DESKPORT – Automated Customer Support Ticket System

A.Praveena[1], R.N.Devendra Kumar[2]
[1]Assistant Professor, Department of CSE, Jansons Institute of Technology, Coimbatore
[2]Assistant Professor(Senior Grade), Department of CSE, Sri Ramakrishna Institute of Technology, Coimbatore
praveena.s@jit.ac.in, devapsna@gmail.com

Abstract— Automated Customer Support Ticket Systems should retrieve exactly the information required to assist user as quickly and as easily as possible either for a lay user who knows little about the domain or for an advanced user who requires more specialist information. Automated customer support systems should also be easily maintainable as knowledge in domains where help is required often changes very rapidly, for example help for computer users. The aim of this study was to develop a help desk information retrieval mechanism suitable. Ease of maintenance is provided by using Multiple for a wide range of users and to provide a way of easily maintaining the system. The prototype developed for use over the World Wide Web combines keyword search and case based reasoning to provide both rapid focusing on a part of the help information and guided interaction when the user is unclear about appropriate keywordsClassification Ripple down Rules to maintain the domain knowledge in the system. Further issues that arise include the problem of inappropriate focusing by keyword and maintenance in a distributed environment.

***Index Terms*— Ticket System, Customer Support**

I. INTRODUCTION

In many areas various forms of help desk service provide users with help. In conventional help desk services, groups of human experts who differ in their knowledge and expertise try to solve the customer’s problems. Their roles are determined according to their problem solving ability and the degree of the problem difficulty. Thus, to provide help desk service of high quality, the availability of high level experts is crucial. However, the number of such high level experts is limited, and the demand for the automated help desk systems is increasing. An expert system approach is a feasible solution. In addition to this, world-wide computer networks such as the Internet are becoming the major communication media. The rapid communication enabled by such computer networks has also increased the demand for efficient maintenance of the knowledge for the help desk services. The overall aim of this study is to develop better methods of maintaining knowledge bases for help desk system while improving their usability.

For the discussion here knowledge based systems are roughly classified into two groups, i.e., rule based and case bases. Although, the rule representation is most popular in the expert system community, the case based reasoning (CBR) approach has been frequently used to build help desk systems (Kriegsman and Barletta, 1993; Barletta, 1993a; Shimazu et al., 1994; Simoudis, 1992). Most of these systems, however, require a major effort to maintain the case base. We propose to use the Multiple Classification Ripple Down Rules (MCRDR) method to reduce the case base maintenance cost and speed up the maintenance process. The MCRDR method is a case based maintenance method with which the expert can develop and maintain the case base without the help of the knowledge engineers. In the MCRDR method, when the CBR system retrieves cases which are identified by the user as inappropriate, the systems simply asks the expert to identify the important features which distinguish the incorrectly retrieved cases from the present case. The expert also adds the relevant information needed by the user to the new case so it can be added to the case base for future retrieval. It is indexed using the distinguishing features identified by the experts and information from the previous incorrect retrieval of the other cases (Kang and Compton, 1994). This simple approach allows large systems to be easily built (Compton et. al., 1993). The earlier simpler version of MCRDR, Ripple Down Rules (RDR), was used to maintain the Pathology Expert Interpreting Report System (PEIRS) in St. Vincent Hospital, Sydney. This system showed a very high level of performance and was developed and maintained by experts as part of their normal duties without any knowledge engineering support (Edwards et al., 1993; Compton et al., 1994; Kang et al., 1994; Perston et al., 1994).

In this study, it is assumed that the help desk system is to be used by various users from experts to novices. It is also designed to be constructed and maintained through the World Wide Web (WWW) with remote users directly retrieving information. How to provide a suitable interface for the various types of the remote user, especially for the novice user, and how to maintain the consistency of the case base perhaps maintained by the multiple experts are important research issues in this study. In some sense, a help desk system can be seen as an information retrieving system. The information is saved as cases. The major difference between the proposed method and the conventional information retrieving methods is that the proposed method focuses on using human expertise to develop and maintain the way in which a user interacts with the system to produce appropriate information retrieval.

II. RELATED WORKS

A user’s request to a help desk service can be classified into two types: information search and diagnosis of his/her problems. One may simply seek for new information. ―What is WWW?‖ is a typical example of this type of information search. One may also seek for a solution to his/her problem. ‖My printer does not work!‖ is a typical example of this type of diagnosis. The issues related to these requests are studied in the information retrieval area and the knowledge based diagnosis area. In this section, we briefly summarize these related issues together with knowledge maintenance issues and interface issues which are also important when we try to develop a practical system.

A. Information Retrieving Studies

Many information retrieval studies focus on how to find the relevant information from a large text base. A simple approach is collecting related documents and providing a search engine with the collection. The major research issues in this area fall into 1) the text representation, 2) the user query representation, and 3) the retrieving method. The text representation is one of the classical issues in the information retrieving studies. The simple approach extracts all words in the documents with exceptions such as pronouns and articles. Use of the statistical measurement of word appearance, i.e., the term frequency, is also used to supply additional information (Salton et al., 1994, Lewis, 1992).

The user query representation is studied to accurately capture the user requests. The simple approach is to capture the requests by keyword combinations. Natural language understanding and sophisticated interaction techniques (Callan and Croft, 1993) are also studied to provide a better interface.

The retrieval function (Salton and J. McGill, 1983) actually selects and ranks the documents. The ranking method is particularly important when many documents are selected. Since simple boolean logic does not cover the ranking, various statistical methods such as k-nearest neighbors are used to provide the ranking for the selected documents. These information retrieving methods are useful in constructing a help desk system. This is particularly so if the task is to search for new information. Diagnosis type help services can also be handled by providing the relevant documents. However, this type of system assumes that the user can specify the appropriate keywords to search for the related documents. If the user does not have the skill to provide proper keywords, the system may fail to retrieve the relevant documents or may find too many irrelevant documents. Since natural language understanding still has a performance problem, how to provide better information retrieval to the novice user who is lacking such skills remains a research issue.

B. Expert System Approach-Rule Based Approach

Many expert systems have been developed for diagnostic problems (Boose, 1989). Rules are the most popular representation for the knowledge base in the expert system. The methods to obtain rules are classified into two categories, automated methods (machine learning) and manual methods (e.g. interviewing) (Boose, 1991). Regardless of which method is used, the rule based approach constructs a knowledge base which interprets the problem and suggests solutions. Though rules in the knowledge base are a good source of help for the user, they are different from a set of documents which are used in an information retrieval system.

There are common criticisms about rule based approaches (Barletta, 1993b). The first one is that it is hard to construct a knowledge base. The second one is that it is hard to maintain the knowledge base. Another criticism is that a rule based system is brittle. The second criticism is particularly crucial for the help desk system development since it should be able to accommodate changing knowledge. Note that these criticisms are based on the classical rule based approaches. There have been many attempts to solve these problems. The most common approach is based on the idea of a ―knowledge level‖ (Newell, 1982) analysis of a situation in a software engineering type of approach to knowledge acquisition (Wielinga et al., 1992).

C. Expert System – Case Based Approach

Although many have worked on rule based systems, a case based reasoning(CBR) approach is frequently used to build help desk systems. SMART(Acorn, 1992), CASCADE (Simoudis, 1992) and CARET (Shimazu, et al., 1994) are the examples of help desk systems which use a CBR approach. CBR builds expert systems using past cases to solve new problems (Sycara and Ashley, 1991). It is based on the cognitive assumptions that real expertise comes from the experience of the expert, and that episodic memory (Slade, 1991; Stottler, et al., 1989) is an appropriate way to model the expertise. The approach of CBR is not to find appropriate rules in a knowledge base, but to find similar cases from the case base. CBR is appropriate when there is no formalized knowledge in the domain or where it is difficult for the expert to express their expertise in the rule format. In general, an expert is good at judging cases but not good at providing knowledge in the abstract (Manago and Kodratoff, 1987).

The functional similarity between CBR and information retrieval methods is that both methods carry out their task by retrieving the relevant cases or documents. Both methods maintain a set of cases/documents and the new cases/documents are added into the database or the later use

(Barletta, 1993b). While information retrieving studies concentrate on retrieval from large document data bases (Barletta, 1993b; Callan and Croft, 1993), CBR approaches try to represent the human problem solving knowledge in the case representation. Many CBR researchers claim that the knowledge acquisition bottleneck is solved by maintaining a case base since the addition of new knowledge into the system can be performed by the simple addition of new cases. However, a CBR system needs a good case retrieval mechanism and a good case base maintenance method. If a CBR system lacks these methods, it can not solve a problem because it may find consistent, irrelevant or outdated cases.

III. SYSTEM DEVELOPMENT MODEL - MVT

A. MVT-(Model View Template)

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data. The Template is a presentation layer which handles User Interface part completely.

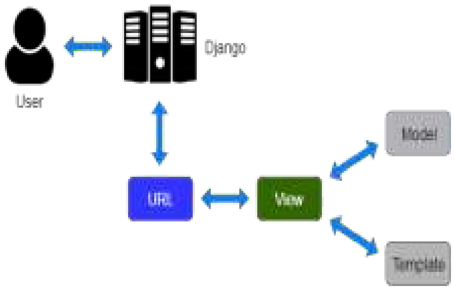


Fig 1 : Model View Template

MODEL: Model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySql, Postgres).

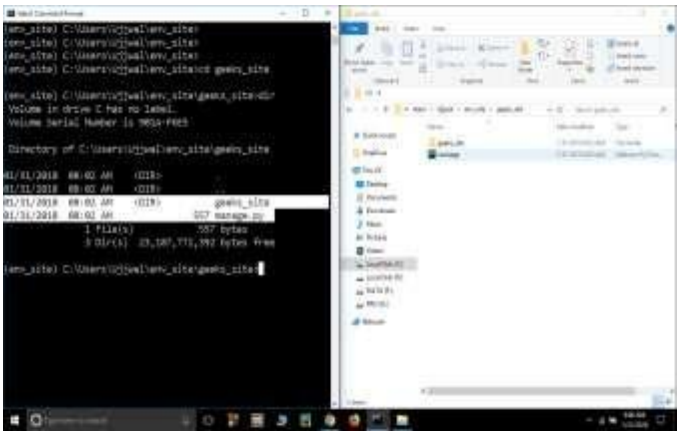
VIEW: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.

TEMPLATE: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

PROJECT STRUCTURE: A Django Project when initialised contains basic files by default such as manage.py, view.py, etc. A simple project structure is enough to create a single page application. Here are the major files and there explanations.

Fig 2-MVT Structure

Manage.py: **T**his file is used to interact with your project via the command line(start the server, sync the database… etc). For getting the full list of command that can be executed by manage.py type this code in the command window-



```
(env_site) C:\Users\Ujjwal\env_site>
(env_site) C:\Users\Ujjwal\env_site>
(env_site) C:\Users\Ujjwal\env_site>
(env_site) C:\Users\Ujjwal\env_site>cd geeks_site

(env_site) C:\Users\Ujjwal\env_site\geeks_site>dir
 Volume in drive C has no label.
 Volume Serial Number is [illegible]

 Directory of C:\Users\Ujjwal\env_site\geeks_site

01/31/2018  08:02 AM    <DIR>
01/31/2018  08:02 AM    <DIR>
01/31/2018  08:02 AM    <DIR>          geeks_site
01/31/2018  08:02 AM               557 manage.py
               1 File(s)            557 bytes
               3 Dir(s)  23,187,771,392 bytes free

(env_site) C:\Users\Ujjwal\env_site\geeks_site>
```

Folder: This folder contains all the packages of your project. Initially it contains four files –_init_.py - It is python package.

settings.py - As the name indicates it contains all the website settings. In this file we register any applications we create, the location of our static files, database configuration details, etc.
urls.py - In this file we store all links of the project and functions to call.
wsgi.py - This file is used in deploying the project in WSGI. It is used to help your Django application communicate with the web server.

IV. UNITS

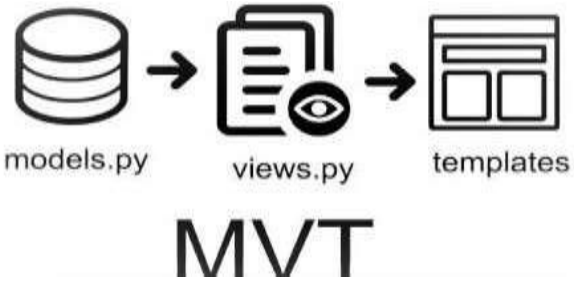
A. AIM & OBJECTIVE

The main aim of this product is providing all functionality related to one software industry. It tracks all the information of the employee in all the department. We are going to develop all type of CRUD (create, read, update and delete) operations of the employee. This is role-based module where admin can perform each and every operation on data. And employee can submit the ticket token and can view the data, so access level restrictions also going to implement in this project.

B. PROJECT OVERVIEW

Tickets, calls, faulty equipment, people complaining. IT support moves at supersonic speed and, I your IT professionals have to manage everything with, say,Outlook, your company may be losing a lot of money.

Pair an email client with the high wages typical for the



industry and you’ll get yourself in a ―throwing money down the drain‖ situation. Resolving a single tech support case can cost anywhere between $2.93 and $49.69 and , if you offer tiered IT support , you can expect the expense to triple every time an issue is escalated.

In short, it’s best to help your IT support staffs to be as possible. Since you’re reading this article, you’re probably aware that the solution to this problem is an IT ticketing system. It is true . Such software can save up to 670 working hours per year. Generally speaking, when an IT support ticket system does is to automate manual tasks and help IT staff manage the lifecycle of each request.

At the heart of an IT support ticket system is, as the name suggests, ticket management. The system creates a ticket each time a new case is submitted, appending all relevant customer and incident data to it. Some systems also have advanced tools for asset management, IT change management, network diagnostics, problem and incident management, etc. Note that the term ―IT help desk‖ and ―IT service desk‖ can somethimes be used interchangeably even though, strictly speaking, they are two separate entities. Unlike the help desk , the service desk puts an emphasis on the users or , as the ITIL framework defined it, a service desk is ―point of communication between the service provider and all its users.‖ By contrast, when IT help desks first appeared in 1980s, they were simply a means of reporting and resolving incidents on an ad-hoc basis.

The terminology gets even more confusing when your IT department doubles as a customer service department . Fortunately , most IT ticketing software can be used for both employee support and customer support. So , you can get away with using the wrong term.

V. FEATURES

*WHAT TO LOOK FOR?:*When looking for a helpdesk ticketing system, it's important to consider both your customer's needs and your agents’ workflow. Not all systems offer the full set of features and you may end up using a combination of tools to achieve the desired results. In general, here’s what to look for when choosing IT ticketing software.

SELF SERVIICE OPTIONS

EXTERNAL KNOWLEDGE BASE

An external knowledge base creates a rare win-win-win (yes, that’s a triple win) situation for customer support. Customers gladly resolve issues themselves which reduces ticket volume and your company’s overall support costs. Among other things, you can use an external knowledge base to publish customer FAQs, product and service updates, and troubleshooting guides. Customers can browse by category or search the knowledge base which helps you deflect ticket creation while keeping them happy—they find the information they need instantly.

INTERNAL KNOWLEDGE BASE

An internal knowledge base helps your IT team work more efficiently because it enables employees to serve themselves. Your busy users will appreciate the ability to find the answers and instructions they need without having to wait. An internal knowledge base can also be extremely useful to agents when dealing with service requests. It can store SOPs for handling requests as well as how-to’s, technical documentation, and best practices.

Not all helpdesk ticket software comes with a knowledge base but you may use a third-party solution.

UNIIFIED SMART INBOX

One of the crucial features of an IT ticketing system is the ability to create helpdesk tickets from various sources such as email, live chat, social media, phone calls, etc. If there is a single dashboard to view and manage all incoming tickets, your team will be more productive.

Here’s what a unified inbox should enable you to do:

- Create tickets manually
- Use quick pre-populated templates for common issues
- Send an automatic response to the end-user when she submits an issue, giving her the ability to reply
- Attach files
- Keep all past conversations in the same ticket to provide context
- Prevent two agents from responding to the same ticket at the same time (collision detection)
- Enable agents to collaborate on a ticket, e.g. by leaving internal notes

AUTOMATION

Most ticketing systems allow you to define triggers and rules to change ticket status, priority, category, and other ticket attributes automatically. Some even give you the ability to define custom workflows, eliminating the larger portion of manual work. Here’s how this works in practice.

Let’s suppose that your ticketing system software lets you set up rules to auto-triage and auto-prioritize tickets based on the email address, subject line, ticket category, and other conditions. You can create a rule to tag all issues coming from a specific email (an important client or a C-level executive) and assign the tickets to your most experienced IT support professionals.

Automation is especially useful to companies that need to tie a custom set of products and services to each customer (e.g. MSRs).

SECURITY

Customer details are often attached to service tickets, passwords are sent back and forth, and so on. Ticketing system software should be able to protect your sensitive data.Some companies opt for on-premise installations to eliminate all potential threats. Some prefer cloud-based ticketing systems. There are benefits and drawbacks to both but security is a concern in all cases.

Here are some important security features to look for in an IT helpdesk:

- IP restrictions to ensure only your staff can log in to the back end
- Message encryption to protect sensitive data
- Access control to define user groups and privileges within your organization
- Virus scanning to ensure no harmful file attachments can be sent
- SSL URL encryption for cloud-based systems

VI. FUNCTIONALITIES

ADMINISTRATOR

i. Login to the first page.
ii. Create new faculty account.
iii. View all student details.
iv. View all faculty details.
v. Check request sent by the user.
vi. Forward request to the faculty as per requirement.
vii. View all queries.
viii. View new queries.
ix. Reply/delete queries.
x. Add new notice for students.
xi. Add new notice for faculty.
xii. Logout.

CLIENTS

i. Register him/her.
ii. Login to the first page.
iii. Change the password after login into the system.
iv. Edit user details.
v. Create queries in the system.
vi. View the status of the query.
vii. View the list of queries created by him/her over the past.
viii. Create a new query by specifying the facility, the severity of the query and a brief description of the query.
ix. Edit or delete queries created by him/her.
x. Live chat with the staffs.
xi. Logout.

a. STAFFS

i. Login to the first page.
ii. Change the password after login into the page.
iii. Edit details of his/her profile.
iv. View all queries.
v. View new queries.
vi. See the queries created by the users and assigned by the admin.
vii. Work on the queries assigned to them.
viii. Reply/delete the query.
ix. View replied queries.
x. Add new notice for clients.
xi. View all notice.
xii. Logout.

VII. IMPLEMENTATION

Systems Implementation is the fourth of five phases in the systems development life cycle. Includes application development, documentation, testing, training, data conversion, and system changeover

The deliverable for this phase is a completely functioning information system

- Explain the importance of software quality assurance and software engineering
- Describe the application development process for structured, object oriented, and agile methods
- Draw a structure chart showing top-down design, modular design, cohesion, and coupling
- Explain the coding process
- Explain unit, integration, and system testing
- Differentiate between program, system, operations, and user documentation
- List the main steps in system installation and evaluation
- Develop training plans for various user groups, compare in-house and outside training, and describe effective training techniques
- Describe data conversion and changeover methods
- Explain post-implementation evaluation and the final report to management

The system design specification serves as a blueprint for constructing the new system. The initial task is application development. Before a changeover can occur, the system must be tested and documented carefully, users must be trained, and existing data must be converted. A formal evaluation of the results takes place as part of a final report to management.

Application Development

Application software is all the computer software that causes a computer to perform useful tasks (compare with computer viruses) beyond the running of the computer itself. A specific instance of such software is called a software application, program, application or app.

The term is used to contrast such software with system software, which manages and integrates a computer's capabilities but does not directly perform tasks that benefit the user. The system software serves the application, which in turn serves the user.

Examples include accounting software, enterprise software, graphics software, media players, and office suites. Many application programs deal principally with documents. Applications may be bundled with the computer and its system software or published separately, and can be coded as university projects

Application software applies the power of a particular computing platform or system software to a particular purpose. Some applications are available in versions for several different platforms; others have narrower requirements and are thus called, for example, a Geography application for Windows, an Android application for education, or Linux gaming. Sometimes a new and popular application arises which only runs on one platform, increasing the desirability of that platform. This is called a killer application.

Traditional methods

A software development process, also known as a software development life-cycle (SDLC), is a structure imposed on the development of a software product. Similar terms include software life cycle and software process. It is often considered a subset of systems development life cycle. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a life-cycle model a more general term and a software development process a more specific term. For example, there are many specific software development processes that 'fit' the spiral life-cycle model. ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

Documentation:

The variation and complexity of requirements documentation makes it a proven challenge. Requirements may be implicit and hard to uncover. It is difficult to know exactly how much and what kind of documentation is needed and how much can be left to the architecture and design documentation, and it is difficult to know how to document requirements considering the variety of people who shall read and use the documentation. Thus, requirements documentation is often incomplete (or non-existent). Without proper requirements documentation, software changes become more difficult — and therefore more error prone (decreased software quality) and time-consuming (expensive).

A very important part of the design document in enterprise software development is the Database Design Document (DDD). It contains Conceptual, Logical, and Physical Design Elements. The DDD includes the formal information that the people who interact with the database need. The purpose of preparing it is to create a common source to be used by all players within the scene. The potential users are:

- Database designer
- Database developer
- Database administrator
- Application designer
- Application developer

A common complaint among users regarding software documentation is that only one of these three approaches was taken to the near-exclusion of the other two. It is common to limit provided software documentation for personal computers to online help that give only reference information on commands or menu items. The job of tutoring new users or helping more experienced users get the most out of a program is left to private publishers, who are often given significant assistance by the software developer.

System Installation:

On Windows systems, this is the most common form of installation. An installation process usually needs a user who attends it to make choices, such as accepting or declining an end-user license agreement (EULA), specifying preferences such as the installation location, supplying passwords or assisting in product activation. In graphical environments, installers that offer awizardbased interface are common. Attended installers may ask users to help mitigate the errors. For instance, if the disk in which the computer program is be installed was full, the installer may ask the user to specify another target path. Installation that does not display messages or windows during its progress. "Silent installation" is not the same as "unattended installation" : All silent installations are unattended but not all unattended installations are silent. The reason behind a silent installation may be convenience or subterfuge. Malware is almost always installed silently.

II. CONCLUSION AND FUTURE ENHANCEMENT

Thus this project has been executed successfully and the output has been verified. All obtained outputs are according to committed in abstract. Initially more problems occurred during the architecture creation. As mentioned above architecture has been implemented successfully. The contents of the case base can be easily maintained by a human expert with the help of MCRDR functions as MCRDR can keep track of the problem solving contexts in the past and store them in the knowledge base. The combined use of keyword search and MCRDR seems likely to provide an excellent interface for a range of users for the help desk system by reducing unnecessary interaction between the system and the user. Since the frame work is quite general, it could be applied to various kinds of help desk systems.

A. FUTURE ENHANCEMENT

Even though many features are implemented in this project, here some of the future enhancement can be done for this application. All the provisions and database design are made in the current project for the future enhancements. So this current system supports various enhancements, they are follows.

- Mobile responsive web application.

- A dedicated mobile application can be develop of Android, IOS, Blackberry and windows. So that people can enjoy the responsive mobile based web application.

- Offline features

o By using adaptive memory management, offline features can be implementing. This feature will load most of the information during online and access or download the data during the offline mode.

- Supports many languages

o Can implement this feature for more languages like Chinese, Arab, French, German and etc. So that it supports multilingual, also can be used in many countries.

References

[1] Joyeeta Bagchi, ―Online Help Desk: Published on College of Engineering andMangement,Kolaghat–May2015. https://www.slideshare.net/JoyeetaBagchi/online-help-desk

[2] Kumar Kartikeya Upadhyay, ― Help Desk System: Published on NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY,Greater Noida(UP)- Apr 2015 . https://www.slideshare.net/kartikeyaupadhyay/help-desk-system-report

[3] Nagarjuna VS, Anjaneedevi K,Akhila reddy J,Vidya Sagar K, ―Online Help Desk: Published on Campus Connect(an Infosys industry-academia partnership program) - Aug 2010 . https://www.slideshare.net/nagarjunagoud/online-help-desk-ppt-5097673