SIMULTANEOUS TRACKING AND RANKING

A PROJECT REPORT

Submitted by

MADHAN. B (210513104057)

PHILIP JOEL DCRUZ. R (210513104070)

SANTHAN PRAKASH. T (210513104083)

in partial fulfillment for the award of the degree

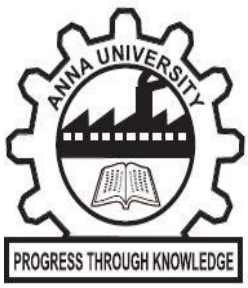
of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING

DMI COLLEGE OF ENGINEERING, CHENNAI



ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report **"STARK – Simultaneous Tracking and Ranking"** is the bonafide work of **Madhan. B (210513104057), Philip Joel DCruz. R (210513104070), Santhan Prakash. T (210513104083)** who carried out the project work under my supervision.

SIGNATURE

Dr. A. Umamageswari

Head of the Department

Computer Science and Engineering

DMI College of Engineering

Palanchur Nazarathpet Post

Poonamallee Chennai-600123

SIGNATURE

Dr. A. Umamageswari

Supervisor

Head and Professor

Computer Science and Engineering

DMI College of Engineering

Palanchur Nazarathpet Post

Poonamallee Chennai-600123

Submitted for the Viva Voce conducted on

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We are immensely pleased in taking up this opportunity to thank the **LORD ALMIGHTY** for showering this unlimited blessing upon us.

We take this opportunity to express our gratitude to our Founder and Chairman **Rev.Fr.Dr.J.E.Arul Raj**, our Correspondent **Rev.Sr.I.Arokiamary**, **M.Tech.,(Ph.D).,** and our Administrator **Rev.Sr. Hamlet**., for allowing us to take over this project.

We thank our Principal, **Dr.M.A.Leo Vijilious, M.E.,Ph.D.,** who has always served as an inspiration for us to carry out our responsibilities also providing comfort zone for doing this project work.

We thank our Dean, **Dr.Suresh Mohan Kumar**, **M.E., Ph.D.,** for his valuable support and providing necessary comforts for doing this project.

We wish to express our sincere thanks to **Dr.A.Umamageswari,M.Tech., Ph.D.,** Head of the Department and **Mr.S.Selvin Ebenezer, M.E.,** Project Coordinator for their scintillating discussion and encouragement towards our project.

We express our deep sense of gratitude to our esteemed Supervisor, **Ms.S.E.Benita Galaxy, M.E.,** for constant guidance and co-operation during the project work.

It is a pleasure to acknowledge our indebtedness to all **the Teaching and Non-Teaching staff** of Department Computer Science and Engineering who aided us successfully to bring our project as effective one.

Further thanks to our **parents, our family members and friends** for their moral support

MADHAN. B (210513104057)

PHILIP JOEL DCRUZ. R (210513104070)

SANTHAN PRAKASH. T (210513104083)

ABASTRACT

STARK (Simultaneous Tracking and Ranking) is a web application which comes handy with Telecommunication companies for knowing their product's major faults and most frequently appearing comments using a Big Data methodology called the Word Cloud. A Word cloud (tag cloud or weighted list in visual design) is a visual representation of text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text. The Word Cloud is an algorithm commonly used in big data to bring an image of words with varying font size based on their number of occurrence in a text. Tags are usually single words, and the importance of each tag is shown with font size or color. This format is useful for quickly perceiving the most prominent terms and for locating a term alphabetically to determine its relative prominence. When used as website navigation aids, the terms are hyperlinked to items associated with the tag. Thus the web application wisely employees the Word Cloud algorithm to form the word cloud image based on the comments and faults provided by the users of the web application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO STARK

Since the introduction of ISPs our society has come across various problems. People are made to wait for ages for their complaints to be recognized. Wait no more STARK is here. STARK provides a major solution to this. STARK stands for Simultaneously Tracking and Ranking. Our application STARK goes through millions of complaints simultaneously and provides instantaneous solutions. STARK keeps track of the complaints that it goes through and ranks them accordingly based on their repetition .STARK takes away the complication of handling complaints on both customer and service provider side.

STARK uses a simple algorithm called WORD CLOUD ALGORITHM to make this process possible. The most repeated problems among the complaints received from the user are tracked and are finally displayed in larger font size while the least repeated words are displayed in smaller size comparatively. For simplicity of the application the non lexical words such as "is", "was", etc., are not taken into consideration by the application itself. Hence STARK focuses on providing solutions to major complaints first and vice versa. The final output for a set of complaints is displayed in the form of an image with all the problems summarized in it.

1.2 MOTIVATION

Maintaining a good customer satisfaction is a primary focus of every ISP, however, it is easier said than done. But the way they handle customer complaints isn’t very efficient, which can lead to customers getting frustrated and dissatisfied. Consequently, the reputation and respect the company has gained through their efforts will be greatly affected. Therefore, we have developed a web application for handling customer complaints more quickly and efficiently than any current method employed.

1.3 PROBLEM STATEMENT

Customers have faced numerous problems with ISPs since the growth of this trend. Another major problem faced by the customers is that their complaints take an immense amount of time to get a response. This causes a galling situation for both the customers and the service providers. The customers wait for a long time and they end up stonewalled before their complaint is opened and help is

provided. The service provider as well faces inquietude. The customer satisfaction drops dramatically and the customer chooses to move away to their competitors. This in turn forces the companies to employ more workers for customer care which affects their revenue. These are the major problems that STARK eliminates.

1.4 GOALS AND OBJECTIVES

- To improve customer satisfaction.
- To reform the methods of complaint handling.
- To make it easier for the technicians to find and fix the problem.
- To provide a dynamic and live status of the complaint to the customer.
- To supply a few temporary and easy fixes that the customer can implement on their own.
- To provide an efficient way for employees to filter and narrow down hundreds of complaints based on multiple criteria.
- To obtain a word cloud of many complaints based on some criteria.
- To provide a quick and efficient communication channel between the customer and the ISP.

1.5 SCOPE

STARK is not limited by the field for which it was originally developed for. It is essentially a complaint handling software and therefore it can be used wherever necessary. Any company that has a customer service department can benefit from implementing STARK. This can be used in a variety of fields such as, e-commerce, telecommunication, movies, restaurants, etc.

By using STARK in any hotel, the management can easily find out what problems their guests encounter. In e-commerce sites this can used to find out what their customers complain about the most about their services. Since this doesn’t need any hardware investment it can easily be adapted to any company Implementing STARK in any company will definitely increase its customer satisfaction level and improve its reputation and value, thereby inflating its income.

1.6 SIGNIFICANCE

There is a great deal of significance for our application using which a lot of benefits are experienced. Right from the customer who files complaint to the administrator who runs the entire company will experience the benefit of this application. The customer is given the satisfaction that their complaint is attended in a quick manner. On the other hand, the immense work load given to the technician is reduced down by a huge margin. While the administrator does not require more man power or labour to work under the customer care section. This increases the revenue of the company and thus various benefits are obtained.

CHAPTER 2
LITREATURE SURVEY

2.1 INTRODUCTION

The STARK is developed by referencing the previous works implemented by many persons. The Literature Survey is mainly dealt in the fields of Word Cloud, Data Analytics and Text Analysis was studied thoroughly. The Word cloud algorithm was studied in order to implement it in a procedural manner.

2.2 RELATED WORKS

I R. Kaptein and J. Kamps. "Word clouds of multiple search results". In IRFC 2011, volume 6653 of LNCS, pages 78–93, 2011.

METHODOLOGY

Search engine result pages (SERPs) are known as the most expensive real estate on the planet. Most queries yield millions of organic search results, yet searchers seldom look beyond the first handful of results. To make things worse, different searchers with different query intents may issue the exact same query. An alternative is to show results in groups. In this paper we investigate if we can use word clouds to summarize groups of documents. We experiment with three word cloud generation methods (full-text, query biased and anchor text based clouds) and evaluate them in a user study. First biasing the cloud does not lead to distinguished relevance and search results hence preferred over other two methods because difference are emphasized. Anchor text is then preferred over full text because of less noisy words.

HIGHLIGHTS

- Since multiple search engines are used, the relevant and corresponding topics and sub topics are mentioned.
- An advantage of word clouds is that there is no need for high quality, grammatically correct text in the documents.
- Better results are obtained through relevance assessment and sub topic matching.

LIMITATIONS

- Due to the usage of three word cloud algorithms the server load increases.
- Anchor text does not contain grammatically correct sentences.
- Any error in one of the algorithm leads to irrelevant results.

II Shailaja Jayashankar, R. Sridaran, "Moving word cloud from visual to text analysis to endow eLearning", Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference, March 2016.

METHODOLOGY

A picture is worth thousand words. Ten words in the form of the picture is can be worth ten thousand. Word clouds are a useful workplace tool for summarizing information. This powerful tool can be incorporated into education as a learning tool to create marvels. Small focus groups were used to obtain student feedback; I contend that it can also be used in assessments. If this enjoyable, fun tool used only for design analysis is extended to text analysis it can brand wonders. With betterment in multi cloud representation its impact in eLearning and research can be all encompassing.

HIGHLIGHTS

- A simple and appealing word cloud can be used as a powerful educational tool.
- The same amount of information obtained by summarizing a paragraph can be obtained by having a glance on word cloud image.
- Most of the word cloud documents are free, easy to use and platform independent.

LIMITATONS

- It can be only used for educational purposes.
- The role of teacher does not play a dominant role anymore**.**

III Peng Wei, Tongge Xu, Xi Qin, Chao Wang "Visualization of Police Intelligence Data Based on Word Clouds", Computational Intelligence and Security (CIS), 2014 Tenth International Conference, Nov. 2014.

METHODOLOGY

This paper studies the method of using word clouds to visualize 110 incidents data. It has been reported that word clouds have become a simple and visual appeal of the visualization method for text. Lots of data are stored in the police intelligence system without a novel visualization

method. In this study, we therefore explore the possibility of using word clouds to visualize 110 incidents data. After processing the data, we construct a word cloud of incident type and a key word cloud of the incident information. The combination of the word clouds techniques and 110 incidents data promote the analyst to explore the deep insights

HIGHLIGHTS

- The word couds can make a contribution to the continued intelligence data mining.
- The public security department can take advantage of information resources into ability to solve cases with the support of word clouds.

LIMITATIONS

- It only has a 38% accuracy.
- Increase in input data will correspondingly decrease the efficiency of the system.

IV Masahiko Itoh, Naoki Yoshinaga, Masashi Toyoda "Word-Clouds in the Sky: Multi-layer Spatio-Temporal Event Visualization from a Geo-Parsed Microblog Stream", Information Visualisation (IV), 2016 20th International Conference, July 2016.

METHODOLOGY

Various events, such as public gatherings, traffic accidents, and natural disasters, occur every day in mega-cities. Although understanding such ever-changing events over all these cities is important for urban planning, traffic management, and disaster response, this is quite a huge challenge. This paper proposes a method of visualizing spatio-temporal events with a multi-layered geo-locational word-cloud representation from a geo-parsed microblog stream. Real-time geo-parsing first geo-locates posts in the stream, using geo-tags and mentioned places and facilities as clues. Temporal local events are then identified and represented by a set of words specifically observed in a certain location and time grid, and then displayed above a map as word-clouds. We detect the locality of events to split them into multiple layers to avoid occlusions between local (e.g., music concerts) and global (e.g., earthquakes and marathon races) events.

Users can thereby distinguish local from global events, and see their interactions over the layered maps. We demonstrate the effectiveness of our method by applying it to real events extracted from our archive accumulated from five years of Twitter posts.

HIGHLIGHTS

- Uncertainties such as traffic management, disaster response and vacation planning are easily solved by this application.
- In case of disasters pre planning can be done and other major event taking place can be removed without any conflict.

LIMITATIONS

- It involves huge amount of calculations due to the live and dynamic nature of the events taken in to account.
- The iterative calculations of dynamic variables lead to server load.

V **Michael Burch, Steffen Lohmann, Fabian Beck, Nils Rodriguez, Lorenzo Di Silvestro, Daniel Weiskopf, "RadCloud: Visualizing Multiple Texts with Merged Word Clouds",Information Visualisation (IV), 2014 18th International Conference, july 2014**

METHODOLOGY

Word clouds are a popular means for summarizing text documents. They usually visualize the word frequencies from single text sources, sometimes along with other attributes. However, the visualization of several text documents in one word cloud has rarely been addressed so far. This paper presents Rad Cloud, a technique for text visualization based on multiple word clouds merged into a single view. Inspired by the Rad Viz approach, the words are radially arranged in an overlap-free layout. The text sources are indicated by the spatial word arrangement and stacked bar charts. The approach has been implemented in an interactive text visualization tool and its usefulness is illustrated by an example.

HIGHLIGHTS

- Multiple word clouds are merged to a single word cloud for quicker understanding.

- The words are radically arranged in an overlap-free layout.

LIMITATIONS

- Since multiple word clouds are combined into one our understanding of individual word clouds is limited.

VI Steffen Lohmann, Florian Heimerl, Fabian Bopp, Michael Burch, Thomas Ertl "Concentri Cloud: Word Cloud Visualization for Multiple Text Documents", Information Visualisation (iV), 2015 19th International Conference on, July 2015.

METHODOLOGY:

Word clouds provide a simple and effective means to visually communicate the most frequent words of text documents. However, only few word cloud visualizations support the contrastive analysis of multiple texts. This paper introduces ConcentriCloud, a layered word cloud layout that merges the words from several text documents into a single visualization. The weighted words are arranged in a concentric layout, with those representing the individual documents on the outer circle and the merged ones on inner circles. Interaction techniques allow to analyze the word cloud composition and to provide details on demand. The approach has been implemented and tested on several examples. A qualitative evaluation indicates the general value of ConcentriCloud and reveals benefits and limitations.

HIGHLIGHTS:

- A ConcentriCloud is composed of several word clouds representing different combinations of the text documents.
- The word clouds are arranged in a concentric layout, with those representing the individual documents on the outermost circle and the merged ones on inner circles.

LIMITATIONS:

- If the analyzed documents do not have any single word in common, the inner circles would be empty and only the word clouds on the outermost circle would display terms.

- In case of text documents that share (nearly) all words, the outer circle would be empty and terms would only appear in the word clouds of the inner circles.
- Smaller words may be added to the word cloud if larger ones do not fit in the remaining screen space.

VII Carla Binucci, Walter Didimo, Enrico Spataro, "Fully dynamic semantic word clouds", Information, Intelligence, Systems & Applications (IISA), 2016 7th International IEEE Conference, 19 December 2016.

METHODOLOGY:

Word clouds are popular graphical representations used to effectively summarize the core content of a document, or of a set of documents. In particular, in a semantic word cloud the positions of the words also reflect their semantic correlation. Several algorithms have been proposed in the literature to compute semantic word clouds, both in a static and in a dynamic setting. However, so far, solutions proposed to generate dynamic word clouds assume that all text changes are known in advance. In this paper we propose a new dynamic scenario. We assume to receive a text stream in a given time period, and we compute, in a sort of real-time fashion, a dynamic word cloud that shows the evolution of the stream. The visualization should preserve both the semantic relationships between words and the user's mental map. We present an algorithmic framework and the results of an experimental study that compares different solutions for the steps of its algorithmic pipeline.

HIGHLIGHTS:

- To compute the geometric layouts, we used algorithms for static semantic word clouds plus a corrective force-directed procedure that takes into account user's mental map preservation.
- Words in the drawing not only change their positions, but they may also appear/disappear several times.

LIMITATIONS:

- It requires a system of considerably high specifications in order to run.

- There is a certain delay time from the istant in which the words are acquired and that in which the evolution is showed.

VIII Ming-Te Chi, Shih-Syun Lin, Shiang-Yi Chen, Chao-Hung Lin, Tong-Yee Lee, "Morphable Word Clouds for Time-varying Text Data Visualization", IEEE Transactions on Visualization and Computer Graphics Volume: 21, Issue: 12, Dec. 1 2015.

METHODOLOGY:

The proposed morphable word cloud generation which consists of three main steps: data preprocessing, shape sequence generation, and word cloud motion generation. The input to the proposed method is a set of text documents with a sequence of corresponding key-shapes. This study aims to generate a morphable word cloud, in which extracted keywords are arranged in their corresponding shapes. At preprocessing stage, document analysis is performed to extract significant words from input text documents. Note that we focus on the morphable word cloud generation rather than analyzing word and semantic significances. Therefore, significant words are extracted simply based on the occurrence frequency of each word after a text document is tokenized into a set of words. For more details on the significance analysis, please refer to [5], [13]. In shape sequence generation, a simple interface with the function of feature point selection is provided for users to define the mapping between the boundaries of two keyshapes. The intermediate shape boundaries are then generated through linear interpolation. Thereafter, the rigid body dynamics is performed to arrange the extracted words in theirs corresponding shapes under various constraints.

HIGHLIGHTS:

- The generated word clouds are inherently overlap-free because of rigid body dynamics.
- With the use of the orientation and position constraints, the proposed method offers several fundamental editing functions, including fixing a word-tag in a specific position, changing the position and orientation of a word-tag, and arranging word-tags in a specific direction.

LIMITATIONS:

- A word-tag cannot be arranged inside another word-tag.
- This method cannot well handle a dumbbell-like shape with very thin connector.
- The generated word clouds cannot represent the changes in the words from a streaming data source.

IX R. Kaptein, D. Hiemstra, J. Kamps, "How different are language models and tag clouds", ECIR-2010, Volume 5993 of LNCS, Pages 556-568, 2010.

METHODOLOGY:

Word clouds are a summarised representation of a document's text, similar to tag clouds which summarise the tags assigned to documents. Word clouds are similar to language models in the sense that they represent a document by its word distribution. In this paper1 we investigate the differences between word cloud and language modelling approaches, and specifically whether effective language modelling techniques also improve word clouds. We evaluate the quality of the language model and the resulting word clouds using a system evaluation test bed, and a user study. Our experiments show that different language modelling techniques can be applied to improve a standard word cloud that uses a TF weighting scheme in combination with stopword removal. Including bigrams in the word clouds and a parsimonious term weighting scheme are the most effective in both the system evaluation and the user study.

HIGHLTGHTS:

- It takes into consideration both tags and bigrams.
- Different modelling techniques are used to improve the word cloud efficiency.

LIMITATIONS:

- It doesn't provide any new practical applications or implementations for Word Clouds.
- There is some discrepancy between good words for query expansion selected by language modelling techniques, and words liked by users.

X **Florian Heimerl, Steffen Lohmann, Simon Lange, Thomas Ertl, "Word Cloud Explorer: Text Analytics based on Word Clouds", System Sciences (HICSS), 2014 47th Hawaii International Conference, 10 March 2014.**

METHODOLOGY:

Word clouds have emerged as a straightforward and visually appealing visualization method for text. They are used in various contexts as a means to provide an overview by distilling text down to those words that appear with highest frequency. Typically, this is done in a static way as pure text summarization. We think, however, that there is a larger potential to this simple yet powerful visualization paradigm in text analytics. In this work, we explore the usefulness of word clouds for general text analysis tasks. We developed a prototypical system called the Word Cloud Explorer that relies entirely on word clouds as a visualization method. It equips them with advanced natural language processing, sophisticated interaction techniques, and context information. We show how this approach can be effectively used to solve text analysis tasks and evaluate it in a qualitative user study.

API USED:

- **Word Cloud Explorer:** A prototypical system that uses word clouds as its main visualization and interaction hub.
- **Stanford CoreNLP Tools:** It performs several processing steps, consisting of tokenization, sentence splitting, part-of-speech tagging, lemmatization, and named-entity recognition.

HIGHLIGHTS:

- Integrates several interactive features into one consistent framework for interactive text analysis.
- Generic focus-and-context techniques and direct interaction with the word cloud.

LIMITATIONS:

- It is a confusing and overwhelming for new users.
- It provides only a statistical summary of the input.

2.5 ALGORITHMS INCORPORATED

The algorithm that is being used in this web application is the Word Cloud algorithm. The basic construct of Word Cloud algorithm is as follows.

1) Text Processing

The data that are needed for the word cloud generation are being fetched first. Then the occurrence of the stop words (it, is, are, etc.) are identified and removed from the context that has undergone to the text processing function.

2) Word Occurrence Frequency

The next step is where two arrays are being created, one for the storage of the words that are being used in the text provided and another is for storing the frequency of occurrence of the words. Thus, the word and its frequency are related using the index number of the array.

3) Word Cloud Generation

Here, after finishing the frequency process the maximum and minimum of the frequencies are being considered in order to find a value called spread. Then a maximum and minimum font size are being chosen which would be already chosen and available, then the following formula will be used to calculate the font size a particular word is to be displayed.

size= minFontSiz + (count – minimumCount) * (maxFontSize - minFontSize) / spread

Where,

minFontSiz = Minimum Font Size,

maxFontSize = Maximum Font Size,

minimumCount = Minimum Count,

count = frequency.

When displaying the word according to the font size that is being calculated based on its frequency a word cloud chart will be generated where most frequent words will be in larger font and low frequent words will be in the most smaller font.

CHAPTER 3

FEASIBILITY STUDY

3.1 INTRODUCTION

The feasibility of the project is analysed in this phase and proposal is put forth with a very general plan for the project and cost estimates. During system analysis, the feasibility study of the proposed system is carried out. This is to ensure that the proposed system is not a burden for the ISPs. We can get a more complete understanding of the system by doing the feasibility study and determine if it is possible to implement this on an industrial scale.

In this chapter, Software development project begins with a feasibility study that considers the technical feasibility, the economic feasibility and the personal feasibility of the project using different realization alternatives.

3.2 ECONOMICAL FEASIBILITY

There are a lot of issues and uncertainties that arise in every customer care service due to implementation on manual labour. In order to cut down on these problems STARK reduces the dependence on employees. Currently, in order to increase the efficiency of the customer care servicer the company has to employ more people to increase productivity. But STARK provides a cheaper and more effective way to increase productivity by giving a simple solution to it. Since STARK is a web application it doesn't need much investment on the company's side. The only pre-requisite is that every computer must have a browser that supports JavaScript and the company must install STARK on their servers. STARK also doesn't need any maintenance or any kind of issues

By implementing STARK the company can greatly increase their customer care service and satisfaction without the need to employ more people and thereby reduce their revenue. Therefore, STARK is very economically feasible to implement.

3.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on

the client. The developed system must have a modest requirement; as only minimal or null changes are required for implementing this system.

STARK only needs a browser that supports JavaScript for its client side implementation. The web application itself is stored and accessed through the company's servers and such doesn't need any new major investment of any kind. On its server side STARK requires Apache Server, PHP 7, MySQL 5 in order to run.

3.4 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

In existing customer care service systems every time, a customer calls the service center a different employee attend the call, who doesn't know that the customer has previously called and what he has mentioned, therefore the customer is forced to explain his entire problem again and again every time he calls. This frustrates the customer thereby creating an unfriendly environment for the customer. Consequently, the customer satisfaction level is greatly reduced.

STARK counters and eradicates this problem by providing a common and experienced technician for thousands of complaints. A single technician will cover multiple complaints by a single user on a particular problem. By this a good relationship is developed between the technician and the customer and makes the entire communication process much more transparent. This provides a friendly environment for the customers to communicate without any difficulties and hesitation. Therefore, STARK is quite socially feasible.

3.5 OPERATIONAL FEASIBILITY

Operational feasibility focuses on the working platform and how easily the technicians can be educated and how quickly they can be trained. This proves that

our application is user friendly and easy to use. Apart from the existing customer care services STARK comparatively has less complications. STARK provides an easy and universal operational platform for the people of diverse languages who fell uncomfortable with the existing system.

On the other hand, STARK has an attractive and user friendly user interface. In order to register a complaint, the user is asked for simple details like his network and mobile number, after which he needs to enter his complaint. The complaint is then processed and a temporary solution appears on the screen. If the problem still persists then he can formally submit the complaint to the ISP. If the complaint is unidentified so far the complaint is still undertaken. Therefore, STARK is very operationally feasible for both customers and the ISP.

3.6 LEGAL FEASIBILITY

Legal feasibility focuses on the possibility of any copyright infringement, legal issues, liabilities, ethical conflicts, policies, terms and conditions. Each ISP has its own terms and condition that has to considered and met by the application so that there won't be any legal conflicts among ISPs in upcoming years. The developers and the ISPs must not have any conflict of interests.

Existing companies along with any start-ups can also merge with our application by accepting our agreements and tie in their legal policies with us. STARK doesn't leak any data about the customers or their ISP as told in the terms and conditions. Even though STARK increases the transparency between the ISP and their customers, it still provides security and protection through segmentation.

CHAPTER 4

SYSTEM ANALYSIS

4.1 INTRODUCTION

This chapter focuses on the analysing the design, strengths, processes, requirements and modules used in STARK. During this phase the requirements of STARK were categorized into three different categories – user requirements, functional requirements and non-functional requirements. The modules contained in this system are listed and described along with all the features of STARK.

4.2 SYSTEM MODULE DESCRIPTION

i. Pattern Detection:

- When the user enters any complaints, a process called pattern matching takes place. The input provided by the user is taken and searched for specific tag words.
- If the tag words are found, then a particular pattern is identified from their order.
- Thus, this web application has a total of 45 tag words which can be identified from the complaint.
- This web application has a total of 49 patterns that can be recognized form the complaint that the user provides.

ii. Pattern Matching:

- This module executes after the pattern detection module.
- The result from the previous module is considered as input for this module.
- Thus, in this application we have a message database which has a total of seven message sets among which any one would be displayed based on the pattern matched.

iii. Message Retrieval:

- Once the Pattern Matching is done, a message set corresponding to the complaint posted by the customer will be fetched and displayed to them.
- This message set will contain some common and easy fixes that may provide temporary or even a complete fix for the customer's problem.
- If the problem still persists, the customer can then officially register the complaint to the ISP's customer care service.

iv. Word Cloud Generation:

- This module is used to generate a Word Cloud for complaints posted by customers.
- The complaints can be filtered based on two criteria which are – location and date.
- Only administrators can access and use this module.

v. Manage Technician:

- This module is used for handling the accounts of technicians.
- Accounts for new technicians can be created using this module.
- Administrators can add, modify and view all technicians' accounts.
- Only administrators can access and use this module.

4.3 REQUIREMENTS SPECIFICATION

After analysing the data collected, we formulated a number of requirements namely user requirement, system hardware software attribute. These were grouped as user, functional, non-functional and systems requirements.

4.3.1 User Requirements

During data collection, we investigated and found out how the current system operates, not only that but also tried out which problems are faced and how best they can be settled. The users described some of the basic requirements of the system this includes storing multiple phone numbers, carriers/ISP, transparency, direct communication, quick response.

4.3.2 Functional Requirements

- Administrator registration.
- Technician registration.
- Customer registration.
- Complaint posting.
- Complaint viewing.
- Filtering of complaints based on criteria.
- Generation of Word Cloud based on complaints.
- Customer complaint history.
- Pre-defined temporary solutions.

4.3.3 Non-Functional Requirements

The system must authenticate and store data provided by the customer and must notify the administrators if there are any unauthorized changes. The system must provide the administrators with complete access and rights so that they can add, view, modify any account. The system must be reliable, fault tolerant and have redundancy schemes to ensure that no data is unintentionally lost.

4.4 Features of the Proposed System

- Narrow down millions of complaints to a few major ones that can be focused for solving.
- Multiple technicians can login into their own dashboard.
- Immediate solution for the complaint posted by a user is provided as a temporary rectification.
- Enormous amount can be saved in narrowing down the complaints to major ones.
- Criteria based word cloud generation can be made from any technician's dashboard.
- Separate login for each user of the network is provided.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 INTRODUCTION

This chapter includes the SRS documentation of STARK, project plan and organization, hardware requirements, software requirements, user requirements, system components and a network and server requirements for STARK.

5.2 PROCESS LOGIC

1. Registration Process

The user, ISP administrator and ISP technicians will be able to register themselves to STARK using this registration process. Here the above-mentioned entities would have to provide the required details such as, name, phone number, address, ISP and email ID in order to create an account for them in STARK.

2. Login Process

In order to access their accounts and utilize the features provided by STARK, everyone must login using their login details which they provided during their registration. Unauthorized logins would be identified and rejected in the login process.

3. Complaint Entry Process

After logging into the user's dashboard, the user must select their network and phone number for which they are going to make a complaint, after which they would be able to enter their complaint using the complaint entry process. The complaint would be transferred to the solution provider process.

4. Solution Provider Process

This process receives the complaint provided by the user and analyzes the words contained in it in order to obtain a pattern which would then be matched with the solution database that has been previously entered and the best result would be shown to the user.

5. Technician Account Registration Process

The administrator has to create an account for the technicians, which includes their details such as name, phone number and address. After registration,

the administrator would then be able to tack the technician's location and their status.

6. Word Cloud Process

This process can be accessed only by the administrator of a particular ISP. By analysing thousands of complaints using word cloud algorithm, this process will generate a word cloud chart based on criteria such as, date and location.

7. Complaint Handling Process

The technicians can view all the complaints made by the users in his particular area. He can then choose a specific complaint to handle on his own. Once a complaint has been chosen the technician's details will be sent to the customer. After the complaint has been processed the technician will then update the customer about its status.

5.3 PROCESS PLAN AND DECOMPOSITION

5.3.1 PERT CHART

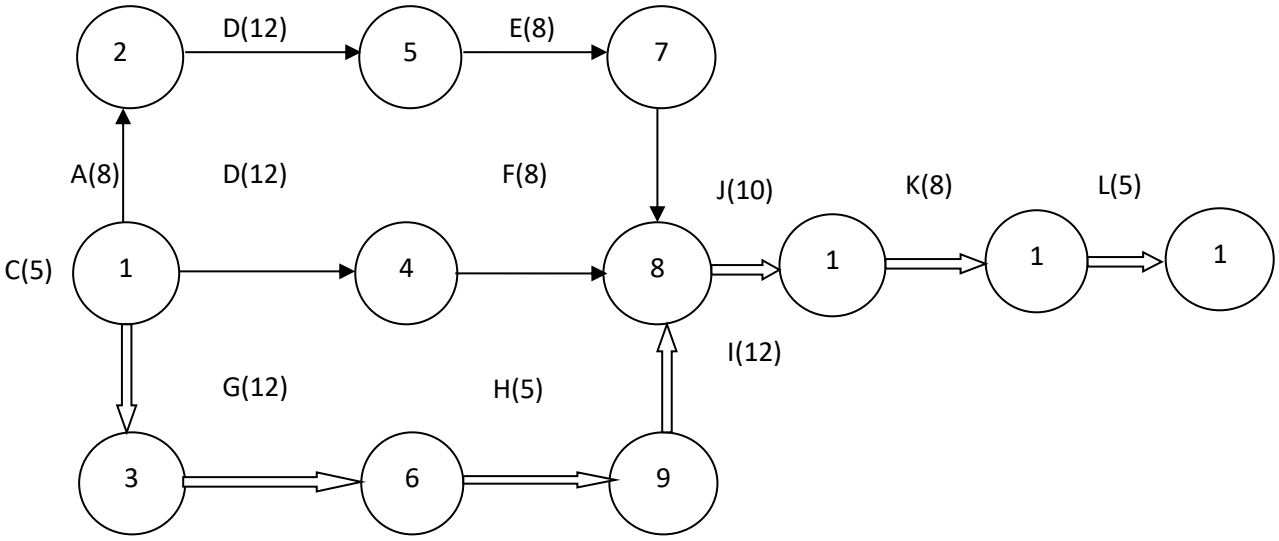


FIGURE 5.3.1

A=Order computing Platform

B=Prepare site

C=Review Specification

D=Install equipment

E= Test hardware

F=Training

G=Write Programs

H=Test programs

I=Test software

J=Convent Systems

K=Implement follow-up

L= Accept

5.3.2 GANTT CHART

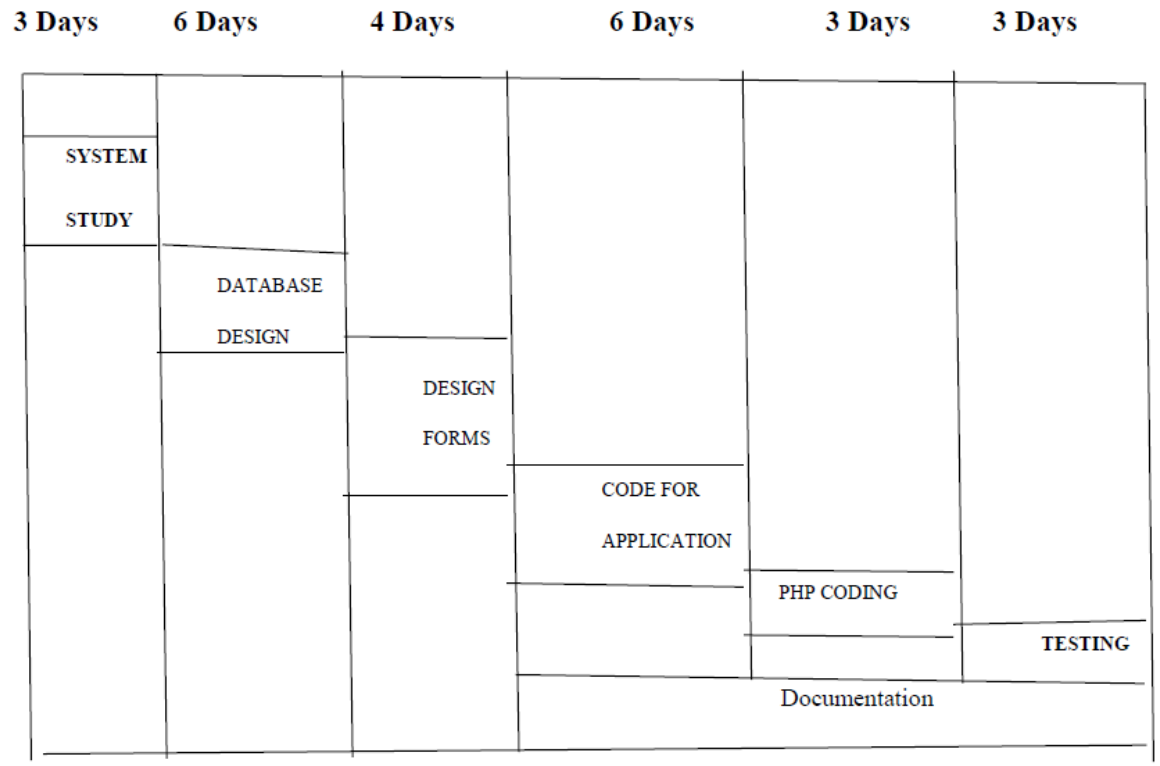


FIGURE 5.3.2

5.4 HARDWARE REQUIREMENTS

Hardware	Minimum System Requirements
Processor	Above 2.4Ghz processor speed
Memory	512 MB RAM
Server Disk Space	Minimum 2TB
Display	800*600 Resolution

TABLE 5.4.1

5.5 SOFTWARE REQUIREMENTS

Operating System	Any OS (Windows/Mac OS/Linux)
Browser	Script supported browser (Internet Explorer, Edge, Chrome, Safari, Opera, Firefox, UC Browser, etc.)

TABLE 5.5.1

CHAPTER 6

SYSTEM ARCHITECTURE AND TECHNOLOGY PLATFORM

6.1 INTRODUCTION

This gives a high-level view of the new system with the main components of the system and the services they provide and how they communicate. The system is implemented using a two-tier architecture that comprises of user interface and DBMS.

6.2 CONTEXT LEVEL DIAGRAM

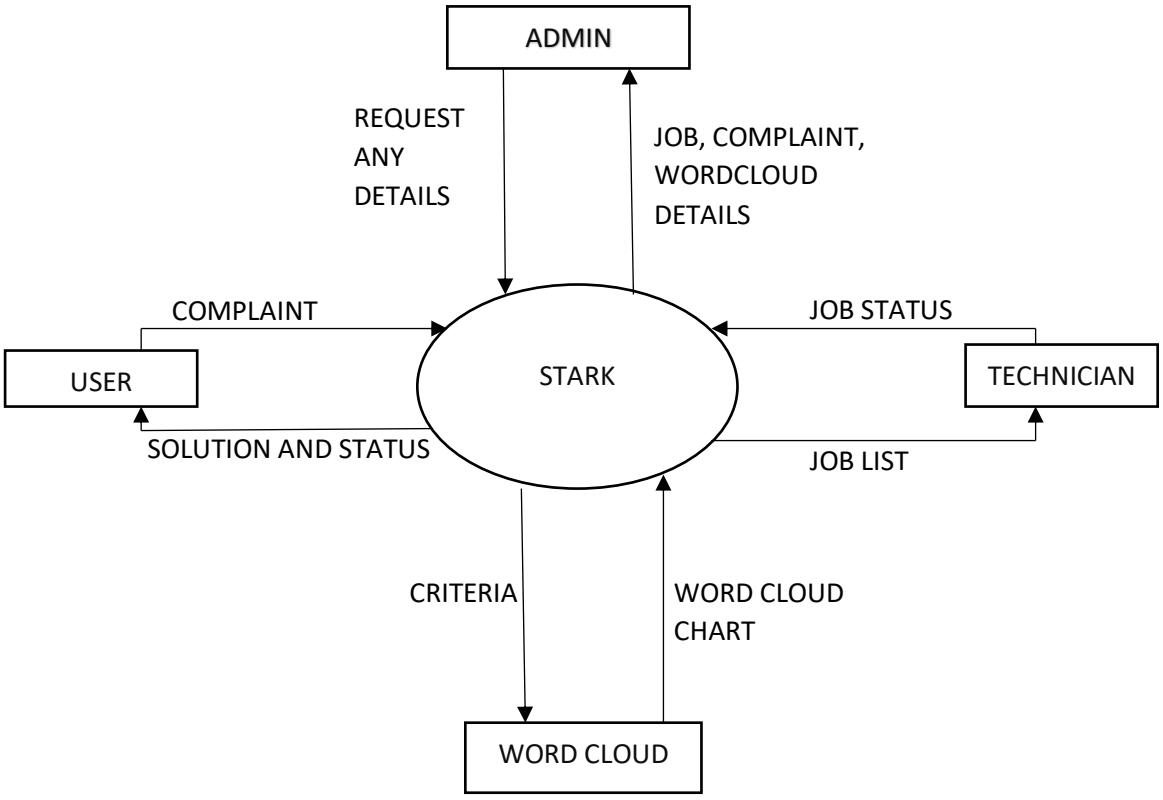


FIGURE 6.2.1

6.3 TECHNOLOGY PLATFORM

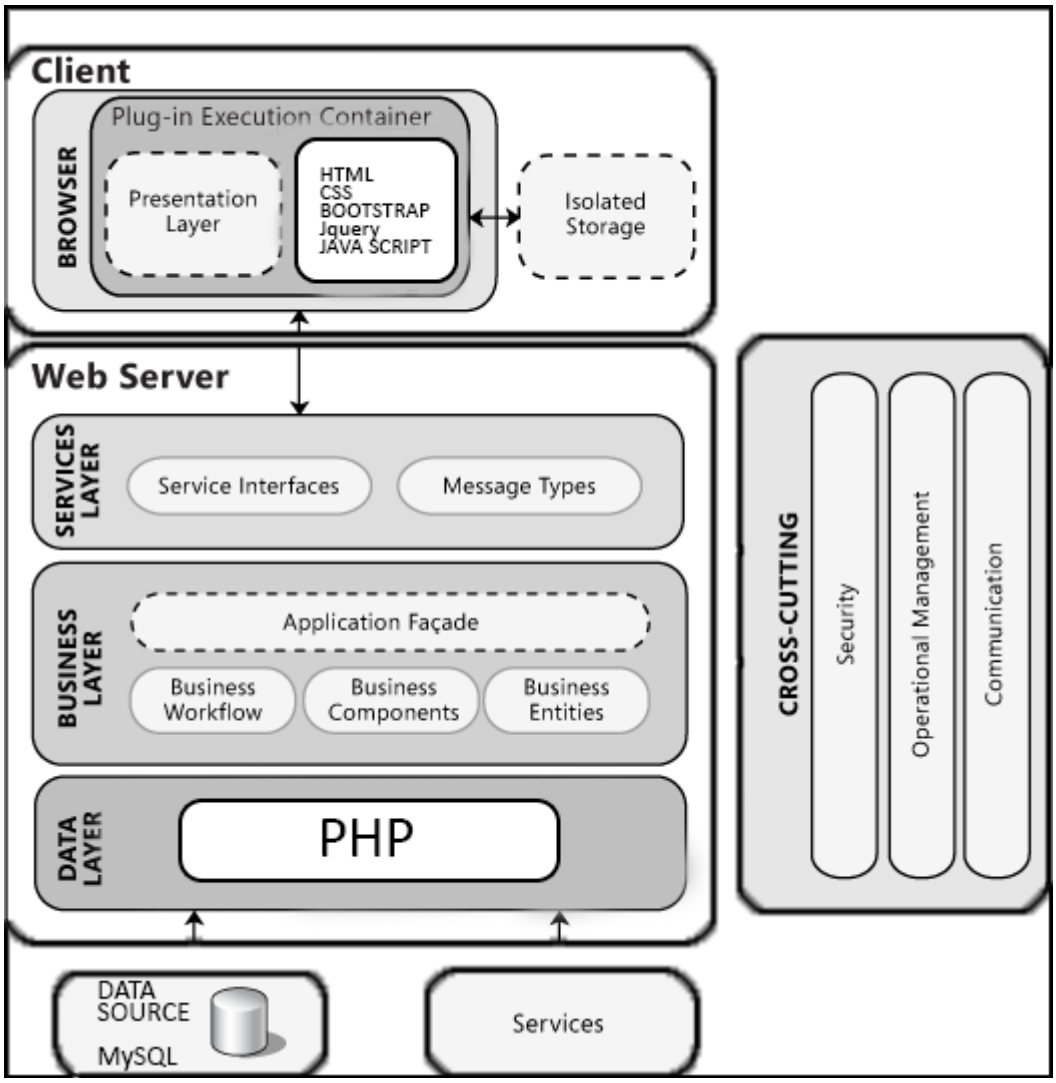


FIGURE 6.3.1

I.PRESENTATION LAYER

PRESENTATION LAYER COMPONENTS

• **User interface (UI) components**.

User interface components provide a way for users to interact with the application. They

render and format data for users. They also acquire and validate data input by the user.

• **User process components**

User process components synchronize and orchestrate user interactions. Separate user

process components may be useful if you have a complicated user interface. Implementing

common user interaction patterns as separate user process components allows you to reuse them in multiple user interfaces.

PRSENTATION LAYER'S TOOLS:

i. HTML

Hypertext Markup Language is the standard markup language used to **create web pages.** Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation making it a markup language rather than a programming language.

ii. CSS

Cascading Style Sheets is a style sheet language used for **describing the look and formatting of a document written in a markup language.** CSS is designed primarily to enable the separation of document content from document presentation. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging websites. Bootstrap Framework is used to provide CSS classes.

iii. JavaScript

JavaScript is a Dynamic Programming language. It is most commonly used as part of web browsers, whose implementations **allow client-side scripts to interact with the user,** control the browser, communicate asynchronously, and alter document content that is displayed

II. SERVICE LAYER

SERVICE LAYER COMPONENTS

Service Interfaces

Services expose a service interface to which all inbound messages are sent. The definition of the set of messages that must be exchanged with a service, in order for the service to perform a specific business task, constitutes a contract. You can think of a service interface as a façade that exposes the business logic implemented in the service to potential consumers.

Message Types

When exchanging data across the service layer, data structures are wrapped by message structures that support different types of operations. For example, you might have a Command message, a Document message, or another type of message. These message types are the "message contracts" for communication between service consumers and providers.

III. BUSINESS LAYER

BUSINESS LAYER COMPONENTS

Application Façade

An application façade combines multiple business operations into single message based operation. You might access the application façade from the presentation layer using different communication technologies.

Business components

After a user process collects the data it requires, the data can be operated on using business rules. The rules will describe how the data should be manipulated and transformed as dictated by the business itself. The rules may be simple or complex, depending on the business itself. The rules can be updated as the business requirements evolve.

Business entity components

Business entities are used to pass data between components. The data represents real-world business entities, such as products and orders. The business entities that the application uses internally are usually data structures such as Datasets, Extensible Markup Language (XML) streams. Alternatively, they can be implemented using custom object-oriented classes that represent the real-world entities your application has to work with, such as a product or an order.

Business workflow

Many business processes involve multiple steps that must be performed in the correct order and orchestrated. Business workflows define and coordinate long-running, multi-step business processes, and can be implemented using business process management tools.

IV. DATA LAYER

DATA LAYER COMPONENTS

Data access logic components

Data access components abstract the logic necessary to access your underlying data stores. Doing so centralizes the data access functionality, which makes the application easier to configure and maintain.

Data Helpers / Utilities

Helper functions and utilities assist in data manipulation, data transformation, and data access within the layer. They consist of specialized libraries and/or custom routines especially designed to maximize data access performance and reduce the development requirements of the logic components and the service agent parts of the layer.

Service agents

When a business component must use functionality exposed by an external service, you may need to create code that manages the semantics of communicating with that service. Service agents isolate your application from the idiosyncrasies of calling diverse services, and can provide additional services such as basic mapping between the format of the data exposed by the service and the format your application requires.

DATA LAYER TOOLS

I. MySQL

MySQL is a relational DBMS and the widely used open source RDBMS. It is a popular choice of **database for use in web applications** and is a central component of the widely used LAMP open source web application Software Stack.

II. PHP

PHP is a server side scripting language designed for development but also used as general purpose programming language. PHP code in combination with HTML code is used in templating engines and web frameworks. After the PHP code is executed the **web server sends the resulting output to its client usually in the form of webpage.**

CHAPTER 7
DATABASE DESIGN

7.1 INTRODUCTION

This chapter deals with the tables that are made for this particular web applications. This chapter covers the details of how the tables are linked with each other and what are all the data that are being stored in the table. This chapter also show the relationship between the tables and what data each cell carries.

7.2 DATABASE ER DIAGRAM

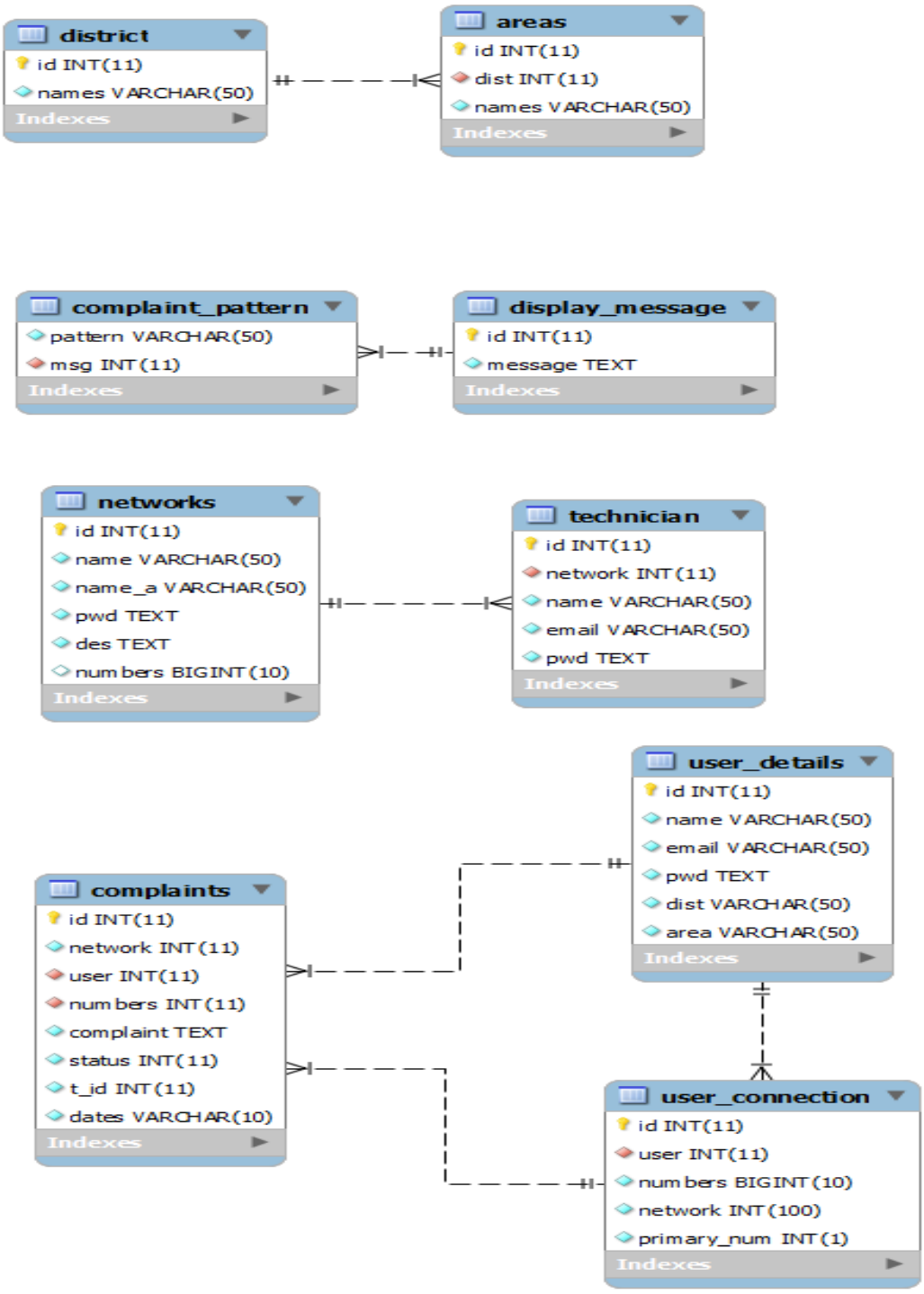


FIGURE 7.2.1

7.3 DATA DICTIONARY

USER DETAILS TABLE

ID	Int(11)	Primary Key
Name	Varchar(50)	
Email	Varchar(50)	
Pwd	TEXT	
Dist	Varchar(50)	
Area	Varchar(50)	

TABLE 7.3.1

USER CONNECTION DETAILS

Id	Int(11)	Primary Key
User	Int(11)	Foreign Key
Numbers	BigInt(10)	
Network	Int(11)	Foreign Key
Primary_num	Int(1)	

TABLE 7.3.2

NETWORKS TABLE

Id	Int(11)	Primary Key
Name	Varchar(50)	
Name_a	Varchar(50)	
Pwd	TEXT	
Des	TEXT	
Numbers	BigInt(10)	

TABLE 7.3.3

TECHINICIAN TABLE

Id	Int(11)	Primary Key
Network	Int(11)	Foreign Key
Name	Varchar(50)	
Email	Varchar(50)	
Pwd	TEXT	

TABLE 7.3.4

COMPLAINTS TABLE

ID	Int(11)	Primary Key
Network	Int(11)	Foreign Key
User	Int(11)	Foreign Key
Numbers	Int(11)	Foreign Key
Complaints	TEXT	
Status	Int(1)	
T_id	Int(11)	
Dates	Varchar(10)	

TABLE 7.3.5

COMPLAINT PATTERN TABLE

Pattern	Varchar(50)	
msg	Int(11)	Foreign Key

TABLE 7.3.6

DISPLAY MESSAGE TABLE

Id	Int(11)	Primary Key
Message	TEXT	

TABLE 7.3.7

DISTRICT TABLE

Id	Int(11)	Primary Key
Names	Varchar(50)	

TABLE 7.3.8

AREAS TABLE

Id	Int(11)	Primary Key
Dist	Int(11)	Foreign Key
Names	Varchar(50)	

TABLE 7.3.9

CHAPTER 8
DATA FLOW DIAGRAM

8.1 INTRODUCTION

The way in which the data provided by the users flow in the application developed can be graphically represented using the data flow diagram. The data flow diagram shows various components and elements of the application system in order to show how these elements and components utilize the data that are being provided by the user of the application.

8.2 DFD LEVEL 0

ADMIN

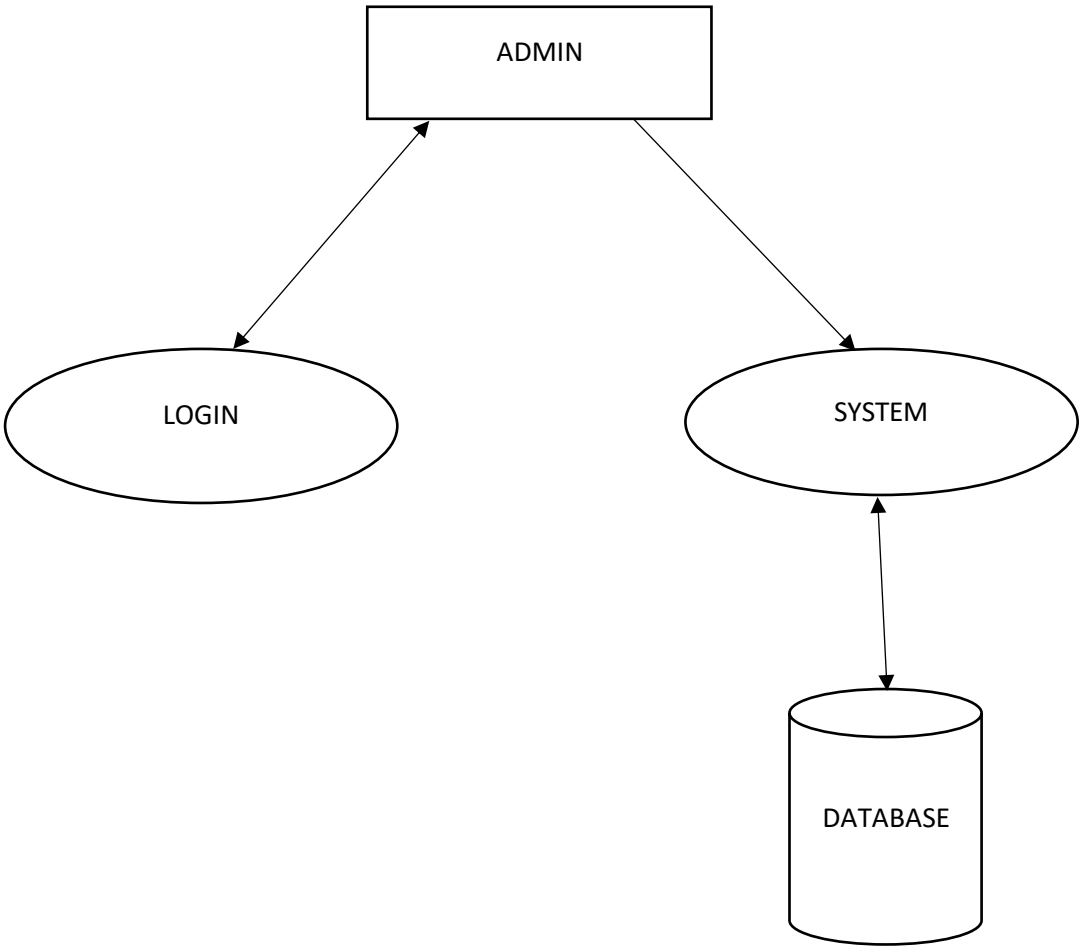


FIGURE 8.2.1

USER

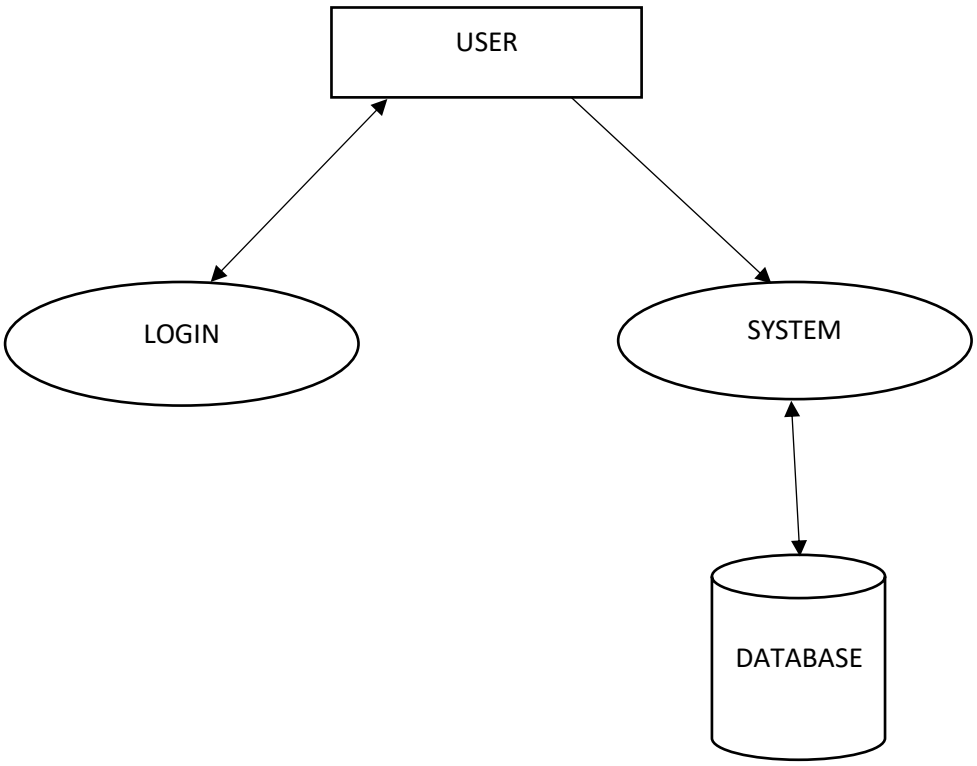


FIGURE 8.2.2

TECHNICIAN

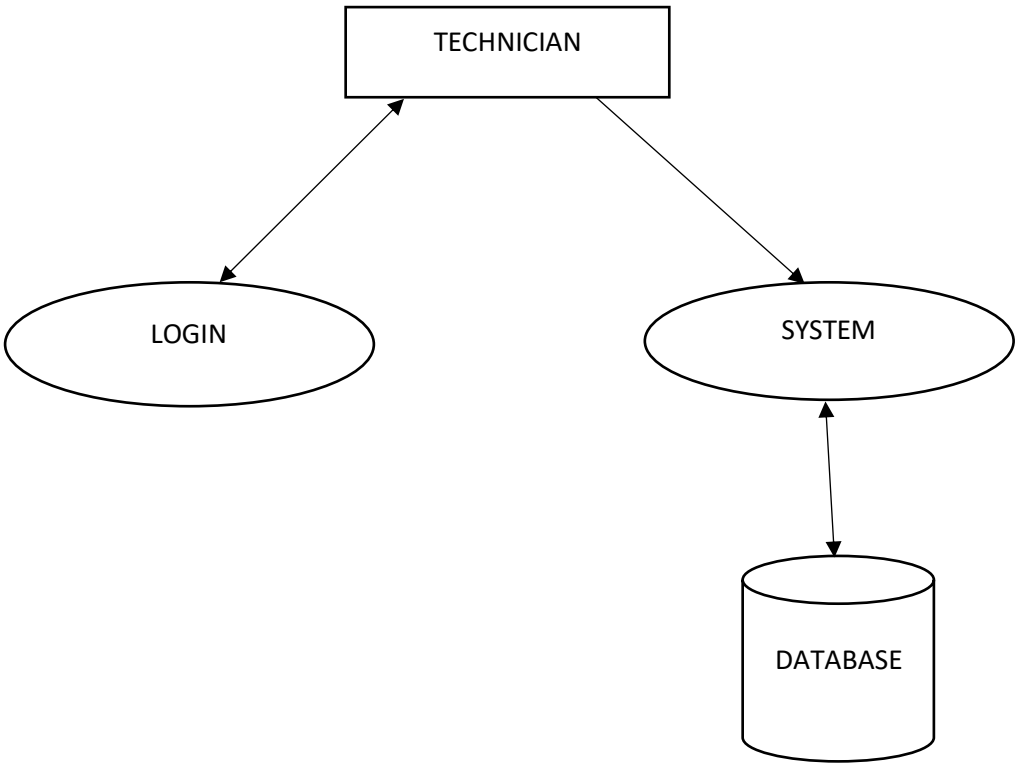


FIGURE 8.2.3

8.3 DFD LEVEL 1

ADMIN

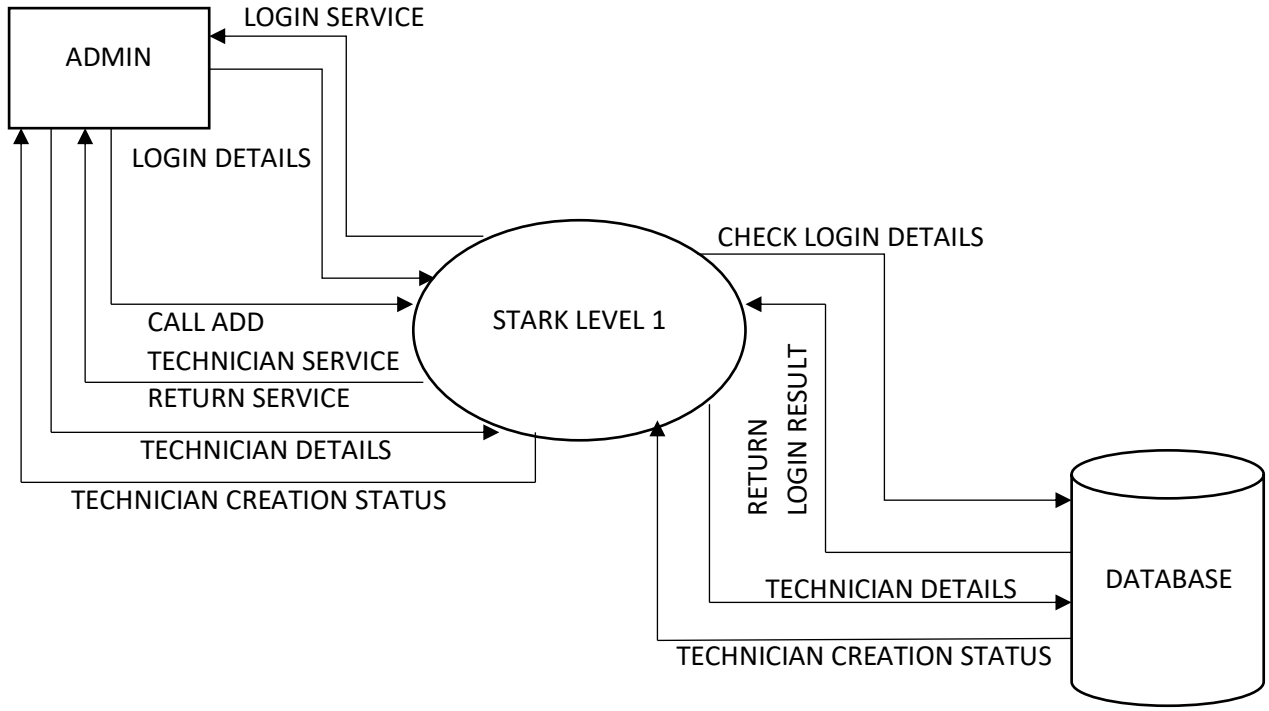


FIGURE 8.3.1

USER

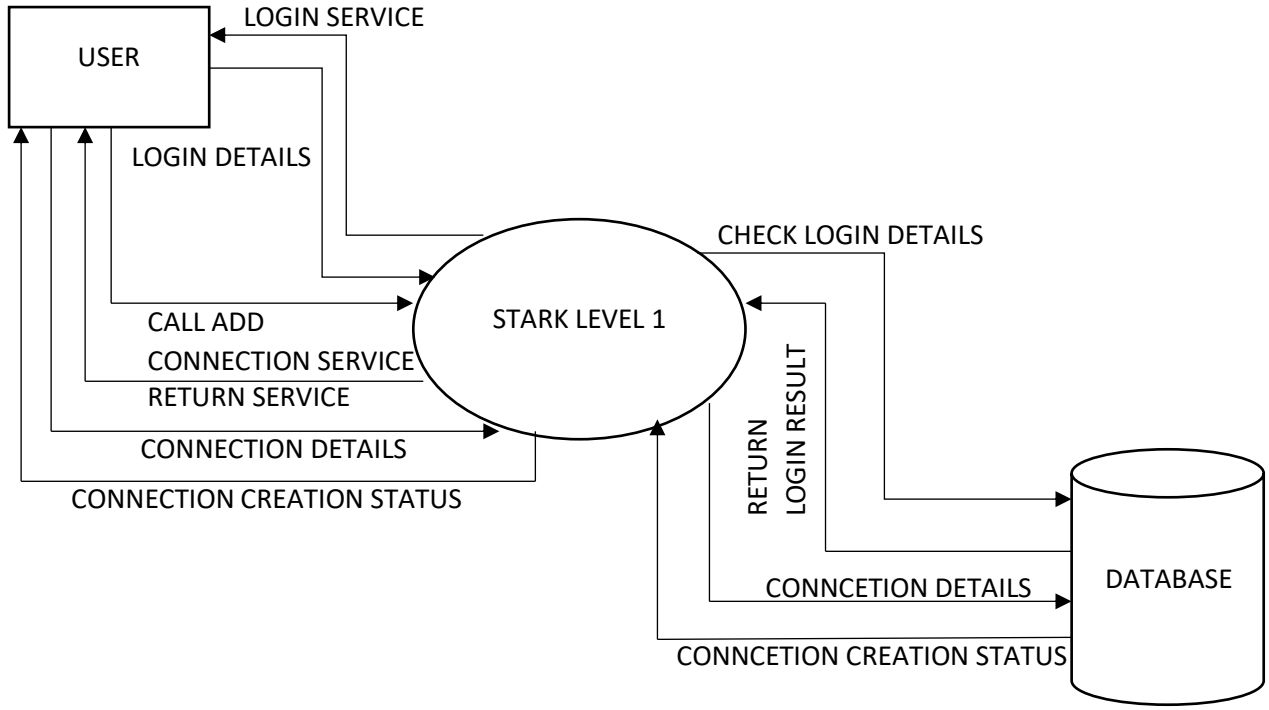


FIGURE 8.3.2

TECHNICIAN

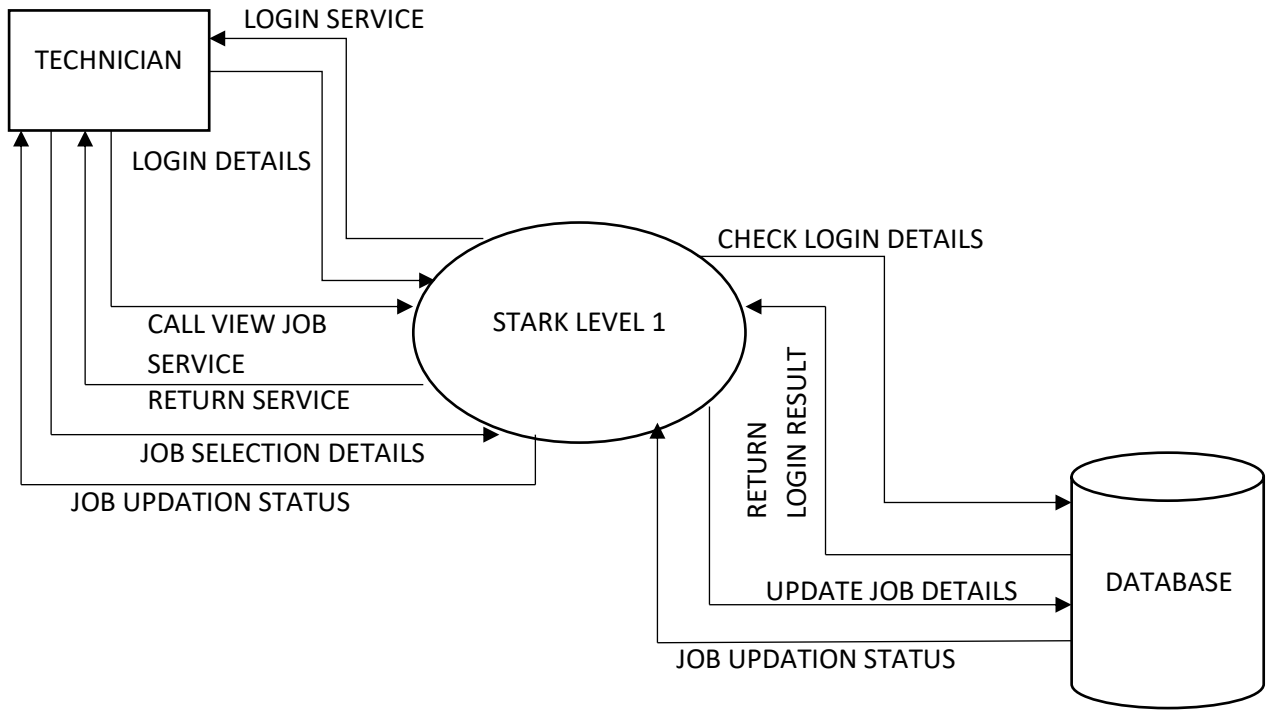


FIGURE 8.3.3

8.4 DFD LEVEL 2

ADMIN

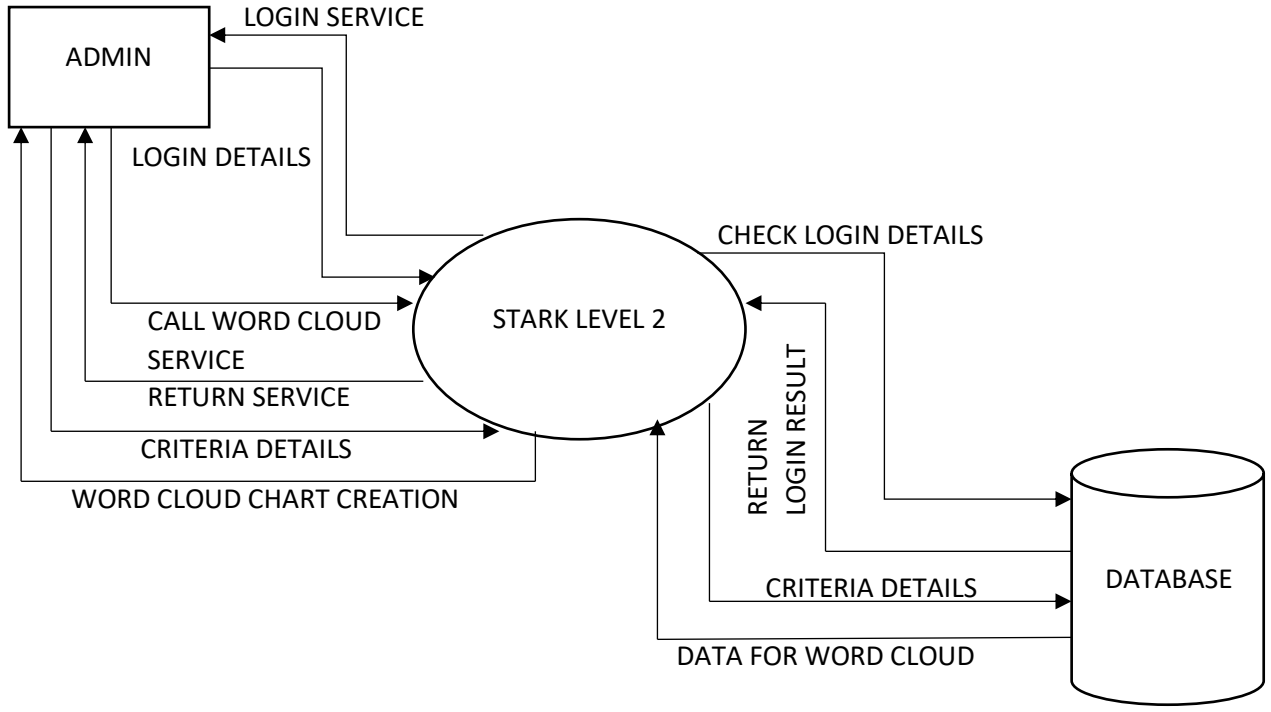


FIGURE 8.4.1

USER

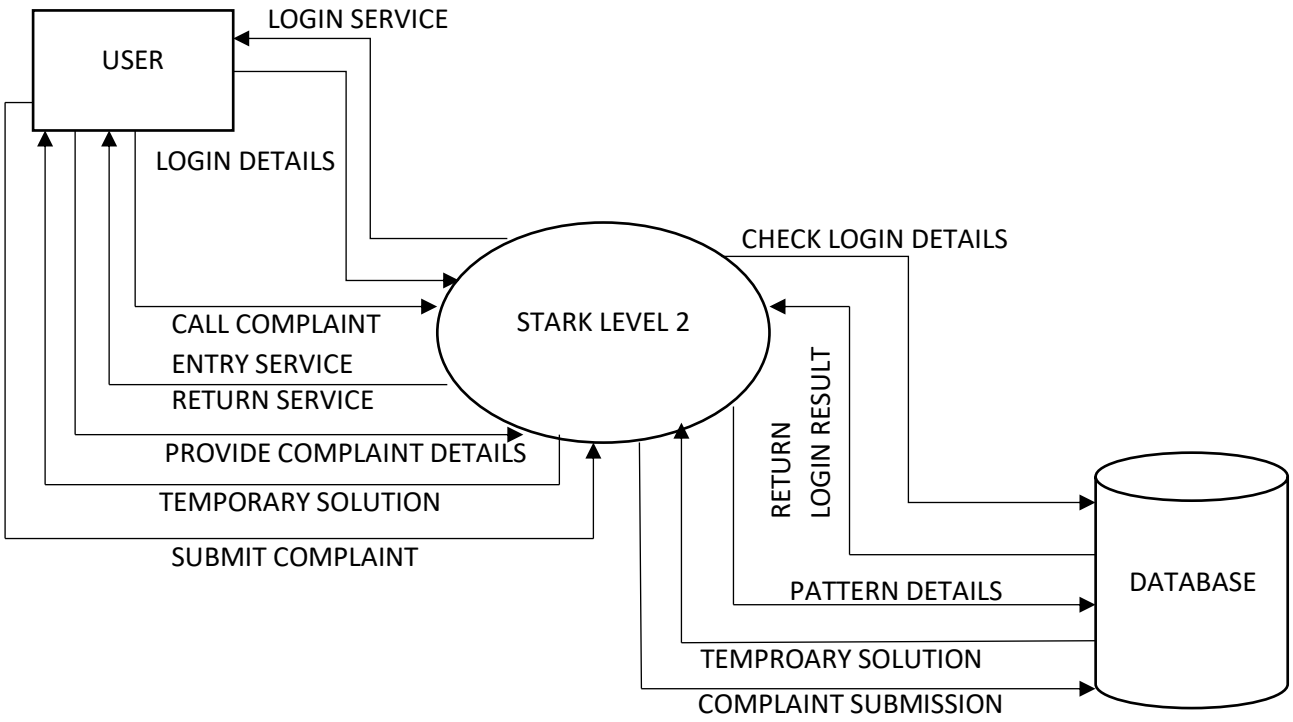


FIGURE 8.4.2

TECHNICIAN

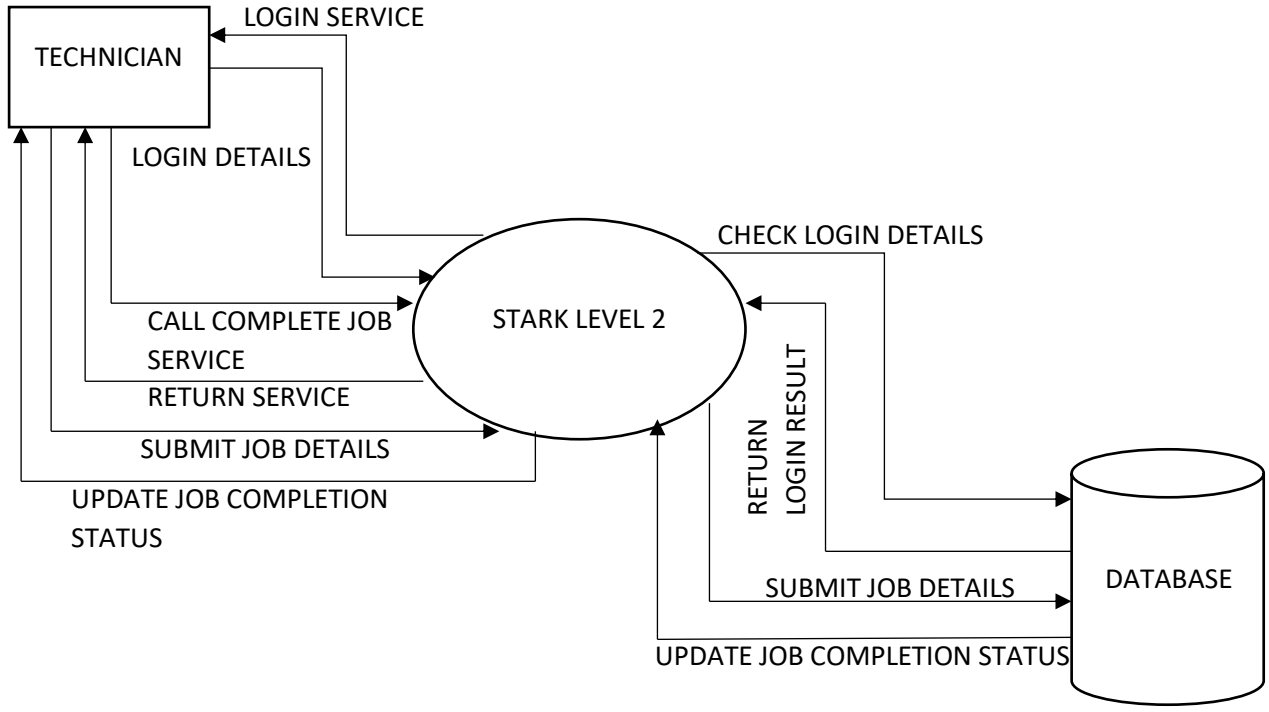


FIGURE 8.4.3

CHAPTER 9

SECURITY MEASURES

9.1 ROLES AND RIGHTS

ADMINISTRATOR

- Administrator has the highest authority and access in STARK.
- It includes the administrator functions of the system.
- It can add, create, modify and remove technician's accounts and details.
- It can view all the complaints made by users pertaining to their ISP.
- It can generate Word Cloud for complaints based on criteria such as location and date.
- It can communicate directly to the customers as well as technicians.

TECHNICIAN

- Technicians can view all the complaints made by the customers in his particular area.
- Technician can choose and solve complaints of his own choice and priority.
- On selecting a complaint, the customer and the administrator will be notified on the activity and status of the complaint the technician has chosen.
- Technicians can directly communicate with the customer and update the technician directly in the event any changes occur in the complaint.

CUSTOMER

- Customers can post their complaints to STARK after the process of logging into their account.
- Customers can add new phone numbers and associated carriers.
- Customer will be provided with the details of the technician after his complaint has been selected.
- In case of any updates the customers can directly communicate with their associated technician.
- Customers are able to communicate directly with the admin after special request has been made.

9.2 SECURITY ALGORITHM

- Every account in STARK is protected by a username and password.
- When creating an account for the first time the customers have to verify their email ID and phone number.
- STARK uses MD5 algorithm to provide authentication to its users.
  - ❖ It stands for Message Digest 5 and it is the best algorithm of its kind.
  - ❖ MD5 algorithm takes in a variable sized input and generates a 256-bit message digest.
  - ❖ The message digest will be generated during login based on the user's credentials which will then be compared with the message digest generated during the registration process that is stored in STARK database.

CHAPTER 10

SYSTEM TESTING

10.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement

10.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Test Results

All the units of coding available with the applications worked successfully on the independent way of execution.

10.3 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10.4 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centred on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.

Functions	: identified functions must be exercised.
Output	: identified classes of application must be exercised
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Test Result: The functions that are available within the source coding are being functionally tested individually and no errors were found.

10.5 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Test Result: The configuration level of testing is done with the source coding and thus no errors are being found.

10.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10.7 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10.8 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 11
CONCLUSION

11.1 OUTCOMES

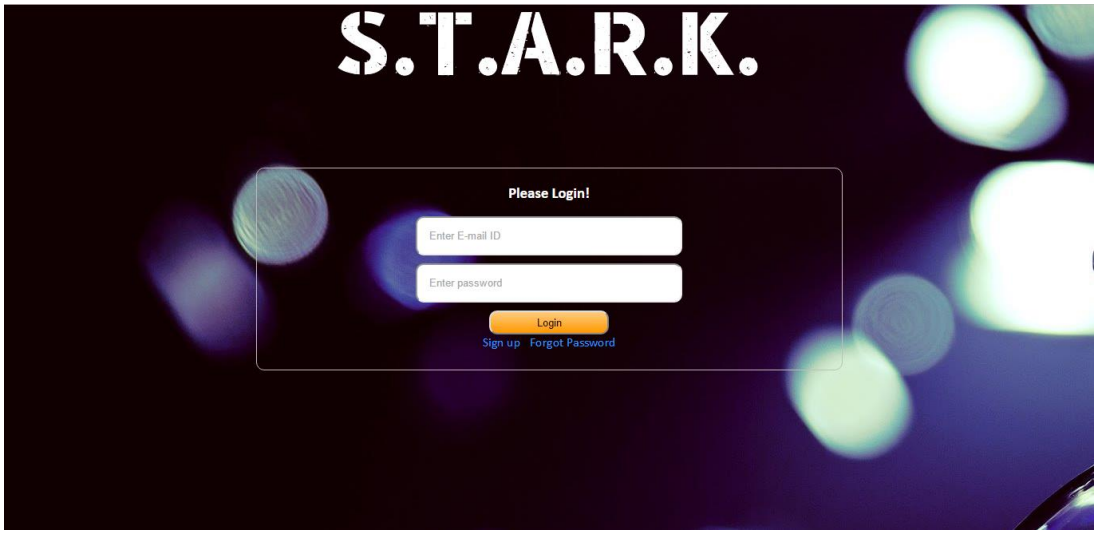
- Customer satisfaction will be greatly increased throughout the industry.
- ISPs won't have to spend as much money as training their customer care employees as they do currently.
- The work of the employees will be reduced significantly because of using STARK.
- About 65-95% of the complaints, analysed using STARK Word Cloud algorithm, are solved.
- The most significant problems faced by the customers are easily identified and are solved.

11.2 FUTURE ENHANCEMENTS

- STARK can be adapted to suit any industry.
- Offers and discounts provided by ISP can be directly advertised to customers through STARK.
- Google Maps functionality can be added so that the technician can easily identify the customer's location.
- Customers can rate the service provided by the technicians based on their personal satisfaction.
- Based on the rating provided by the customers the technician will be provided points which will be taken into consideration by the administrator.

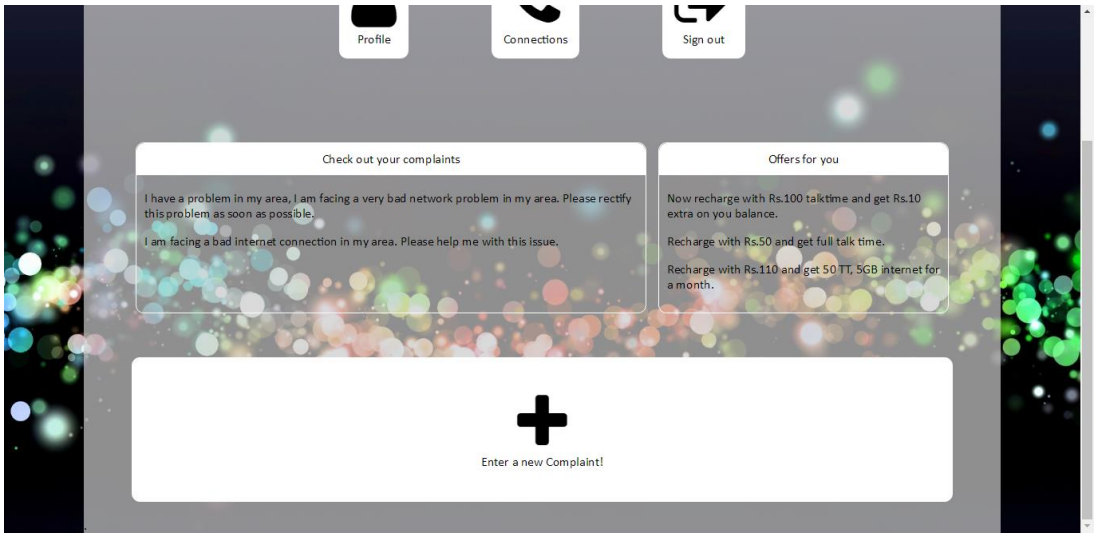
APPENDIX A
SCREEN SHOTS

LOGIN PAGE

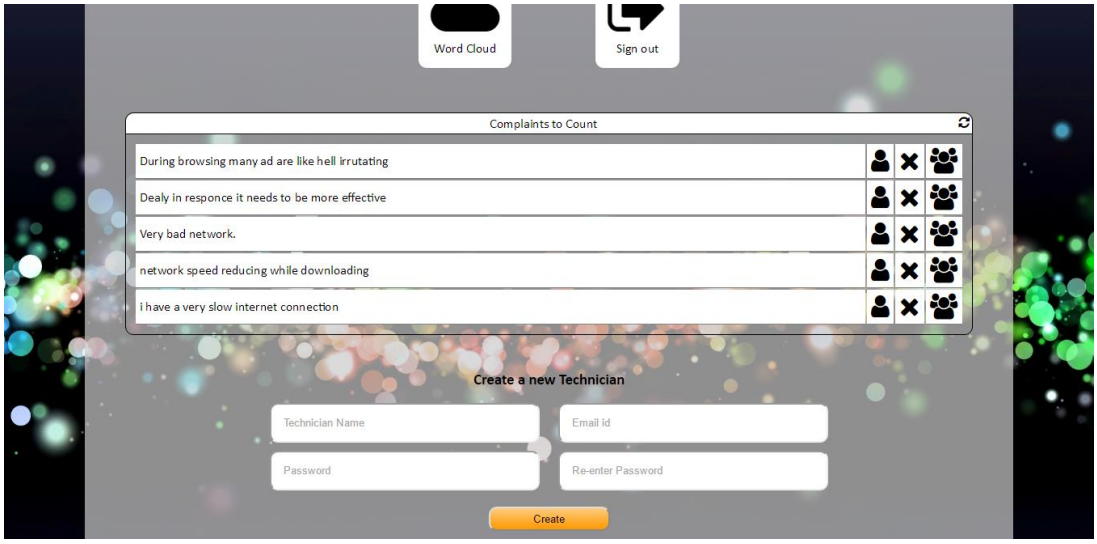


SIGN UP

USER DASHBOARD

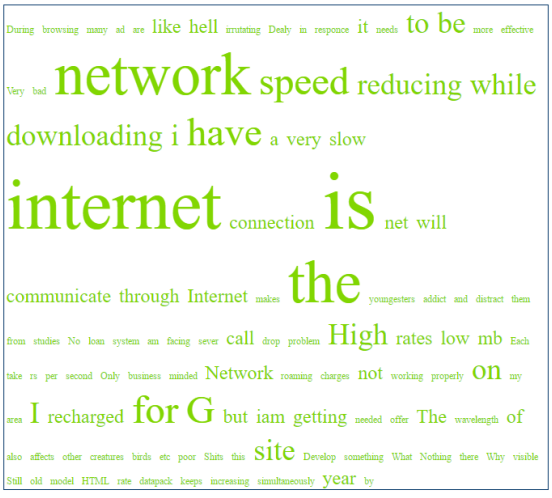


ADMIN DASHBOARD

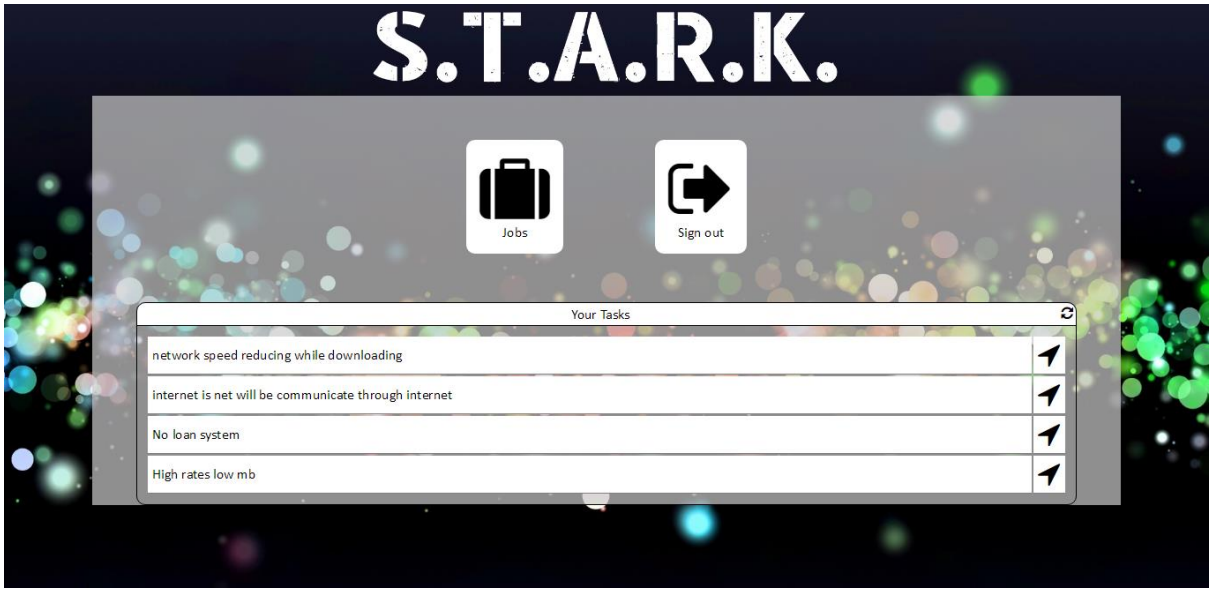


WORD CLOUD GENERATOR

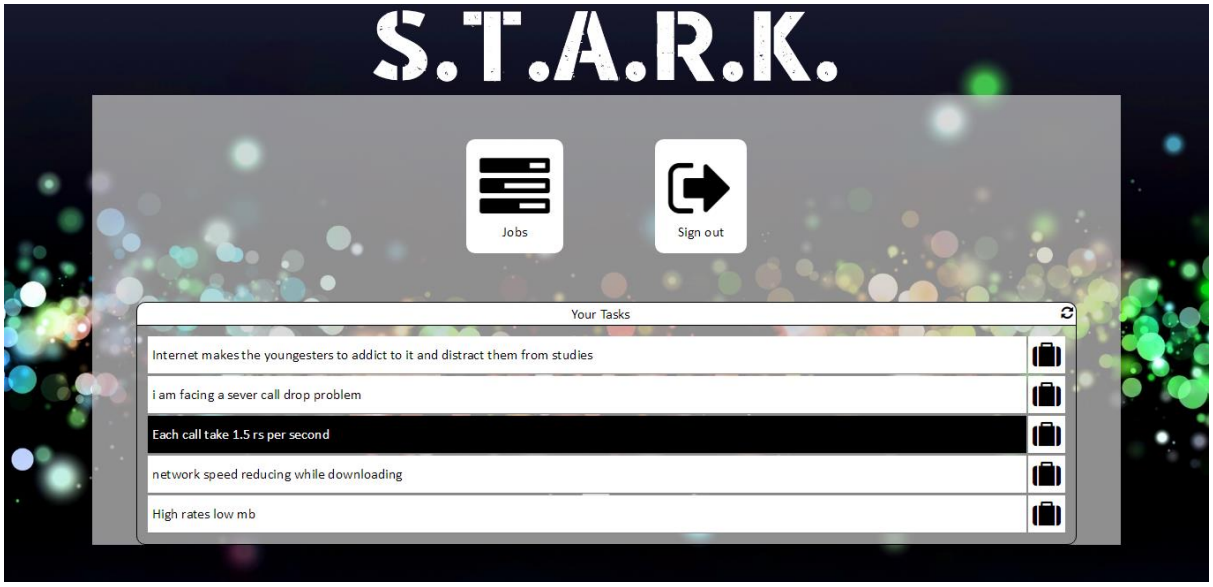
WORD CLOUD RESULT CHART



TECHNICIAN DASHBOARD



TECHNICIAN JOB SELECTION



APPENDIX B
SOURCE CODING

HTML AND PHP

I. WORD CLOUD GENERATOR

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
	<title>STARK - word cloud generator</title>
	<style type="text/css" media="screen">
		/*<![CDATA[*/
		.tag_cloud { padding: 3px; text-decoration: none; }
		.tag_cloud:link  { color: #81d601; }
		.tag_cloud:visited { color: #019c05; }
		.tag_cloud:hover { color: #ffffff; background: #69da03; }
		.tag_cloud:active { color: #ffffff; background: #ACFC65; }
		.wordcloud {
			border: 1px solid #036;
			margin: 0.5in auto;
			padding: 0;
			page-break-after: always;
			page-break-inside: avoid;
			width: 7in;
		}
		/*]]>*/
	</style>
</head>
<?php
 include 'reloga.php';
 include 'db_connection.php';
$freqData = array();
$lorem ="";
if((isset($_GET['q'])) && ($_GET['q'] == "ys")){
	$d = strtotime("Yesterday");
	$query = "SELECT complaint FROM complaints WHERE dates =
'".date("d/m/Y",$d)."' AND network = ".$_SESSION['ida'];
	$result = mysqli_query($con,$query);
	if(mysqli_num_rows($result) > 0){
		while ($row = mysqli_fetch_assoc($result)) {
			$lorem = $lorem." ".$row['complaint'];
		}
	}
	else {
		header("Location:word_cloud.php?t=not&s=".date("d/m/Y",$d));
	}
}
```

```
elseif((isset($_GET['q'])) && ($_GET['q'] == "lm")){
	$d = strtotime("Last Month");
	$query = "SELECT complaint FROM complaints WHERE dates LIKE
'%".date("m/Y",$d)."' AND network = ".$_SESSION['ida'];
	$result = mysqli_query($con,$query);
	if(mysqli_num_rows($result) > 0){
		while ($row = mysqli_fetch_assoc($result)) {
			$lorem = $lorem." ".$row['complaint'];
		}
	}
	else {
		header("Location:word_cloud.php?t=not");
	}
}
elseif((isset($_POST['ps'])) && ($_POST['ps'] == "byloc")){
	$area = $_POST['area'];
	$query = "SELECT c.complaint FROM complaints c, user_details u WHERE u.area =
".$area." AND c.user = u.id AND network = ".$_SESSION['ida'];
	$result = mysqli_query($con,$query);
	if(mysqli_num_rows($result) > 0){
		while ($row = mysqli_fetch_assoc($result)) {
			$lorem = $lorem." ".$row['complaint'];
		}
	}
	else {
		header("Location:word_cloud.php?t=not");
	}
}
elseif ((isset($_POST['pd'])) && ($_POST['pd'] == "bydate")) {
	$da = $_POST['d'];
	$mn = $_POST['m'];
	$yy = $_POST['y'];
	$dates = $da."/".$mn."/".$yy;
	$query = "SELECT complaint FROM complaints WHERE dates = '".$dates."' AND
network = ".$_SESSION['ida'];
	$result = mysqli_query($con,$query);
	if(mysqli_num_rows($result) > 0){
		while ($row = mysqli_fetch_assoc($result)) {
			$lorem = $lorem." ".$row['complaint'];
		}
	}
	else {
		header("Location:word_cloud.php?t=not");
	}
```

```
}
foreach( str_word_count( $lorem, 1 ) as $word )
{
	array_key_exists( $word, $freqData ) ? $freqData[ $word ]++ : $freqData[ $word ] =
0;
}
function getCloud( $data = array(), $minFontSize = 12, $maxFontSize = 100 )
{
	$minimumCount = min( array_values( $data ) );
	$maximumCount = max( array_values( $data ) );
	$spread       = $maximumCount - $minimumCount;
	$cloudHTML    = '';
	$cloudTags    = array();

	$spread == 0 && $spread = 1;

	foreach( $data as $tag => $count )
	{
		$size = $minFontSize + ( $count - $minimumCount )
			* ( $maxFontSize - $minFontSize ) / $spread;
		$cloudTags[] = '<a style="font-size: ' . floor( $size ) . 'px'
		. '" class="tag_cloud" href="http://www.google.com/search?q=' . $tag
		. '" title="\" . $tag  . '\' returned a count of ' . $count . '">'
		. htmlspecialchars( stripslashes( $tag ) ) . '</a>';
	}

	return join( "\n", $cloudTags ) . "\n";
}
?>
<body>
	<div class="wordcloud" id="wrapper">
		<?php
		echo getCloud( $freqData );
		 ?>
	</div>
</body>
</html>
```

II. PATTERN IDENTIFICATION

```
<?php
	function wordAnalyser($word){
		$matchWord = array("connection", "no", "slow", "signal", "call", "calls",
"internet", "functioning", "not", "available", "low", "bad", "unavailable", "allowed",
"interrupted", "disconnected", "disconnecting", "disconnection", "working", "poor", "proper",
```

```
"sucks", "line", "cut", "tower", "problematic", "reception", "problem", "network", "connect",
"available", "breaking", "breakage", "non", "functional", "do", "dont", "dropping", "dropped",
"drop", "failed", "failure", "fails", "received", "receive");
                for($i = 0; $i<sizeof($matchWord); $i++){
                        if($word == $matchWord[$i]){
                                return $i+1;
                        }
                }
                return "notavail";
        }

        function getPattern($words){
                $pattern = array();
                for($i = 0; $i < sizeof($words); $i++){
                        $val = wordAnalyser($words[$i]);
                        if(($val != "notavail") ){
                                array_push($pattern, $val);
                        }
                }
                return $pattern;
        }
?>
```

III. MESSAGE DISPLAY

```
<?php
        function pattern_match($pattern){
                include 'db_connection.php';
                $query = "SELECT msg FROM complaint_pattern WHERE pattern =
'".$pattern."'";
                $result = mysqli_query($con,$query);
                $row = mysqli_fetch_assoc($result);
                return $row['msg'];
        }
        function message_display($mark){
                include 'db_connection.php';
                $default = "<h3 align='center'>Sorry we dont have temproary solution for the
problem you have mentioned, the solution may be updated in the future!</h3>";
                if(!(is_null($mark))){
                        $query = "SELECT message FROM display_message WHERE id =
".$mark;
                        $result = mysqli_query($con,$query); echo mysqli_error($con);
                        if(mysqli_num_rows($result) > 0){
                                $row = mysqli_fetch_assoc($result);
                                return $row['message'];
```

```
                }
                else{
                        return $default;
                }
        }
        else{
                return $default;
        }
    }
?>
```

APPENDIX C
REFERENCES

[1] R. Kaptein and J. Kamps. “Word clouds of multiple search results”. In IRFC 2011, volume 6653 of LNCS, pages 78–93, 2011.

[2] Shailaja Jayashankar, R. Sridaran, “Moving word cloud from visual to text analysis to endow eLearning”, Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference, March 2016.

[3] Peng Wei, Tongge Xu, Xi Qin, Chao Wang “Visualization of Police Intelligence Data Based on Word Clouds”, Computational Intelligence and Security (CIS), 2014 Tenth International Conference, Nov. 2014.

[4] Masahiko Itoh, Naoki Yoshinaga, Masashi Toyoda “Word-Clouds in the Sky: Multi-layer Spatio-Temporal Event Visualization from a Geo-Parsed Microblog Stream”, Information Visualisation (IV), 2016 20th International Conference, July 2016.

[5] Steffen Lohmann, Florian Heimerl, Fabian Bopp, Michael Burch, Thomas Ertl “Concentri Cloud: Word Cloud Visualization for Multiple Text Documents”, Information Visualisation (iV), 2015 19th International Conference on, July 2015.

[6] Carla Binucci, Walter Didimo, Enrico Spataro, “Fully dynamic semantic word clouds”, Information, Intelligence, Systems & Applications (IISA), 2016 7th International IEEE Conference, 19 December 2016.

[7] Ming-Te Chi, Shih-Syun Lin, Shiang-Yi Chen, Chao-Hung Lin, Tong-Yee Lee, “Morphable Word Clouds for Time-varying Text Data Visualization”, IEEE Transactions on Visualization and Computer Graphics Volume: 21, Issue: 12, Dec. 1 2015.

[8] Florian Heimerl, Steffen Lohmann, Simon Lange, Thomas Ertl, “Word Cloud Explorer: Text Analytics based on Word Clouds”, System Sciences (HICSS), 2014 47th Hawaii International Conference, 10 March 2014.

[9] R. Kaptein, D. Hiemstra, J. Kamps, “How different are language models and tag clouds”, ECIR-2010, Volume 5993 of LNCS, Pages 556-568, 2010.

[10] Michael Burch, Steffen Lohmann, Fabian Beck, Nils Rodriguez, Lorenzo Di Silvestro, Daniel Weiskopf, “RadCloud: Visualizing Multiple Texts with Merged Word Clouds”,Information Visualisation (IV), 2014 18th International Conference, july 2014