Components modeling and design ...................................................................... 44
DESIGN VALIDATION .............................................................. 47

DEMO IMPLEMENTATION ...................................................... 51
  8.1 Apache server setup ...................................................... 51
  8.2 Setting up database ..................................................... 53
  8.3 Local Server Testing ..................................................... 55
  8.4 Setting up ftp software .................................................. 55
  8.5 Setting up domain name ............................................... 56
  8.6 Testing at server side ................................................... 57

DISCUSSION AND CONCLUSIONS ........................................... 58
  9.1 Discussion of the results ............................................... 58
  9.2 Limitations of the system ............................................. 60

REFERENCES ........................................................................... 61

APPENDIX .............................................................................. 69
FIGURES

Figure 1: “FixMyStreet” UK ................................................................. 16
Figure 2: “FixMyStreet” unsolved problem status .................................. 17
Figure 3: Hotline number .................................................................... 18
Figure 4: Karta Russia interface ............................................................ 19
Figure 5: FixMyStreet Canada ............................................................... 20
Figure 6: mas-city.com interface ............................................................ 21
Figure 7: mas-city.com problem solving report ...................................... 21
Figure 8: Report a problem (“FixEmUp” application) .............................. 24
Figure 9: Reported problem notification (“FixEmUp” application) ............ 25
Figure 10: User Registration ................................................................. 25
Figure 11: SQL query example .............................................................. 34
Figure 12: ER Diagram (FixEmUp application) ...................................... 35
Figure 13: Google map of Tornio city .................................................... 37
Figure 14: Django architecture ............................................................. 40
Figure 15: Django framework interface ................................................ 41
Figure 16: Software Design procedure identification by Basem(2010) ....... 48
Figure 17: XAMPP control panel .......................................................... 52
Figure 18: XAMPP database setup ....................................................... 53
Figure 19: FixEmUp at local-server ....................................................... 54
Figure 20: coreFTP interface ............................................................... 55
1 INTRODUCTION

Reporting street problems is no longer an easy process for the citizens. They have to undergo a long procedure to report problems such as street damages, street cleaning shortcomings and light post damages. To facilitate this complain procedure, a new web application has been introduced under the name of “FixEmUp”.

1.1 Background and orientation

“FixEmUp” is an online based application which has introduced a new technology of sharing problems between civil services authorities and the public for resolving problems such as road maintenance, abandoned vehicles, dog fouling, flyposting or graffiti, fly tipping or litter, street cleaning, broken glass in a cycle lane, unlit lamp post, potholes and such kind of daily life disturbances. By this application people can easily complain to civil service authorities and the smart database system keeps tracing all the records of complains and problem solving statements. The main focus of this system is to maintain a satisfactory relationship between citizens and governance and accelerate the process of civil development. It creates a bridge between citizens and governance, promotes exchange and it all contributes to improving the city.

“FixEmUp” is user friendly and convenient. It introduces citizens to the benefits of using community tools and communities to new Internet uses. To report problems, users have to enter the name of the street of the problematic zone and Geo-tag the place with the Google map. For more convenience, users also can identify the streets with its postal address. To make complains more reliable and specific, users can attach pictures of the present conditions of that area with the complaint document. The intelligent automated system will build a form consisting of all the details entered by the user which can be monitored by the administrator of this application. The best part of this
scenario is civil service authorities can monitor the situation and take swift decisions to recover the problem.

1.2 Objectives and methods

Basically “FixEmUp” was developed as an advanced database management system. People can complain about the small or big problems which get recorded at the database and steps taken by authorities can also tracked by the application. As a result, the total situation of the complaint and the steps taken regarding the complaints, are stored on the application database. It is important for the final review report to show how active the authorities are regarding the expectations by the citizens.

There are some interesting features in “FixEmUp” that make the application efficient. People can geo tag their expected location at the Google map with the revolutionary red balloon. People also can use pictures taken by mobile or digital camera of the reporting area that the civil service authorities can see the original view. “FixEmUp” has the features such as RSS feeds and email alerts. The alerts generated for all problems within a particular ward or council, or within a certain distance of a particular location.

“FixEmUp” is open source web application. The application will be available publicly with its source code and basic architecture. This thesis will help to understand the application and its usability more significantly. “FixEmUp” is complete web application, although update and modification of this application is a basic concern as user demand.

As an example, the scenario of Tornio city can be counted, as it is not so different from the other cities of the world. For any social problems, such as road damage, people first have to complain to the local municipality office. After receiving the complaint, the civil service authority inspects the problematic area and takes the initial steps needed. It takes a long time and long process to solve the problems. However, by using
“FixEmUp” the problem can be solved swiftly. The application will help the Tornio city and its citizens to mitigate the communication gap for reporting problems between the city council and citizens. Moreover, the code and documentation also could be used later on for business purpose in different municipalities or organizations.

As “FixEmUp” has been made as a next generation system, users can get additional benefit from its advanced technology. The intelligent system of map tracking can be beneficial for the civil service authorities to authorize the right place. The advanced database builds a quick summary of the total problem reported and steps taken regarding the problem reports in a moment. Moreover, the comment giving technology is important to discuss any topic which creates increased accountability. The thesis describes the questions regarding the application and its back-end architecture. The thesis also describes the functionality of “FixEmUp” work process and its usability. The thesis also focuses on compatibility of “FixEmUp” and other software programs for example, Internet Explorer, Firefox, Chrome. In brief, “FixEmUp” is an application functioning as web software to enhance the reporting services to government and how it maintains the application challenges.

Opinions regarding the idea of the system and its use are collected with the help of questionnaires. The questionnaires are attached on appendix 1 at the end of the thesis. This is very important because if the users of the system are not willing to implement the new system they will only become annoyed of it and they might even start fighting against the upgrade.

The data collection will be maintained through series of interviews consists of 100 citizens of Tornio city. The questioner with the citizens of Tornio city details are attached at appendix section. Collected data are analyzed for the purpose of application development. The collected data received from the questioners are gathered and used in the “FixEmUp” tutorial for Tornio City Hall staff. After gathering enough data the first version of the software will be developed, which will be available for the city hall staff for feedback. The demo version of that application showed to 100 citizens and collect
feedback from them. The target citizens are It-professionals, teachers, students and municipality. Based on their feedback, necessary modifications and amendments have been done to make the application more users friendly and interactive. Then a release candidate of the application has made which is available for everyone. If any bug detected that’s will be detected as a high priority task and fixed immediately.

1.3 Structure of the thesis

Chapter two discusses the research project scopes and related questions and methodology. It describes the project purpose in every aspect of user and researcher. The third chapter describes the concept of the application. It provides a detail comparison between other social problem solver applications and “FixEmUp”. This chapter also includes the history of social problem solver applications that how it comes and expends to make citizens life easier.

In chapter four requirement analyses has been described. It provides a brief idea of “FixEmUp” applications requirement for social life in every prospect. This chapter also includes analysis form technical view.

Chapter five is the most important chapter as it includes the details of the tools that are being used. In this chapter all used tools are described in detail as a normal user can get an idea about the tools and why they were used. The chapter also describes the advantages of using the tools that are being used, to grow a future research scope for the user.

After describe about tools, chapter six includes the architectural design review for the software. This chapter includes the analysis of the components of the application as well as components modeling and design. Chapter six is important because it shows all the details of architectural components.
Chapter seven is an extension of chapter six. This chapter describes the design validation of “FixEmUp” in all aspects. This chapter specifies the design model and describes they were used. This chapter also creates a clear view of previous chapter of architectural design.

Chapter eight is actually a write up of “FixEmUp” deployment. It describes all the process of the application’s setup and testing. This chapter also provides the information of “FixEmUp” applications after deployment performance.

Chapter nine is the discussion and conclusion in theoretical view of the application. This chapter provides a theoretical summary of total application as well as review from the developers. This chapter also includes the result of “FixEmUp” applications development and deployment.
2 RESEARCH QUESTIONS AND METHODOLOGY

The thesis is about “FixEmUp” Development and Deployment. The general aim of this thesis work is to study “FixEmUp”. The specific objectives of this thesis include developing a demo implementation of the application called “FixEmUp” Tornio city. In addition it is identified and analyzed the available solutions for developing and deploying the software.

2.1 Research questions

In order to achieve the objectives of the thesis, three research questions have been developed.

Q1. What are the functional and non-functional requirements of “FixEmUp” web application?

“FixEmUp” introduces a substantial resource of different web applications and tools on its back-end functionality. Based on user requirements and the software functionalities detail analysis, the functional and non-functional requirement of “FixEmUp” has described on “Requirement analysis for FixEmUp” part. This question also provides evidence about how “FixEmUp” integrates the functional and non-functional requirements.

Q2. What kind of techniques/technologies could be used for “FixEmUp” software application architecture, design and development?
“FixEmUp” uses object-oriented architectural style. To ensure more complex form at the back-end architecture of “FixEmUp” development, Evolutionary Development Model, short form EVO, has been used at its back-end modeling. It enables a substantial prospect of complex back-end system safety and establishes reliability for the next updates of software which has been described briefly on architectural design chapter. “FixEmUp” development has been inspired by Agile development. As the user criticism and feedbacks, which are attached with the thesis, have placed a vital role for development of “FixEmUp”, it is considered as priority basis requirement for its development. The thesis describes these criteria so briefly at the development chapter.

“FixEmUp” target clients are the municipality of Tornio city. The machine set up and requirements also fulfill the criteria of the municipality’s requirements. The thesis describes them briefly.

Q3. How will “FixEmUp” mitigate the reporting problem between the city council and the citizens?

The level of civil satisfaction is getting increasingly low as it takes a long period of time for any reconstruction or reparation process of roads and other governmental properties by the civil service authorities. The “FixEmUp” application brings a quick solution for these challenges. The thesis shows how “FixEmUp” speeds up the process and reduces the distance between the civil service authorities and citizens.

Although the demand for and the prospect of the software is more or less progressive to the citizens of Tornio municipality, the statistics of people opinion attached to the thesis shows a good picture of the responses. It proves how much people such as the system and thinks that they will be benefited. The thesis also includes the citizen’s suggestions and expectations which can enhance the future plan more efficiently.
“FixEmUp” is open source web application and free for all. Therefore, a continuous process of upgrading and opinion collection from its users is also available. The thesis shows that how the advanced features recommended by its users can enrich the software. Also “FixEmUp” demo review is attached to the thesis to clarify the user satisfaction with the application.

2.2 Methodology

This thesis has been conducted with constructive research method. The objective of this work is to design and implement a demo version of “FixEmUp”. Constructive research provides a solution to a real world problem. In my work I focus on the theoretical aspects of how to develop the software. In addition, I solve the problem of reporting problems between the municipality of Tornio city and its citizens. This research methodology provides a detailed framework to find a real world problem and will help to solve the problem.

A research has to be structured, logical, empirical and replicable. Moreover, a scientific research must be able to clarify the objectives of the research and use the common concepts. (Kothari, 2004) According to Kothari (2004, 107-109) a good research has to comply with the following criteria:

- The research procedure used should be described clearly to permit other researchers to continue the research work for further development.
- The procedural design of the research should be planned carefully so that the results of the research reflect the objectives.
- The researcher should report the frankness, flaws and limitations of the procedural design and predict their effects in the result.
- The validity and reliability of the data should be justified and checked. (Kothari, 2004, 107-109)
In my thesis, I have complied with the given criteria to justify the procedural design of the research work. Therefore, I have clarified the research method used in my work with several logical argumentations.

The research method used for this thesis is constructive research method. The decision to choose the constructive research method was logical because the objective of this thesis was to construct the design of “FixEmUp” application and demonstrate the prototype which simulates the implementation, maintenance and testing of the software under a testing environment. In reference to Järvinen’s (2001, 88) suggestion, “It is typical for constructive method to build a new innovation and this process is based on existing (research) knowledge and/or new technical, organizational, etc. advancement” which guided the research work.

Furthermore, my final goal of the thesis is to design the system and implement the design in developing the prototype to demonstrate under simulation environment. Therefore, reference to Järvinen’s (2001, 88) suggestion on constructive research method which is, “The constructive research is applied research, but instead of the final product it is possible to accept the prototype or even a plan” relied on in this work as well.
3 THE CONCEPT OF SOFTWARE FOR REPORTING SOCIAL PROBLEM “FixEmUp”

“FixEmUp” is an application made for reporting social problems such as, road damages, lamp post damages, and water pipe damages. The concept of this kind of software first comes from the website named "mySociety". It is an English website which worked for solving social problems, for example: users can report potholes, broken street lights.

The website first named was "Neighborhood Fix-It". But in June 2007, the name was changed as “FixMyStreet” to make it user friendlier. In 2008, an iPhone application was developed, named “FixMyStreet”. From that application, problems can be reported the same way as computer from iPhones. Since then volunteers have tried to release software for Nokia, Android as well as another application for iPhone.

Figure 1: “FixMyStreet” UK

The “FixMyStreet” application has done a revolutionary change for the citizens of UK. Since its launch, more than 25,000 problems were reported in UK. One of the reasons of this substantial response is it allows citizens to directly identify problems in their neighborhoods and cities are able to address problems more quickly. According to
MySociety’s survey approximately 45% of problems submitted through “FixMyStreet” are addressed. ([http://www.fixmystreet.com/faq](http://www.fixmystreet.com/faq))

Although there still some problems with the application. According to Stephen (2007, 72-80), he discussed on his paper that, many problems reported to the site remain "unfixed" with status "unknown".

![FixMyStreet](image)

Figure 2: “FixMyStreet” unsolved problem status

“Local government officers voice a number of concerns: the site duplicates their own websites; they cannot report fixes directly to ensure the information is not up-to-date; nor can they manage user expectations regarding service performance. The site also has no ‘sense of virtual community’ (SOVC), with individuals posting in isolation without the support of fellow citizens to amplify their voices or to contribute to problem resolution.” (Stephen 2007, 72-80).

In Russia a site named “carta vibor” accessible at [http://karta.vibor33.ru/](http://karta.vibor33.ru/) does the same job for Russian citizens. The application is user friendly and it has some interesting
features integrated. One of the features is the Hotline numbers. The application have two hotline numbers on where citizens can complain if any urgent situation.

Figure 3: Hotline number

Another noticeable feature of this application is the language. By default Russian language has been used for this site. It creates more flexibility to the users to sue the application. Another unique feature of this site is the visual complain status. With every complain there comes a status bar where red block indicates send request, yellow block represents answered and the green block represent problem solved.

Karta application is special for its suggestion feature. With every problem there comes a suggestion for possible solutions of the problems. It is special because only karta among all applications such as fixmystreet, provide this feature. More than that, this feature is really helpful for the civil service authorities to judge for possible solutions. Provided tutorial instructions for using the application make it richer.

Another good feature for the application is the use of two maps at index page. The big map provides the information of problems and in testing. The small map shows the road map. More than that, special tagging system at map makes the application easier to use. The tag is so smart that user can browse the problem only clacking the icon at the map. A detail pop up appears with picture. Overall the application is an ideal of these applications.
In Canada, a same type of site has been worked for Canadian citizens and it is accessible at http://www.fixmystreet.ca/. The site named as FixMyStreet Canada which is maintained by the non-profit organization visibleGovernement.ca. The web site is inspired by the “mySocity” website. This site also ensures the same reporting facilities for the Canadian citizens.
One of the remarkable features of this site is it ensures to keep in touch with city councilors to obtain ward maps for the reporting area. It supplies all the reports to city councilors by email or by using “311” hotline number. Another feature of FixMyStreet Canada is it easily imports in many formats. Such as, the ESRI shape files, TIGER shape files, GML, KML, Postgres or MySQL/SQL and WKT 84 Polygons. The maps also can get in other formats as per request from the users.

In Germany, the name of this type of available software is “Mark-a-Spot” accessible at http://mas-city.com/. “Mark-a-Spot” has used some new technologies at their GUI and technical side. It gives the freedom to its users to choose between Google maps and Open Street Maps. The software is royalty-free and it does create transparency and document active management action. AdHoc’s image use is another good addition of this application.
One of the key features of this application is voting on proposals. Mark-a-Spot is developing now increasingly becoming a platform for online dialogue with a focus on geo-referencing. The practice of Maerker Brandenburg shows that besides the actual "Scandals" and useful ideas and suggestions will be entered which are worth between administration and citizens / internally discussed, evaluated and adapted to be. (Kreis 2010).

Mark-a-Spot also integrate the system with Facebook and tweeter. The tweeter addition makes the system a new user experience. People can tweet their problem and tweet fixed issues directly to a twitter-account.
Another big feature of Mark-a-Spot is the development scope by using the system. Mark-a-Spot creates opportunity to web developers and integrators various opportunities. For Example, web based installer and configuration, cakePHP and jQuery usage, open standards and interfaces, web application in mvc pattern. Flexibly expandable is another key feature of this application. This feature provides details about ticket systems, workflows, attachments. Some of the web browsers don’t have java Script enable for security reason. But the application ensures run with the same look and functionality in all conditions such as, the java script on or off. The application ensures separate administration surface for user and administrator. The application also ensures the right distribution of user administration. The application is more graphically interactive to its users and come with user friendly features.
4 REQUIREMENTS ANALYSIS FOR “FixEmUp”

To check “FixEmUp” web application compatibility on user prospect it is important to focus on software requirement analysis. The analysis can be described into two sections as functional and non-functional requirements. The functional requirements describe the prospect and capabilities of “FixEmUp” whereas the non-functional requirements describe the issues as security and performance.

4.1 Functional requirements

The functional requirements describe the product capabilities and prospect regarding the client requirements. It captures the intended behavior of the system. This behavior can be described as services, tasks or functions that the system is required to perform. The International Institute of Business Analysis (IIBA) defines functional requirements as “the product capabilities, or things that a product must do for its users.”(Ellen 2005, 79-85) The functional requirements analyze the behavior of software that how it meet the user requirements. Such as functional requirement of a stock trading system must have "update and remember stock prices", "accurately parse Boolean queries" and “must process withdrawals and dispense cash to the customer”. The more detail client requirement integrated by the software the more high-level functional requirement it mitigated.

By analyzing “FixEmUp”, it can be said that functions such as “Report a problem”, “View submitted Reports” etc. can be detect as functional requirements. Because the “Report a problem” section is made for users as users can make complaints there. The “View submitted problem” part is for those users who can handle or fix problems such as the civil service authorities. Some advance searching features in “FixEmUp” web
application also maintain the functional requirement such as “search by problem”, “search by city”.

The “Reporting Problem” is one of the basic goals and most important functional requirements of “FixEmUp” web application. In every steps of “FixEmUp” orientation, this criterion has been maintained perfectly. The biggest advantages of the application are the fact that people can find the place by search, and can complain about it with all the details and from anywhere. Such as in figure 8 first the Tornio city has been searched from the map and then user complains the problem with all details. To make the feature more specific and detailed, the users also can add images as a visual proof of the complaint.

![Figure 8: Report a problem (“FixEmUp” application)](image)

Viewing submitted problem is another functional requirement of “FixEmUp”. As Figure 9 shows that all the reported problem by the user is notified at the Google map at “FixEmUp”. By clicking at the balloon (red balloon at figure 1 or Google map) user can see the previous reports / complaints about the place.
Figure 9: Reported problem notification (“FixEmUp” application)

To make this feature more interactive and user friendly a tab named “View the Submitted Report” has been added at the website top menu bar that the users can easily watch the submitted reports. In addition, after every report submission the user can view the result.

However, “User Registration” is another functional requirement of “FixEmUp” application. A user can register and use all the features of “FixEmUp”. They also can give unlimited number complaints.

Figure 10: User Registration
4.2 Non-functional requirements

The non-functional requirement of software is based on the quality or the standard of software. It is a standard that that might automate a sort of task by the system. Such as “System must be secured against Trojan attacks”, “system must run on windows server” etc. The IIBA (International Institute of Business Administration) defines non-functional requirements as “the quality attributes, design and implementation constraints, and external interfaces which a product must have.” (IBID 2010) Mostly non-functional requirements emphasize on the quality control attributes such as availability, maintainability, performance, portability, reliability, robustness, security, scalability, testability, usability and others. Some non-functional requirements may have specific usages of the system. Such as sometimes 24 hours online availability can be a high-level non-functional requirement.

In “FixEmUp” application the non-functional requirements applied so well, as it has a very high level security system. The software mainly focuses on “Cross-site scripting prevention”, “Spam protection”, “Bug detection”, “prevent online fishing” and so on. The 24x7 availability is of course well established in this software which meets the high-level non-functional requirement. The interface of “FixEmUp” looks so easy and user friendly which is also replicated non-functional requirement.

Furthermore, one of the basic non-functional requirements of “FixEmUp” is the use of SQL. The software database is written in SQL language and using advance techniques such as SQL injection, scripting language prevention, has make the software so secure and reliable. As the database of the software is too large and substantial data stored expected, the database use advance SQL searching to use the database so faster.

Todays, cross-site scripting (XSS) is one of the common and effective ways to hack or harm a site or web applications. “The name cross-site scripting is derived from the fact that one web site or person can inject script of their choosing across security boundaries to a different and vulnerable web site. XSS is a type of injection attack, but rather than the attacker directly performing the injection, the attacker must lure the victim to
perform the injection.” (Cannings 2007) It is a type of scripting which enables hackers to inject client-side script into the web pages or applications and bypass access control such as the same origin policy. In “FixEmUp” the cross-site scripting has specially handled with SQL injection system.

In addition, “Bug Detection” is another high level non-functional requirement in “FixEmUp”. As the software is under deep authorization the bug detection level by the developer is very high.

In the back-end database system, HTTP apache server has been used as a non-functional requirement. As the first ever server introduced by Netscape, Apache has its own reputation of a safe and flexible server usage. Apache HTTP server is the world’s most popular web server, which has been used by more than 64% of the sites on internet. HTTP server, along with PHP, is a project of Apache software foundation (www.apache.org). One of the biggest features of apache server is its open source. So the bug detection level is very high and frequent updates make the software so rich and stable. Another advantage of using Apache is a lot of effort has been put into optimizing the Apache’s C code for performance. As a result, it runs faster and consumes less systems resources than many other servers.
5 “FixEmUp” TOOLS

“FixEmUp” web-application has been designed based on user friendly interface and features. The smartest and advanced technology such as, Google map API for map, MySQL and PostGIS for smart database, Django for frameworks, PHP and HTML for GUI and most important Apache for smart and reliable server has been used for “FixEmUp”. These make this application so much efficient and reliable. To understand the back-end technology of “FixEmUp”, it is important to take a short overview of every language that has been used.

5.1 HTML

HTML stands for HYPER TEXT MARKUP LANGUAGE is a language for describing web pages. “The word ‘Hypertext’ was created along with other Internet terms and technologies during the evolution of the Web. It was coined to describe documents that could change, redirect, and otherwise overcome the linearity of normal text. In short, ‘Hypertext’ describes text on the World Wide Web”. (Schafer 2010, 10-15)

Traditional HTML is defined by a SGML (Standardized General Markup Language) DTD (Document Type Definition—see the upcoming section “XML”) and comes in three primary versions (HTML 2, HTML 3.2, and HTML 4). “HTML 4 comes in the following three varieties: transitional, strict, and frameset and in most documents authors use the transitional variant.” (Thomas 2002, 74) HTML 5 is the most current and final version of HTML.

In HTML code text between <html> tags [<html></html>] describe the webpage, text between body tags contains the visible page content, text between h1 and h2 tags contain the headings format and text between p tags contain the paragraph.
An HTML document, it has two main structural elements such as

1. HEAD contains - setup information for the browser & the Web page
e.g., the title for the browser window, style definitions, JavaScript code etc.

2. BODY contains - the actual content to be displayed in the Web page

In “FixEmUp” the design of total application specified with css script (section 4.2). The index page has been written in HTML and all other pages have been integrated with PHP.

By analyzing the back-end of “FixEmUp”, the css script included as

```html
<script>
<link rel="stylesheet" type="text/css" href="http://www.zcoded.com/fixms/css/styles.css" />
</script>
```

In ‘top.html’ page at the HEAD. Also the Google map (section 4.4) JavaScript has placed in HEAD. As top.html page has used all over the application it reduce the runtime of total software and recursively call same interface all time. Although the PHP contains remains different regarding the functionalities.

5.2 Style Sheet Technologies

Markup languages such as, HTML do not excel at presentation as it was not designed for this task. CSS (Cascading Style Sheets) has the control to do this thing. Although in some cases such as, using XML language when the markup transformation may also be required to create the appropriate presentation format, the XSL (eXtensible Style Language) will be used as well.
30

CSS is used to specify the front look/design of a webpage. This technology has been used from Internet explorer 3.0 to all recent browsers. CSS-based styles sheets specify rules that define the presentation of a type of a type, for example, `<h1>` - a group or, more correctly, class of tags or a single tag as indicated by its id attribute. Style sheet rules can be used to define a variety of visual aspects of page objects, including color, size, and position. The various style rules can be combined depending on tag usage thus the “cascading” moniker for the technology.

5.3 PHP

On the other hand the server side language PHP has been used for embedding the MySQL database and HTML for interface. PHP is a general-purpose server-side scripting language designed to dynamic web pages development purpose. According to its official web site at http://www.php.net/, PHP is “. . . a widely used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. (Vaswani 2005) The most advantageous feature of PHP is that it can be embedded in to the HTML source document and PHP processor module interprets it with web server which generates the web page document as well as MySQL. It also has the capability to include a commend-line interface and can be used in stand-alone graphical applications. (Deric 2005) On most web servers PHP can be used as a standalone interpreter on almost every operating system and platform at a free of cost. Another feature of PHP is its rich library. As this language has been used by millions of developers it has almost complete library collection. In “FixEmUp” the PHP5 has been used.

The (`<?php ?>`) is called the PHP tag. Any character outside this tag will be counted as normal HTML. Although there are four styles of PHP tags and all are equivalent.

- **XML style:** `<?php echo ‘<p>Hello World</p>’; ?>`

  This tag style can be used in XML documents. But not recommended to use it normally.
• Short style: `<? echo 'Hello world.';?>`

  This tag style is the simplest and follows the style of Standard Generalized Markup Language (SGML). We need to enable short_open_tag option to use the style on PHP server.

• SCRIPT style: `<script language='php'>echo 'hello world.';</script>`

  This is the longest version of PHP tags and similar to JavaScript or VBScript.

• ASP style: `<% echo 'Hello world.';%>`

  This tag style used in ASP (Active Server Pages) or ASP.NET. By enabling asp tags on PHP server anyone can use it.

PHP’s main competitors are ColdFusion, Microsoft ASP.NET, RUBY, and PERL. Although among these PHP has a very strong and significant platform such as

• Performance - very fast, single inexpensive server and the Benchmarks support by Zend Technologies ([http://www.zend.com](http://www.zend.com)) shows significant stronger than its competitors.

• Availability of support and documentation - Zend technologies ([http://www.zend.com](http://www.zend.com)), the company behind PHP development funds to enrich it and also offers support and related software at a commercial basis.

• Cost - as PHP is open source and free any one can download it from [http://www.php.net](http://www.php.net)

• Flexibility and development approach- PHP allows core coding as well as framework support such as CAKE PHP, Codeigniter etc. to its developers.

• Portability – PHP is available for nearly all popular operating systems such as Microsoft Windows, MAC etc.

• Database Integration – PHP supports many database systems such as MySQL, PostgreSQL, and Oracle etc. Although PHP 5 has built in SQL called SQLite.

• Built-in libraries – As PHP has designed for the web, it has many built-in functions to perform significant web related tasks. (Luke 2010)
The recent added feature for PHP is the OOPHP (Object Oriented PHP). OOPHP is most efficient programming module for the java, python or other object oriented programmers. There are lots benefits of OOPHP. Such as Interfaces which Gives the ability for a class to fulfill more than one is-a relationships such as class can inherit only from one class, but may implement as many interfaces as it wants. The final keyword allows you to mark methods so that an inheriting class cannot overload them. Explicit object cloning can clone an object. By declaring a `__clone()` method, which will be called during the clone process, after the properties have been copied from the original object. Abstract classes can be declared by using abstract to prevent it from being instantiated.

The most interesting feature for which PHP is most famous is the dynamic content. This is most important because content that changes according users need, will grab visitors to using the site. Such as, the example code below.

```php
<?php echo '<h1>It is</h1>'.date('H:I, js F Y')?>
```

Where the date is dynamic and will change according to user's requirement at a specific format. In “FixEmUp” one of the biggest features is dynamic application which only possible for using PHP. By giving a deep look at application it can be quite visible that the full complaint box and the form are dynamic. At the complaint report box, the new complains are automatically added as the PHP dynamic feature has been used here so smartly. One of the common disadvantages of using server side language is its long runtime. But in “FixEmUp” this type of special case handled so intelligently. At the first run of this application it stores some data at cache memory so that after the 1st run the total run time of application being half from previous.

ERROR handling efficiently is another great feature of OOPHP as well as PHP. In “FixEmUp” error handling and bug detection was a great challenge that code whit PHP. But PHP’s advance functions, such as, mysql_error(), debug_backtrace(), error_log() etc. make the procedure so simple and efficient. One of the powerful features of PHP is
its manual error handling. Such as, other server site programming language PHP enables a client-server runtime environment but the auto generated error handling makes the difference.

5.4 MySQL

“MySQL is the database construct that enables PHP and Apache to work together to access and display data in a readable format to a browser. It is a Structured Query Language server designed for heavy loads and processing of complex queries. As a relational database system, MySQL allows many different tables to be joined together for maximum efficiency and speed.” (Naramore 2005) MySQL is an open source, Enterprise-level, multi-threaded, relational database management system. It is actually a program that can store large amounts of information in an organized format that’s easily accessible through programming language such as, PHP. MySQL actually runs two programs at a time as MySQL operates using client/server architecture. (Robin 2009, 137)

As Naramore (2005) explains, there are so many good features of MySQL such as:

- kernel threads ensures Multiple CPUs usability
- Multi-platform operation
- Every types of data virtually covered by numerous column types
- Sorting and mathematical calculations performed by group functions
- Commands that allow information about the databases to be easily shown to the administrator
- Function names that do not affect table or column names
- A password and user verification system for added security
• Up to 32 indexes per table permitted; this feature has been successfully implemented at levels of 60,000 tables and 5,000,000,000 rows (version 4.1.2, currently in development, will allow 64 indexes)

• International error reporting usable in many different countries (Gerner 2005)

There are some more benefits of using MySQL. Such as:

Reduce record retrieval time: With MySQL database only few queries can retrieve data from a substantial database at shortest time expected.

  e.g.: SELECT (field name) FROM (table name) WHERE (expected data).

Flexible retrieval order: With MySQL it is quite easy to gain data as expected order. Such as, from queries the maximum or minimum sequence can be maintained so easily.

Simultaneous multiple user access to records: MySQL gives the user multiple-user capability so that multiple users can access on a record simultaneously. For application such as, “FixEmUp” it is a necessary feature where lots of users hit at a same time to input or retrieve data.

The way of communication with MySQL takes place via SQL (Structured Query Language). SQL is the standard database language and all major database systems understand it. It is a standard language for specifying accesses and modifications to relational databases. SQL was originally standardized by American National Standards Institute (ANSI) and the International Standard Organizations (ISO) in 1986. SQL was significantly expanded and modified in its early years, the result of which was standardized in 1992. SQL is supported by the databases provided by all major database vendors and is a standard that has truly become the standard. It is used to create, query and modify relational databases, regardless of the particular database vendor. Such as, to create a table the query has to be run in sql server shows in figure 11.

```
CREATE TABLE x ( name CHAR(30), num INT, address CHAR(30), phone CHAR(12) );
```

Figure 11: SQL query example
Where x is the table name and name, num, address and phone are the fields with different values. One important thing for a table field is that for every input a default unique id produces which represent the specific record for a row.

One of the biggest advantages or feature of MySQL is the automotive ER Diagram (Entry Relationship Diagram), the diagram used to display a graphical interface of the back-end system of database. SQL can auto generate ER diagram with APACHE server that being used at the “FixEmUp” server and database. The ER diagram of “FixEmUp” application shows as below.

Figure 12: ER Diagram (FixEmUp application)

In figure 5, the register table and table type has n:1 relationship as many users can be assigned to a particular role. The users have defined as user and admin. It depends on
the user type that which table has assigned for them. On the other hand register table
and table discussion has 1:n relationship as one registered user can comment to as many
reports as he wants. Info table and discussion table has n:m relationship. It manages the
comments on the report. Info table and register table has n:1 relationship. It manages
the user authority that how many reports can be submitted by a user. It also gives
information about the register or non-register users. Info table and status table has n:1
relationship. It conforms that one report can only have one status. Info table and city
table has n:1 relationship. It manages the relation bridge of a city and its reports. Such
as a city could have many reports. Info table and category table has n: 1 relationship. It
manages and conforms that a report has only one category.

Every particular table has different functionality. Such as,

- The register table holds the information when a new user registers. So when
  somebody registers their information goes to 'register' table.

- When a user reports a problem about a faulty street, this information goes to
  info table. Thus 'info' table holds the information of the reports about the streets.

- When a user gives a comment about a report it is stored in the discussion table.
  So 'discussion' table stores all the comments about the reports.

- The city table stores the cities which are supported.

- The status table stores the status of a report.

- The type table stores the information about the type of the user

- The category table holds the category of a report.

Foreign Keys of database:

'Discussion' table has foreign key register_id, info_id;

'info' table has the foreign key register_id, category_id, city_city_id, status_status_id;

'register' table has the foreign key type_id;
5.5 Google map

Google map is a widely used web application which not only provides map based services for free by Google but also produces maps embedded on third-party websites via the Google maps API. The Google map can find locations, using layers, adding locations and layers, navigating and getting coordinates. The best added feature for Google map is its 3d street view and satellite view. People can view there expected area by both graphical map view and satellite image view. Google has explicitly introduced another service that is the navigator system. By Google map people can view the shortest route of their expected places. It also provides the map tagging service by bubble tag mark (place marker icon).

Figure 13: Google map of Tornio city

API- Application Programming Interface, the particular set of rules and specification that a software program (e.g. Google map), can follow to communicate with each other. API response as an interface between different software programs and alleviate their interaction as user interface (ui) of a computer alleviate interaction between Humans and Computers. An API uses for applications, operating systems (os), libraries etc. to define their function calling convocations and can manage specifications for protocols, object classes, data structures and routines which are mainly use for communicating between the consumer program and the implementer program.
Google Map API mostly uses Ruby on Rails. The Rails framework facilitates radical improvements in productivity within its niche: database-backed web applications. Rails is intuitive, powerful and free. Together Rails and Google maps enables the developer to build impressive web applications. (Micheal 2007, 2-10)

Google map API has worked in three categories. Categories discussed below:

The first category APIs are use the Google data protocol (a protocol that provides an opportunity for external developers to write new applications which permits the end user access and update the data stored by many Google products) to provide an easier way for reading and writing data on web. These APIs worked based on Atom and RSS syndication formats and Atom publishing protocol. APIs also have a support for querying, authentication and version conflict detection. On the client side, client libraries are found in any languages so the applications can use.

The second category APIs work across many of the Google Apps products which the extensions and implementations to the gadgets API. And the third categories APIs allow the user to automate tasks across many Google Apps products which are actually services available for Google Apps Script (a java script cloud scripting language and a third party service).

Google API mainly written in JavaScript. Although the libraries written in JSON/XML. But a small amount of library work has been done in PHP. By using some PHP classes does the same.

5.6 PostGIS

On other hand PostGIS is an open source software program which adds supports for geographic objects to the PostgreSql. It has features such as geometry types for points,
line-strings, polygons, multipoint, multiline-strings, multi-polygons and geometry collections. PostGIS have the capability for determining geospatial measurements such as area, distance, lengths. It also can determine geospatial set operations such as union, symmetric difference etc. it use GiST (Generalisted Search Tree) for high speed spatial querying and provides index selectivity support for high speed spatial querying. PostGIS has specially designed and implemented based on “light-weight” geometries. The indexes optimized to reduce disk and memory footprint which really helps servers to increase the amount of data migrated up from physical disk storage into RAM and improve query substantially.

Using PostGIS on PostgreSQL for “FixEmUp” database designing and implementation make the software database so fast, powerful and sustainable. More than that the relational table structure of “FixEmUp” database has been executed so well and secured and it can handle much bigger database enough sufficient for storing and maintaining millions of complains.

5.7 DJango

On the web designing part, Django web framework has determined the total implementation. DJango is a prominent member of new generations of web frameworks. It grew organically from real world applications written by web development team at Lawrence journal world newspaper in fall 2003 while they try to build application with python. (Holovaty 2007, 3)

Django is an object-relational mapper which defines data models entirely in Python. Included with dynamic database access API feature, it also has the sustainability to write SQL. One interesting feature with Django is that it can automatically create interface and update content which is time savvier and efficient to use.
Django can produce craft-free URLs (removing unwanted extensions) as well as can hook into “memcached” or other cache frameworks for great performance. Django provides an optional CRUD (create, read, Update and delete) interface generated dynamically and configured via admin models which creates great flexibility to use the framework. It is lightweight and so much secure.

Figure 14: Django architecture

On the server side, The Django development server, also called the run server, after the commend that lunches it, is a built-in, lightweight web server that can use while developing site. It is included with Django so sites can be developed rapidly, without having to deal with configuring the production server (apache) until it is ready for production. The development server watches the code and automatically reloads it and making it easy to the user to change the code without restarting anything. (Adrian 2007, 19)

Django requires MySQL 4.0 or above. It also works on oracle database. The most impressive feature of Django is it automatically escapes all special SQL parameters according to the quoting conventions of the database server which prevents high risk threads. This framework also designed to prevent session forging and Cross-Site Scripting (XSS) which is most effective to prevent hacking. By use of this framework “FixEmUp” becomes so safe and flexible at the web.
The basic components that DJango requires Python 2.3 or later, DJango, a supported sql database, apache 2.X and mod_python 3_x. After installing the components DJango will be ready to use.

Figure 15: Django framework interface
6 ARCHITECTURAL DESIGN

In this chapter the architectural design of “FixEmUp” application is described as well as an analysis of components and components design. This chapter also provides details about the different approaches of components design.

Richard (2005, 250) have elaborated the functions of software architecture, architectural style, and the role of object-oriented approach to representing these styles. Architectural designs focus on the architectural level of system design— the gross structure of a system as a composition of interacting parts. They are primarily concerned with

1. System structure — the high-level computational elements and their interactions.
2. Rich abstractions for interaction. Interaction can be simple procedure calls, shared data variable, or other complex forms such as pipes, client-server interactions, event-broadcast connection, and database accessing protocols.
3. Global properties — the overall system behavior depicting such system-level problems as end-to-end data rates, resilience of one part of the system to failure in another, and system wide propagation of changes when one part of a system such as platform is modified.

According to Mohapatra (2010) the nature of computations required to solve a given problem and the quality attributes of interest govern the choice of an architectural style. In practice, most software systems do not follow any particular architectural style; rather they combine different styles to solve a design problem. Shaw (1998) identifies three ways to combine architectural styles. They are the following:

1. Hierarchical heterogeneous style
2. Simultaneous heterogeneous style
3. Locational heterogeneous style
However, in “FixEmUp” application object-oriented architecture has been used. Computation is a process which followed by a well-defined model and expressed in an algorithm, network topology, protocol of object oriented architecture has been restricted to fixed number of operations and for each element of a set of entities. The main advantage of this architecture is reusability and modifiability.

6.1 Analysis of components

“Component” in software is a rough synonym for “module” or “unit” or “routine”. It began by referring to the source code in programming language. Some of these terms have now acquired additional precision; for example, “module” may mean an abstract data definition, or an object-oriented class (Kung 2004). Clemens Szyperski has cut through much of the difficulty by removing the focus from the code source. He defines a software component as executable, with a black-box interface that allows it to be deployed by those who did not develop it (Szyperski 1998, 554).

The paradigm of component-based development (CBD) has been introduced as an excellent solution for building complex enterprise-scale information systems (Brown and Wallnau 1998; Szyperski 1998). CBD provides organizations with a method for building flexible, enterprise scale solutions that are able to accommodate ever-changing requirements in a cost-effective, timely manner. “Following the CBD approach, system development becomes the selection, reconfiguration, adaptation, assembling, and deployment of encapsulated, replaceable, and reusable building blocks called components, rather than building the whole system from scratch. CBD provides higher productivity in system development through reusability, more effective system maintenance, higher quality of solutions, and the possibility for parallel work.” (de Cesare 2005, 86-102)
It is so important for a software validation procedure. As de Ceasare (2005) suggests component provides better system adaptability through the replace ability of parts, localization and better control of changes, system scalability, and possibility of using legacy assets. Many software systems many similar or even identical components that are developed from scratch over and over again have led to efforts to reuse existing components. Structuring a system into largely independent components has several advantages (Sametinger 1997, 1-10).

Components reuse methodology is one of the important parts of software engineering. A software component can be deployed independently and is subject to compositions by 3rd parties. (Szyperski 1997, 1-10) To build software systems, software components almost automatically lead to software reuse. One of the strongest benefits of software reuse is Quality improvements (Sametinger 1997). In “FixEmUp” software components reuse has done so perfectly. In the back-end coding part, various functions reused significant times. The advantage of it is to reduce the runtime. More than that in design part many components has been reused.

6.2 Components modeling and design

The goal of this chapter is to describe component technology and its design part. This chapter does not intend to provide an in-depth coverage or present the different facets of these technologies. The component model underlying each component technology will be tried to instead identify. This will allow identifying similarities and differences, both between the various component technologies and current component technologies.

“A component model specifies, at an abstract level, the standards and conventions imposed on software engineers who develop and use components. Compliance with a component model is one of the properties that distinguish components from other software entities.” (Larsson 2003) As Larsson (2003) maintains, components represent the computational elements and data stores of a system. There are no restrictions on the size of a component. A component can be as small as a button in a graphical user
interface or as large as a Web server. For large and complex components, it is essential to be able to hide their internal structures. “The interface of a component, that is, its external view, is described as a set of ports. The ports are the points of interaction between a component and its environment.” (Crnkovic 2003, 67)

According to Basem (2010, 98), component-based design approaches for embedded systems address in a unified way both hardware and software components. They can handle constraints on performance and dependability as well as different cost factors. Component-based design is a bottom-up approach. To produce the predefined platform, it assembles existing heterogeneous components by inserting wrappers between these components.

Basem (2010, 79) states on his writing that there are two main issues that the component-based design approaches need to handle. One is the presence of heterogeneous components. The components description requires concepts and language supporting explicit behavior, time resource and their management because the hardware components are inherently parallel and synchronous. The second one is the predictability of basic properties of the designed system. The ability to describe formally the concurrent behavior of interacting components is a key aspect in component-based design. It is necessary that theoretical results be integrated into logical component-based design flows, validated through comparison with existing industrial practice. (Basem 2010, 79)

Connectors represent interactions between components. Such as components, connectors can be of arbitrary complexity. This notion makes it possible to represent, in a unified way, simple concepts such as a method call as well as complex and abstract connections such as an SQL connection between a client and database server. (Crnkovic 2003, 67-69) In “FixEmUp” application connectors do the same as it discussed. The SQL connections use in PHP scripts make connection between database and software. The apache server has been used here as the server.
The main advantage of component-based system development is the reuse of components when building applications. Instead of developing a new system as initio, components already existing are assembled to give the required result. Systems are being increasingly constructed using off-the-shelf software components but this is not a straightforward procedure. (Crnkovic 2003, 155) In “FixEmUp” reusing components has been applied many times. It decreases the software runtime and increases the flexibility. Moreover, recursive functionality increases the system accuracy during runtime and makes the system faster.
7 DESIGN VALIDATION

The software design validation activity occurs after the software requirements have undergone the software V&V process and the software design, or an increment of the software design, have been completed. According to the model of software development, the software V&V process may be exercised on the entire software design or software design increments, but always traceable back to the software requirements. The software V&V tasks of traceability, evaluation and interface analysis provide assurance that software requirement are not misrepresented, incompletely implemented, or incorrectly implemented. By verifying the software design meets its software requirements, the software design V&V activity also supports validation that the software design meets system requirements. (Wallace 1996)

In general there may be some rules for the design validation. According to Wallace (1996), he shows a procedure for the design validation.

- Conduct a software design tractability analysis – Trace software design to software requirements and vice versa. Check the relationship for accuracy, completeness, consistency and correctness.
- Conduct a software design evaluation.
  - Evaluate the software design for accuracy, completeness, consistency and correctness and testability.
  - Evaluate software design for compliance with software design standards, language standards if appropriate and software engineering practice.
  - Access software design against assigned quality attributes.
- Conduct a software design interface analysis – evaluate software design for accuracy, completeness, consistency and correctness of hardware, operator and software interface requirements.
- Verify the software requirements for assertions, responses to assertions and other required system algorithm and integrity checks or fault tolerance portions
have been designed into the software. Check that the software design is complete and accurate and will not adversely affect system performance.

- Coordinate with software integration test planning.

Naturally, the process of software design begins when there is a need, an impact. According to Basem (2010, 177), people create the need whether it is a problem to be solved, e.g. if a functionality or use interface is not user friendly, then the GUI needs to be redesigned, or a new invention. Design objective and scope are critical in the impacts stage. A design project charter should describe simply what is to be designed. It cannot be vague. It is the first step. The second step is the design details, which should be a complete version. In step 3 the design should have several solutions of its implementation procedure. It is very important to draw every idea in a paper to create a clear concept about the design. The 5th stage is to keep track of all the ideas and develop solutions in the preliminary design stage. Step 6 is the validation and verification of the design as well as step 7 is the lunch of the software. Below, the flowchart shows the steps graphically. (Basem 2010, 179)

![Figure 16: Software Design procedure identification by Basem(2010)](image)
In “FixEmUp” design, the Evolutionary Development Model (EVO) has been used. The aim to use the model is that it helps the system to work faster. According to Puntambekar (2007, 4-15), “in this approach of system development, the initial prototype is prepared and it is then refined through number of stages to final stage.” The principle objective of evolutionary model is to deliver the working system to the end user. As Puntambekar (2007, 4-15) states the advantages of this system is

- Fast delivery
- User involvement during developing the system.
- Prospect of more useful system.
- Specification, design and implementation work in coordinate manner.

Although in some cases this model is not good enough to use as where there is no specification. But in “FixEmUp” there was no management and maintenance problem as well as verification problems which are discussed by Puntambekar as the drawbacks of EVO modeling.

On the architectural part Agile modeling has been used. Agile software development method imposes as little overhead as possible in order to develop software as fast as possible and with continuous feedback from the customers. “Agile methods have in common that small releases of the software system are developed in short iterations in order to create a running system with a subset of the functionality needed for the customer.” (Stamelos 2007, 93) The agile methods are based upon 12 principles (Beak, 1999). The principles are planning game, small releases, simple design and refactoring.

In “FixEmUp” application these principles are strongly maintained during the development. At the beginning a sample of the software with all the technical and non-technical attached has been released. It was the planning part. Then after a nearly complete development “FixEmUp” has its first demo release. Then after some modifications the software is in current condition. And the plan is for some more releases and a complete version. After every release the defects were detected and that
has named as refactoring. The refactoring is as important as it removes all the bugs from the software and makes it perfect.

In the highly dynamic process used in Agile methods, terms and organizations need automated tools and technics to support their work without much time (Stamelos 2007, 93). Automation can be used to detect quality defects such as code smells (Flower 1999) and bug patterns (Allen 2002), etc. In “FixEmUp” development these criteria has maintained so well and in an organized way.
8 DEMO IMPLEMENTATION

The objective of “FixEmUp” application demo implementation is to introduce a new technology to the Tornio city citizens as well as the civil service authorities. For the demo implementation of “FixEmUp” it maintains some steps as it follows the agile software development on its back-end architecture. Agile methodologies focus on quick turn-around and frequent design or develop or release rather than up-front planning (Leppik 2005). Under Agile development the development term cycles through project phases several times, each time producing a more sophisticated application (Leppik 2005).

“FixEmUp” implementation discuss below step by step.

8.1 Apache server setup

To test the “FixEmUp” functionality first it has been tested at the local server. An apache server has been set up as a local server. Apache HTTP server (version 2.2.21) has been used for it.

The main feature of this server is:

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MySQL Database, Database backup, Sbackup</td>
</tr>
<tr>
<td>2.</td>
<td>PHP5</td>
</tr>
<tr>
<td>3.</td>
<td>FTP access / SFTP access</td>
</tr>
<tr>
<td>4.</td>
<td>SFTP / FTP</td>
</tr>
<tr>
<td>5.</td>
<td>SSH access</td>
</tr>
<tr>
<td>6.</td>
<td>SSI</td>
</tr>
</tbody>
</table>
It is necessary to test the software before posting it at the real server. XAMPP server has been used in this case. XAMPP is an Apache distribution containing MySQL, PHP and Perl. It is a free server capable of serving dynamic pages. The server is available for all the popular OS’s such as Microsoft Windows, Linux, Mac OS X etc. The XAMPP name comes from, X (meaning of cross platform), A (Apache HTTP Server), M (MySQL), P (PHP), P (Perl).

For the demo implementation, XAMPP server 1.7.7 (stable) has been used. The feature of this version is:

Apache 2.2.21, MySQL 5.5.16, PHP 5.3.8, OpenSSL 1.0.0e, phpMyAdmin 3.4.5, XAMPP Control Panel 2.5, Webalizer 2.23-04, Mercury Mail Transport System v4.72, FileZilla FTP Server 0.9.39 and Tomcat 7.0.21, with mod_proxy_ajp as connector.

To install the XAMPP server, the easiest way is to use the installer version. After installation is completed, the XAMPP Control Panel to start/stop all servers and also install/uninstall services.
After enable the server the whole package of “FixEmUp” application has been placed at XAMPP directory. The main directory for all WWW documents is \xampp\htdocs. So the FixEmUp access it with the URI "http://localhost/fixemup/index.html".

8.2 Setting up database

At the database end MySQL has been used. It is the world's most popular open source database software. With its consistency processing speed, reliability and ease of use, MySQL has become the best choice for web and web 2.0 (Ulman 2006, 1). At demo implementation MySQL version 5.5.16 has been used.

The configuration file of mysql is located at: /etc/mysql/my.cnf.
In MySQL, the existing database of “FixEmUp” has been imported. It is recommended to set up a password of existing server. However, by default MySQL creates user as root and runs with no password. The following statements were executed to set up the Root Password.

```
mysql -u root
mysql> USE mysql;
mysql> UPDATE user SET Password=PASSWORD('new-password') WHERE user='root';
mysql> FLUSH PRIVILEGES;
```
8.3 Local Server Testing

After implementing all the steps “FixEmUp” is ready to test for local server. At the local server site the address was http://localhost/fixemup/. After running the link it looks like figure 19.

![FixEmUp at local-server](image)

Figure 19: FixEmUp at local-server

8.4 Setting up ftp software

After testing the software in the local server perfectly the next step was to set up in web server. For “FixEmUp” web upload, a free ftp server named coreFTP was used.

FTP stands for the file transfer protocol. It is a standard way to send and retrieve files across the internet (Teare 1996, 31).
coreFTP is free and it is reliable. Some features of coreFTP such as:

- latest encryption methods has been used for Encrypt and decrypt files to servers
- IBM information exchange support with commend line updates
- Schedule ftp transfers
- Email notification, zip support, view thumbnails images of remote directories etc. (http://www.coreftp.com/)

With the coreFTP software all the files of “FixEmUp” application has transferred to the web server.

8.5 Setting up domain name

“Web hosting and domain name registration are usually separate services. Although the hosting company may give or sell a domain name with website, most hosting companies are not domain registrars and chances are they are procuring it from a third party” (Greenberger 2010). It is important to pick an easy to remember web address. In case of “FixEmUp” software the chosen name is http://www.zcoded.com/fixms/. One of the basic purposes to choose this name is it is easy to remember.
8.6 Testing at server side

After transferring the complete data to fixemup.tk server and create the database at the apache server now it is the time for testing the total system.

- The first step for this testing is to run the software in default browser and check all the functionality and links that are used in the software.

- Second step is the browser testing. To make sure the software is compatible for different browsers it is recommended to test the software in all popular web browsers such as Internet Explorer, Mozilla Firefox, Opera, and Google Chrome. For Mac users it is suggested to test the software on safari browser, as safari is the default browser of Mac OS.

- After browser testing successful it is time to test the software in different operating systems. Sometimes web software faces problems for different operating systems. So it is important to test it before public.
9 DISCUSSION AND CONCLUSIONS

This chapter contains the discussion on result of the thesis, limitations and challenges of “FixEmUp” software’s implementation. This chapter also includes the plans and directions for further research and a conclusion of thesis.

“FixEmUp” is web based application that has all the functionalities and technologies of an ideal web application. All these properties have been described so elaborately in this thesis. The thesis describes “FixEmUp” from a developer’s point of view and it successfully to do so. In this thesis, information about system security, back-end architecture of the software, software design with appropriate design methods and analyses these methods etc. have been reviewed as well as the future aspects and plans for this software discussed.

Furthermore, further research and development work has been proposed and highly encourage. As the software is open source so there is a very big space for the students and researchers to work on it. Students are motivated and appreciated to work on this software and meet with the latest technologies of web. More than that developers have enough room to prove their creativity to work on this software and make it more efficient.

9.1 Discussion of the results

The result of this thesis is “FixEmUp” demo integration and implementation with future plans. New concepts of digital complaint box have been revealed with the analysis of this software. The software is a combination of technology and service which are detailed by the thesis. Models such as evolutionary model, Agile model represent the uniqueness and advanced technologies.
The objectives of this thesis were to analyze the architecture of “FixEmUp” as well as analyze the demo implementation and prospect of usability of this software. During analyzing “FixEmUp”, some core components of this software have figured out.

- The software has written in different programming languages such as HTML, CSS, PHP and JavaScript
- The database was made by Apache server where MySQL language has been used to write the database.
- Framework such as Django has been used for core implementation.
- The software follows Agile modeling on its architectural part.
- It uses the component base design on its designing part.
- The software follows evolutionary model concept.

After analyzing the fundamental architecture of “FixEmUp” design, the working mechanism has been described briefly as well as the demo implementation procedure. Since there used many protocols of software development and deployment, the logic has been established briefly in support of reasons to use protocols or methods that are used in “FixEmUp” development.

“FixEmUp” has the recent updated advance database system which makes the dynamic system more reliable. Some technologies such as SQL injection system, cross scripting prevention system, fishing preventing session etc. makes the software so secure that prevent the hacking at its best. The searching system of this software is so powerful which make it so fast and user friendly.

One of the basic objectives of the thesis is to describe the dynamic database system of “FixEmUp”. The software has been manipulated with core dynamic system at its database and back-end architecture. The database has managed by Apache server. Use of foreign keys on the database tables help to create relations between tables which integrate an automotive environment.
“FixEmUp” is designed with powerful web languages such as HTML, CSS and JAVA script. This application supports all operating systems, including Windows (98/vista/7), Linux and Mac OS. By using the software, a municipality of any country can come closer to the citizens or residence as they can directly communicate and share problems with each other. The governmental service will be fast then.

9.2 Limitations of the system

Despite the fact that “FixEmUp” has full functionality activated, there are still some limitations on it. As the software is in its demo version running and the target users are Tornio town citizens and the civil service authorities, currently the software is unavailable for other cities of Finland. However, the future plan is to make “FixEmUp” available for the users of all over Finland. Furthermore, if the government of other countries wants the people of those countries can use this application.

Another limitation of “FixEmUp” is that, it uses the Google map for its location detection and tagging. The plan was to use the map which has been using by the Tornio municipality but for some security reasons the Municipality deny to provide it. But the future plan is to use that map with the proper license and permission.

There are still some limitations for the users to complain a problem. Such as now at the demo version, a user only can give limited complains. However, the full version software has planned to give the freedom to its registered users to submit as many problems as they wish.
REFERENCES

Print-based sources:


Chaudron, Michel R. V. & Szyperski, Clemens & Reussner, Ralf (Eds.) (1999), Component Based Software Engineering. Addison-Wesley Professional, USA

Crnkovic, Ivica & Larsson, Magnus (2003), Building Reliable Component-Based Software Systems. Artech House, Inc., 685 Canton Street, Norwood, MA.


Gottesdiener, Ellen (2005), The Software Requirements Memory Jogger. Goal Q P C Inc, 8E Industrial Way, Suite 3, Salem, NH, USA.

Greenberger, Josh (2010), Introduction to Website Hosting and Affiliate Income.

Gutamans , Andi & Bakken, Saether, Stig & Rethans, Deric (2005), PHP 5 Power Programming. Prentice Hall PTR, USA.


King, F., Stephen & Brown, Paul (2007), Fix my street or else: using the internet to voice local public service concerns. UK


Larry, Ulman (2006), MySQL. Peachpit Press, Eighth street, Berkeley, CA.

Leppik, Peter & Leppik, Devid (2005), Gourment Customer Service. Vocal Laboratories Inc. 8421 Wayzata BLVD., Suite 260, Golden Valley, MN.


Masalakowski, Mark, Sam’s teach yourself MySQL in 21 days. Sams Publishing, USA.


Mohapatre, P.K.J (2010), Software Engineering. New Age International Pvt Ltd Publishers, Ansari Road, Darya Ganj, New Delhi, India.


Sametinger, Johannes (1997), Software engineering with reusable components. Springer-Verlag Berlin Heidelberg, New York, USA.


Stametos, Loannis G. & Sfetsos, Panagiotis (2007), Agile Software Development Quality Assurance. Information Science Reference, Chocolate Avenue, Hershey, PA, USA


Vaswanin, Vikram (2005), How to Do Everything with PHP and MySQL. McGraw-Hill Osborne Media, Columbus, OH, USA.


Young, Michael (2007), Google Maps Mashups with Google Mapplets. Apress, USA.
Not - printed:

<http://www.apache.org/foundation/how-it-works.html>

<http://www.oldversion.com/Core-FTP-LE.html>


<http://eli.thegreenplace.net/2011/06/24/how-django-sessions-work-introduction/>

<http://www.site.uottawa.ca:4321/oose/index.html#evolutionarymodel>

<http://econym.org.uk/gmap/>

<http://code.google.com/apis/maps/index.html>

<http://www.greatstatistics.com/serverstats.php>


<http://www.htmlcodetutorial.com/document/>


<http://www.hotjoomlatemplates.com/blog/joomla-installation/6-installation-of-the-web-server-on-homeoffice-windows-computer>


Introduction to HTML 1998. 2.3 Naming scheme for HTML documents. Downloaded 8 December 2011.

<http://www.utoronto.ca/webdocs/HTMLdocs/>


<http://lpmj.net/8.php>


MarkASpot 2010, Version 1.4: On the road to dialogue platform. Read at 23 December 2011


<http://www.mysql.com/why-mysql/>

<http://oreilly.com/catalog/0636920014034>

<http://www.slideshare.net/neotohin/introduction-to-lamp>

<http://searchsoftwarequality.techtarget.com/answer/Functional-requirements-and-nonfunctional-requirements-in-software-engineering-explained>


<http://www.djangobook.com/en/1.0/chapter01/>


<http://devzone.zend.com/article/627>
APPENDIX

Questioner

1. Which City do you live?

1.1. How long you are living in this city?
   a. less than 10 years  b. 10 to 30 years  c. this is my birth place

2. Do you complain to municipality if you see a problem related to street furniture or social problems?
   a. yes  b. no  c. I do not want to answer

3. Why do you want to report a problem by yourself?
   a. to keep the city clean and safe  b. just to inform authorities

4. Do you think if online complain box is introduced it will be easier to complain rather than the current complaining system?
   a. yes  b. no  c. I want both

5. Do you want to track the status of your complain?
   a. yes  b. no

6. Do you want to check the map when you complain?
   a. yes, of course  b. no  c. neutral  answer
7. Do you want other citizens can see what you report or you want to keep it secret?
   a. everyone should see b. I want to make a secret complain

8. Do you think online complain box will make the process faster?
   a. yes obviously b. I don’t think so

9. Would you like to attach a picture of the issue while you report online?
   a. yes make it mandatory b. yes sometimes, it could be useful c. not necessary

10. What kind of problems you want to report?
    a. Street furniture b. Daily life issues and street furniture c. I am not sure yet