

University Support System

BY

Afroja Shorno

ID: CSE1801013013

Habiba Rhaman

ID: CSE1801013018

Md. Monoar Hossen

ID: CSE1801013048

Naimul Hasan Tosif

ID: CSE1803015014

Supervised by

Khadija Islam

Submitted in partial fulfillment of the requirements for the degree of Bachelor of
Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SONARGAON UNIVERSITY (SU)**

September 2022

APPROVAL

The project titled “**University Support System**” submitted by Afroja Shorno (CSE1801013013), Habiba Rhaman (CSE1801013018), Md. Monoar Hossen (CSE1801013048), Naimul Hasan Tosif (CSE1803015014) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Khadija Islam
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the project performed by us under the supervision of **Khadija Islam**, lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

Khadija Islam
Supervisor

Afroja Shorno
ID: CSE1801013013

Habiba Rhaman
ID: CSE1801013018

Md. Monoar Hossen
ID: CSE1801013048

Naimul Hasan Tosif
ID: CSE1803015014

ABSTRACT

When the COVID-19 Pandemic started spreading across the world in 2019, the reaction by universities was swift. Many higher education providers made the move to online learning virtually overnight, after that all students suddenly depend on online not only for classes but also various kinds of support. Sometimes students face various problems like accounts, course, credit, exam, etc related but there is no support system without WhatsApp messaging, email or phone calls which takes more time. But most of the time students are not aware of the correct concern person so that it's a matter in question to whom to look for a solution. Most of the time It's difficult to manage time to go to our university without class days to resolve the problem or it is not possible to reach out to authorities without office duration for support on time. Apart from that in this universe everyone is getting busy day by day. We all are searching for not only new things but also to be conscious about time. Focused on that we want to launch the **University Support System (USS)**. USS service is an essential component of e-support provision. Students' retention, success and satisfaction are their main objectives. Although the delivery of the University Support System (USS) May vary between institutions, some aspects of student support should be considered in all e-support programs. Summarizing, University Support System services for students which are pedagogic, technical and administrative aspects that affect the online support. Through these service web students can reach any kind of solution without any hassle. Also, it will be time consuming for both sides.

The programming language or front-end tool used here is PHP, HTML and database or back-end used here is MySQL. Unified Modeling Language will be used to analyze current system procedures and problems' requirements, design a logical solution to the problems, and then implement the result through programming language and database. My approaches allow the same concepts and notation to be used throughout the entire software development process.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Khadija Islam**, Associate Professor & Head, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Dr Md Alamgir Hossain**, Dean, Faculty of Science and Engineering for his kind concern and precious suggestions. We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

| | |
|--------------|---|
| FAQ | Frequently Asked Question |
| FOW | Future of Work |
| HCI | Human Computer Interaction |
| HTML | Hyper Text Markup Language |
| ICT | Information and Communications Technology |
| IT | Information Technology |
| MySQL | Structured Query Language |
| PHP | Hypertext Preprocessor |
| SLA | Service-Level Agreement |
| USS | University Support System |

TABLE OF CONTENTS

| Title | Page No. |
|--|----------|
| DECLARATION | iii |
| ABSTRACT | iv |
| ACKNOWLEDGEMENT | v |
| LIST OF ABBREVIATION | vi |
| CHAPTER 1 | 1– 5 |
| INTRODUCTION TO UNIVERSITY SUPPORT SYSTEM | |
| 1.1 University Support System | 1 |
| 1.2 Knowledgebase | 2 |
| 1.3 Challenges | 2-3 |
| 1.4 Initial Study | 3-4 |
| 1.5 Background of the Study | 4 |
| 1.6 Theoretical/Conceptual Framework | 5 |
| 1.7 Project Vision and Aim | 5 |
| 1.8 Project Scope | 6 |
| 1.9 Hardware and Software Requirement | 6 |
| 1.10 Further Enhancements | 7 |
| CHAPTER 2 | 8-11 |
| PROJECT BACKGROUND | |
| 2.1 Project Management | 8-9 |
| 2.2 Development Methodologies | 9-10 |
| 2.3 Summary | 10-11 |
| CHAPTER 3 | 12-17 |
| SYSTEM REQUIREMENTS | |
| 3.1 Requirements Analysis | 12-14 |
| 3.2 Requirements Capture | 14 |
| 3.3 Research | 14 |
| 3.4 Requirement Analysis | 15 |
| 3.5 Use Cases | 15 |
| 3.6 Functional Requirements | 16 |
| 3.7 Best Practice Guidelines | 17 |

| | | |
|---------------------------|--|-----------|
| CHAPTER 4 | | 18– 25 |
| TECHNICAL ANALYSIS | | |
| 4.1 | Component of Technical Analysis | 18-19 |
| 4.2 | User Interface | 19-20 |
| 4.3 | Task Identification | 20 |
| 4.4 | Short Term Memory | 21 |
| 4.5 | Execution-Evaluation Cycle | 21 |
| 4.6 | Form Filling | 21-22 |
| 4.7 | Menu's | 22 |
| 4.8 | System Options | 22 |
| 4.9 | Student/Server Architecture | 22-23 |
| 4.10 | Web Technologies | 24 |
| 4.11 | Active Server Pages | 24 |
| 4.12 | PHP | 24-25 |
| 4.13 | Summary | 25 |
| | | |
| CHAPTER 5 | | 26– 34 |
| SYSTEM DESIGN | | |
| 5.1 | System Block Diagram | 26 |
| | 5.2.1 Application Architecture | 27 |
| | 5.2.2 Database Modeling and Design | 28 |
| | 5.2.3 Conceptual Database Design | 28-29 |
| | 5.2.4 Logical Database Design | 29 |
| | 5.2.5 Data Dictionary/Database Files | 30 |
| | 5.2.6 Data Elements | 30 |
| | 5.2.7 Functional Design | 30 |
| | 5.2.8 System Maintenance | 30 |
| 5.3 | User Interface | 31 |
| 5.4 | System Implementation | 32 |
| | 5.4.1 Development Environment | 32-33 |
| | 5.4.2 Data Access | 33-34 |
| | 5.4.3 Page Template | 34 |
| | | |
| CHAPTER 6 | | 35-44 |

SYSTEM FUNCTION AND OUTCOME

| | | |
|--------|-------------------------------|-------|
| 6.1 | Functional Testing | 35 |
| 6.2 | User Acceptance Testing | 36 |
| 6.3 | Usability Testing | 36-37 |
| 6.4 | System Outcome | 38 |
| 6.5 | Outcome Interface | 38 |
| 6.5.1 | Log In | 38 |
| 6.5.2 | Registration | 38 |
| 6.5.3. | User Panel | 39 |
| 6.5.4 | Admin Panel | 40 |
| 6.5.5 | Admin Task | 40 |
| 6.5.6 | Academic Supporter | 44 |
| 6.5.7 | Academic Supporter Task | 44 |

CHAPTER 7 46-47

CONCLUSION AND FUTURE WORKS

| | | |
|-----|-------------------|-------|
| 6.1 | Conclusion | 46 |
| 6.2 | Future Works..... | 46-47 |

REFERENCES 48

LIST OF FIGURES

| <u>FigureNo.</u> | <u>Title</u> | <u>Page No.</u> |
|-------------------------|---|------------------------|
| Fig.2.2 | The Waterfall Model | 10 |
| Fig.3.5.1 | User Diagram | 15 |
| Fig.3.5.2 | Use Case Diagram | 15 |
| Fig.4.9 | Current and Proposed Architecture | 23 |
| Fig.5.1 | Application Architecture - UML Deployment Diagram | 28 |
| Fig.5.2.1 | System Block Diagram | 26 |
| Fig.5.8 | Update Problem | 30 |
| Fig.5.9 | User Problem Creation | 32 |
| Fig.6.2 | Usability Testing | 36 |
| Fig.7.3 | All Problem List | 37 |
| Fig.6.5.1 | Login Interface | 38 |
| Fig.6.5.3 | Registration Panel | 39 |
| Fig.6.5.4 | User Problem Creation | 40 |
| Fig.6.5.7 | Academic Support Analytic Chart | 44 |
| Fig.6.5.8 | Academic Support Problem Details | 45 |

CHAPTER 1

Introduction

USS (University Support System)

University Support System software offers numerous benefits to system administrators and IT pros. University students always appreciate a helpful resource for their potential issues and queries; when students submit a report, they're assured that their problems are forwarded to the correct member of the support staff. Once a report has been submitted to the system, the students will have the ability to log in and track the progress of their problem.

University Support System software acts as a web-based problem system that manages inquiries, as well as other types of support processes. The software also ranks inquiries and classifies them all by priority. At the same time, the software transfers them to the appropriate department for issue resolution. This type of software can also help reduce the amount of training that's needed for the support staff. As a result, my support staff can become experts in a shorter amount of time. Such an advantage allows for a much speedier resolution of student's IT issues, which in turn frees up my support staff to support an even higher volume of students.

Support staff can also benefit from University Support System software as their jobs become easier. In addition, students will receive service in a more efficient manner and wait times are dramatically reduced. Because problem history is stored, the support staff is better able to accurately assess issues and take appropriate action. Another benefit to leveraging the University Support System is that managers have the ability to keep track of members and their performance in the university. Since the typical University Support System solution has resolution and tracking tools, reports are easily completed. Students will ultimately be more productive with shorter periods of downtime, which benefits the university as a whole.

USS is the University Support System which will ever need. Able to automatically assign support problems, keep users informed of the status of their problem and help requesters to adhere to SLA rules. HDTs is the quicker solution for end users and higher productivity for IT technicians.

Knowledgebase

The Knowledge Base is a database that contains problems reported by users with the corresponding solutions. It is not a FAQ in the sense that a question doesn't need to be frequently asked. A question should go into the knowledge base if a supporter considers that it could be of general interest. This allows users to reduce the necessity of creating problems by consulting the Knowledge Base.

A good knowledge base can save help desk staff so much time. IT technicians will be able to check the knowledge base for resolutions to problems and so will users.

We can make a number of resolutions public to users allowing them to solve minor issues themselves. Also, knowledgebase has the following benefits:

- Create knowledge rich articles to provide solutions, workarounds, and FAQs.
- Include rich text, images, and attachments to the knowledge base content.
- Ensure the quality of knowledge base content with a streamlined approval mechanism.
- Organize knowledge articles under configurable topics to let end users and technicians easily browse and access.
- Provide advanced keyword search capability and the solutions auto suggest feature to enable end users and technicians to quickly pull-out relevant knowledge articles.

Challenges

Information Technology (IT) support for end-users has emerged as one of the leading concerns of organizations. Continuous adapting and updating of new technologies have made development of effective and efficient problem-solving services challenging for organizations. Organizations must actively search for new ways to provide better problem solve services that can satisfy the growing student demands and expectations.

Support System is a student support center in an organization that provides information, administrative and technical support to users, with the view to solving problems that users encountered in the course of using the organization resources or facilities. A Support System

could consist of one person or group of persons that make use of telephone devices or software applications to keep track of problem(s) status and thus provide solution(s) that satisfy the users.

Support system could also be seen as an information and as an assistance resource that supports the functionality of an organization by responding to users' requests in a timely manner. It is hence, a core sector through which problems, complaints and requests are reported, managed, coordinated and resolved. USS software is a solution application that is used for managing an organization's help desk. It is accessible to student support personnel who could direct request(s) to the servicing department(s).

Technical concerns are becoming a normal scenario in everyday work environments both in education and corporate. Thus, we need to constantly and effectively monitor these concerns. These require a system that can handle them. With this in mind, an Automated Support System: Student Support for Information Technology Resource Center is a fit solution that can provide an effective approach in handling all reported technical concerns with proper record keeping and monitoring to students and technical personnel as well as systems administrators. The most business area is converting into computerized systems. I will take on a challenge to make a project, which is based on the common concept problem System in any types of organizations willing to automate their support activities system through online. This project will help me to have a great chance to increase my ICT knowledge as well as software development like project management knowledge, programming knowledge etc. and serve the society as well.

Also, we will gather knowledge about real life Software development like software analysis, software design and development, database concept, testing knowledge and Implementation.

Initial Study

The concern of the study is to develop and design an Automated Support System. The system is deployed to our Tiny Coders Group, to test its functionality and usefulness. Based from the findings, we can herewith drawn the following conclusions:

1. The assessment of the respondents in terms of support tools to help desk which is available to the students reveals that help desks provide a variety of online tools and resources for the students to use to resolve their IT-related problems.

2. Self-help tools can effectively extend the help desk's hours of availability, allowing users to get answers

to their questions when the help desk is not staffed. Even during the help desk's normal operating hours,

The availability of self-service resources can reduce demand for direct interaction with the Support System Agent while keeping service availability and quality.

3. The assessment of the respondents in terms of proposed automated Support System performance shows that personnel have knowledge of information technologies to enhance teaching and learning, research, administration with continuous updates. Moreover, they appreciate the help desk especially the features and tools that it provided them as they utilized the system. These tools can stand alone and are functionally interrelated and integrated to the Support System automation systems.

Background of the Study

Regularly the term help desk is utilized for interior backing within the organization or for outside care groups. Numerous organizations are turning to support work areas to mechanize a mixed bag of errands and, at the same time, lessen costs by cutting staff and giving more student help from the current agent. Organizations need to give high caliber student administration and backing to get by in today's business surroundings.

Having the right help work area would guarantee high student fulfillment. Student help consolidates profits that support a student or student fathom and benefit from things limits by noting requests, handling issues and giving online information.

The preferences of automated Support work areas are basic in that they permit fewer individuals to manage larger work volumes. The university support system is increasing its importance as companies move to student-server architectures. Users who interface with the Support System often form a general perception of the information system group. Information systems play an important role within an organization. The Support work area is in charge of uniting an association's assets with a specific end goal to give its students quality help and administration.

Theoretical/Conceptual Framework

Support System automation is for many companies the first application area of knowledge-based systems “The automated help desk is a knowledge distribution while payroll was an automation of record keeping. The theory above anchors on the help desk management system which has attracted a number of research works. Such as, in the developed world, a support system has been established as a tool for inquiries made by users like students and staff of an institution for facilities and services. Further, the Support system information retrieval mechanism will be suitable for users in managing the complaints and proper system maintenance. The system helps improve support system usability and functionality. The figure above shows the help desk system entails the following: receiving requests, queries and complaints, generating reports on identified problems, classification of mails received, filing emails, responding to problems/ queries/ complaints stated in mails, and keeping track of problem status.

Project Vision and Aims:

Support System is designed and customized to provide businesses with an internal support system as well as a link for providing support to its students. Help desk applications host a number of benefits that includes:

- Giving existing students with information and Frequently Asked Questions (FAQ's) concerning the organization's frameworks and approaches.
- 24-hour availability thus catering to the trend of office personnel working late and to those overseas or in different time zones.
- Troubleshooting peculiarities gives students the capacity to take care of numerous help issues all alone. This apparatus gives the students brisk and simple arrangements and sparing the organization's cash.
- Serves as an instrument for following and recording help work area concerns, which gives an information base of resolutions to past exchanges concerning comparable issues.
- Supplies information concerning trends and other issues, which aids in the continuing improvement of products and services.

Project Scope

This is an Information Systems project that will cross many disciplines including management, software design, software development and systems architecture. The features of this project will be the most significant way of the resolution of the problem and student satisfaction. These features will be distinctive in comparison with the other similar types of application. Utilization and benefits of this project may increase the efficiency of the organization. The feature will be as follows:

- Problem Create
- Problem Assignment
- Problem Monitoring
- Problem Priority
- Student Admin Communication
- Assigning to different Department
- Notifications
- Reports

Hardware and Software Requirements

Minimum Software Requirements:

Operating System : Windows 7/ Windows 10

Application Server : XAMPP v3.3.0

Microsoft Windows Environment : Git bash / Github

Other Software Requirements:

Project Management Tools: Microsoft Project Server 2016

Designing Tools: Visual Studio

Text Editor: Notepad++, MS Office

Internet Browser: Mozilla Firefox, Google Chrome

Further Enhancements

Further enhancements could be made to the system prototype and will be implemented depending on the time available.

- Further development based on the evaluation of the prototype.
- Integration with Southtech Limited' time recording system.
- Ability to store details of and assign calls to a third-party engineer.
- Management reporting– enabling managers to obtain information from support calls such as response time and time taken to fix issues.

CHAPTER 2

Project Background

Project Background

This section outlines the need for a structured background in order to achieve a successful project. I strongly believe that a well-defined development cycle with simple, yet rigorous processes will allow us to deliver on time an efficient software system for student complaint management system. The following phases represent software development life cycle process:

1. Planning
2. Analysis
3. Design
4. Prototyping
5. Coding
6. Testing
 - Unit Test.
 - System Test.
 - Acceptance Test
7. Deployment
8. Training
9. Maintenance & Support
10. Project Handover

Project Management

Before starting any project, it is important to be clear of the outcomes that are expected, and the processes involved that will achieve those outcomes. There are a number of approaches that can be taken in order to manage the resources that are required to complete a successful project. Outline the activities to be carried out in project management as:

1. The feasibility study
2. Planning
3. Project execution

When faced with this particular project one finds oneself in a unique position. Whilst it is an academic assignment, it is also a live project that can bring genuine benefits to our group. Despite this, the feasibility study will not be carried out within the scope of the project. It is argued that the only way of building a full scenario of the project is to catch every fragment of detail as early as possible. Clearly, this will be

accomplished by considering carefully each aspect of the project before it is undertaken. However, planning is an ongoing process of refinement, and that each iteration of planning becomes more detailed and more accurate than the last. As such information gathering will be required at the beginning of each stage to ensure that it is carried out effectively and on time.

Development Methodologies

Having decided on a general approach to managing the project, a decision must be made on how best to execute the project plan.

Rapid Application Development:

This methodology puts emphasis on quickly producing prototypes for the users to evaluate. The Methodology is not suitable for all projects, particularly ones that require thorough planning. It is best used when there is a focused product scope, decisions can be made by few people, there are not many members in the project team, and the technical architecture is clear. This would not be a suitable method for this particular project because, as yet, there is not a focused product scope and the requirements are not clear. It is possible to adopt once all user analysis is complete, but then it will not constitute a methodology that can be applied to the project as a whole.

Joint Application Development:

Joint Application Development (JAD) is a methodology where developers work intensively with users in workshops and agree documented business processes. The advantage of using such a process is that all decision makers are generally in one room. This should speed up the communication process and enable decisions to be made instantly. On the other hand, the sessions may not cover everything that needs to be covered in one go. Prototyping, for example, could not take place unless many of these sessions were to be held. Due to the nature of this project and of business itself, it is unlikely that JAD will be adopted as the

methodology used. It is very unlikely that the directors, administration, and support staff will have time to all sit down together for an extended period of time to discuss what, essentially, is a small project. However, lessons can be learned from the methodology, and being based in Leeds, it is probably a good idea that meetings are arranged when the key decision makers will be around; this should help aid the communications process and ensure that decisions are made by the relevant people.

The Waterfall Model:

This approach suggests making just one attempt at a project and getting it correct the first time. When It works well, the waterfall approach allows project completion times to be forecast with more confidence than with some more iterative approaches allowing projects to be controlled effectively.

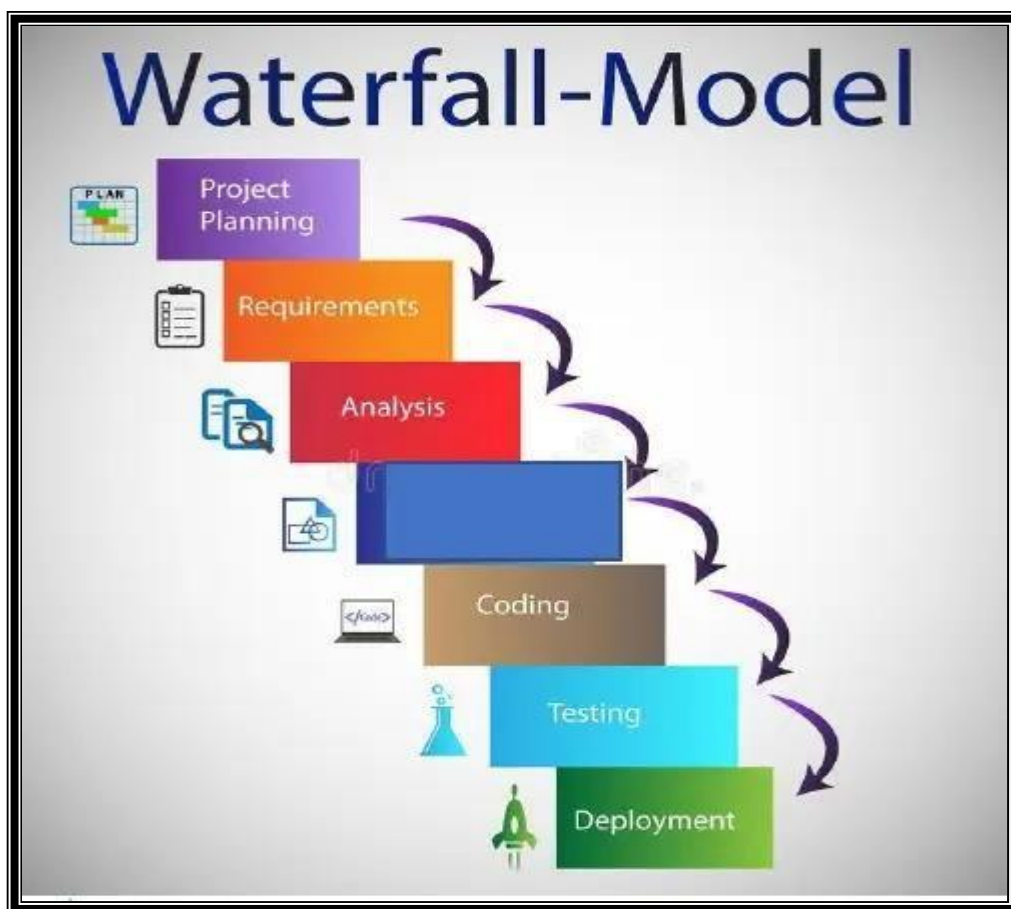


figure: 2.2

Whilst the approach is generally suited to the project at Southtech Limited, it is felt that a prototype may be necessary to gauge users' initial feelings about interface design and the functionality of the system. The process has the advantage of being able to determine exactly which stage the project is up to. For this reason, the overall aspect of the Waterfall cycle will be adopted for this project, although it will be altered somewhat in the coding, and testing stages. Instead of adhering to the waterfall approach for these levels, a prototype approach should be adopted that will allow for iterative implementation techniques to be utilized.

Software Prototyping:

Software prototyping enables developers to quickly portray the system functionality, can be classified as either throw-away or evolutionary, that is, the prototype is either discarded after demonstrating it to the user or it is used in the next iteration of development. We also quite a number of reasons to use prototyping; a few are outlined here:

- Learn by doing
- Clarification of partially known requirements
- Production of expected results

Despite these advantages, there are a number of disadvantages that must be taken into consideration when using such an iterative process. Many of the steps required in good programming practice may be overlooked. Documentation, system testing and efficient programming may be overlooked if the final prototype is deemed acceptable to implement by the project champion. These issues can be avoided by making the link between the waterfall model and prototyping methodologies. This ensures that the design is well documented and, as Southtech Limited are a small organization, provides all parties with necessary feedback early on in the development cycle.

Summary

Many of the development methodologies and project management techniques above have been designed for large scale projects that require teams of developers working alongside numerous people with the student. In such a scenario it is clear to see that meticulous planning is essential to the success of the project.

However, these may not form the best approach for a small project. By taking strengths from a wide selection of methods, the project should be made a success as long as care is taken throughout to ensure that the work is on schedule and that planning is communicated effectively. The decision has been made to use marry the best features of the waterfall model with evolutionary prototyping to form the implementation and testing stages of the project. It is hoped that in doing so the both the advantages are gleaned from the use of prototyping, but the disadvantages are avoided.

CHAPTER 3

System Requirements

3.1 Requirements Analysis

In systems engineering and software engineering, requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Conceptually, requirements analysis includes three types of activities:

1. Eliciting requirements: (e.g. the project charter or definition), business process documentation, and stakeholder interviews. This is sometimes also called requirements gathering or requirements discovery.
2. Analyzing requirements: determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.
3. Recording requirements: Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases, user stories, process specifications and a variety of models including data models.

Requirements analysis can be a long and tiring process during which many delicate psychological skills are involved. Large systems may confront analysts with hundreds or thousands of system requirements. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the student. These may include the development of scenarios (represented as user stories in agile methods), the identification of use cases, the use of workplace observation or ethnography, holding interviews, or focus

system that can be demonstrated to stakeholders. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced. Requirements quality can be improved through these and other methods.

- Visualization. Using tools that promote better understanding of the desired end-product such as visualization and simulation.
- Consistent use of templates. Producing a consistent set of models and templates to document the requirements.
- Documenting dependencies. Documenting dependencies and interrelationships among requirements, as well as any assumptions and congregations.

3.2. Requirements Capture

Grasping a full understanding of the students' requirements is fundamental to the success of the pr

object. There are a number of approaches for obtaining the users requirements and each of these will be explored in order to acquire the broadest and deepest appreciation of what must be achieved in the system. The importance of requirements analysis isn't just for functionality. Without it, there would be no way of evaluating the system, as there would be nothing to evaluate against. There are five commonly regarded methods of gathering the requirements of a new information system that may be relevant to the project in hand.

Research

Many projects are undertaken by a third party that have limited understanding of the organization they are about to carry out the project for. In many cases it is the first opportunity the analyst has to gauge an understanding of the business activities, processes and practices that may go on within the university. There is no necessity for research or background reading into Southtech Limited because of the twelvemonths the author spent there as a support technician gaining understanding and experience of the procedures within the university.

Requirements Analysis

The output from the requirements capture must be analyzed to obtain a complete list of functional and non-functional requirements that can then be used to design and implement the system. This analysis forms a fundamental part of the system design stage in building on the requirements captured in the previous section and creating a comprehensive appreciation of what must be achieved.

Use Cases

One of the main uses of the interviews that were held at Southtech Limited has been to create Use Case. Diagrams that provide a high-level model of what should be achieved in the system. A Use Case diagram provides not only a simple way of communicating ideas with users but also, when complete, produces a high level understanding of what must be achieved in the system. The Use Cases were built up over a number of interviews with different people within the organization including stakeholders and a number of Developers and Support Technicians. These results were then summarized both to create the Use Case diagram but also to elaborate on each use case and generate a comprehensive requirements list.

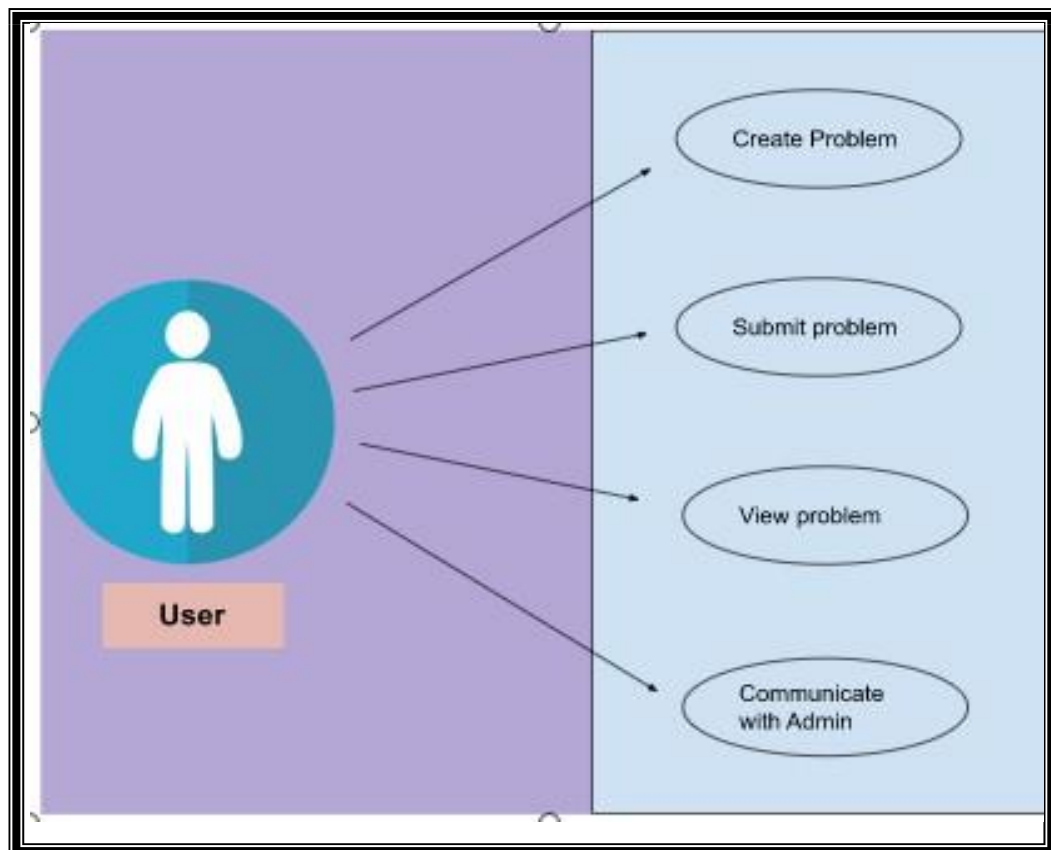


figure: 3.5.1 User Diagram

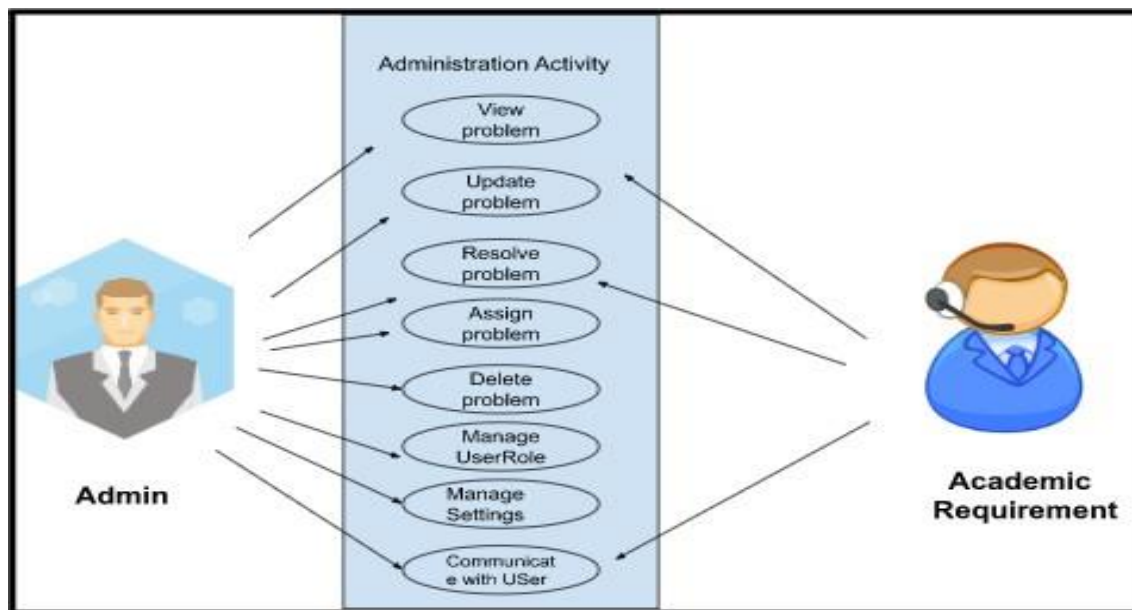


figure: 3.5.2 Use Case Dia

Functional Requirements

Based on the Use Case diagram above, a list of requirements is produced at a lower level of abstraction that the design and implementation incorporate the appropriate functionality. One of the minimum requirements for the project is to produce a suitable prototype call logging system with the ability to track cases and create

reminders for support staff when deadlines approach in line with the students Service Level Agreement. One question that must now be raised is what is deemed to be suitable for a prototype system. The decision taken in Chapter 2 to use hi-fidelity, evolutionary prototyping must be used, to some degree, as a guide. The following list of requirements outlines exactly what is considered necessary in the system, and whether or not the function forms one of the minimum requirements of the project.

Non-Functional Requirements

The non-functional requirements for the system form a set of the main desires of the users interviewed. It is wished that the system currently in place is improved upon in terms of the user interface. The security aspects must be kept simple by relying on the user credentials entered when the user first logs into Windows and, finally, the system must be easy to manage from a technical point of view.

Though these will not be explicitly dealt with in terms of functional requirements, the design must be based around these concepts –they are no less important than the functional requirements.

Best Practice Guidelines

The IT Infrastructure Library (ITIL) is a set of best practice guidelines that has been widely accepted as the industry de facto standard. The ITIL standards break down service requests into several distinct areas:

- Incident Management
- Problem Management
- Configuration Management
- Change Management
- Release management

Understanding how the system should incorporate these areas is important before the design stage is entered. This section concentrates on the relevance of each of these areas and considers their inclusion in the new system. Whilst these are arrived at from best practice guidelines, they may not be appropriate for the kind of support that Southtech Limited provide or may be out of the scope of what the system should deal with.

CHAPTER 4

System Analysis

Many software projects have failed due to an incomplete or inaccurate analysis process, especially system analysis. Technical Analysis is a key step while developing a software application. It can be useful to-

- Confirm with the student that we have gathered all business requirements accurately, and begin with designing and building the application after approval from the student.
- It can be used by the Designers and Developers as a reference when building the application.
- It can be used by the student to verify that the final application actually matches what was initially agreed upon.

Components of System Analysis

There are two parts to Technical Analysis – drafting an Application Specification Document and generating Use Cases. An Application Specification Document is usually derived from the Requirements Documentation from a Business Analyst. This document briefly specifies various features of the application, details parameters of how the application will be built, etc. A sample structure of an Application Specification Document for a typical Web Development project would be as follows:

1. Introduction

- Background
- Purpose
- Scope
- Definitions, Acronyms and Abbreviations
- References
- Overview

2. Overall Description

- Use-Case Model Survey

- Assumptions and Dependencies

3. Specific Requirements

4. Non-functional Requirements

- Browser Compatibility
- Layout
- Graphics and Web Design
- Resolution
- Log History Analysis
- Cookies
- SMTP Server
- Secure Sockets Layer
- Accessibility
- Performance

User Interface

The issue of usability is an important issue that must be addressed during the design of the user interface. Though the system is not intensively used, it is still important to try to reduce the time it takes to carry out a task. While it is perceivable that cutting the time it takes to carry out a task by fractions of a second may save certain companies, employing hundreds of people carrying out the same task, a lot of money, it simply won't be the case at Southtech Limited. More importantly for the university is the need for the interface to be designed so that learning the system is simple and carrying out day-to-day tasks is more efficient than the current system. Many web based systems fail to combat basic usability issues and end up being less efficient than their predecessors.

It is believed that by looking at current methods and theories in Human-Computer Interaction (HCI) and by evaluating the user interface on the current system, a more effective interface can be achieved in the new system.

Getting Human Computer Interaction (HCI) issues solved in systems is not an easy task to carry out and requires knowledge from many fields of science. The ideal designer of an interactive system would have expertise in a range of topics: psychology and cognitive science to give her knowledge of the user's perceptual, cognitive and problem-solving skills;

ergonomics for the user's physical capabilities; sociology to help her understand the wider context of the interaction; computer science and engineering to be able to build the necessary technology; business to be able to market it; graphic design to produce an effective interface presentation. Clearly, an Information Systems degree program is not going to cover every area that is preferred in HCI but it is possible to pay careful attention to the design of the system to increase, as much as possible, the usability to its users. This will be the focus of this section.

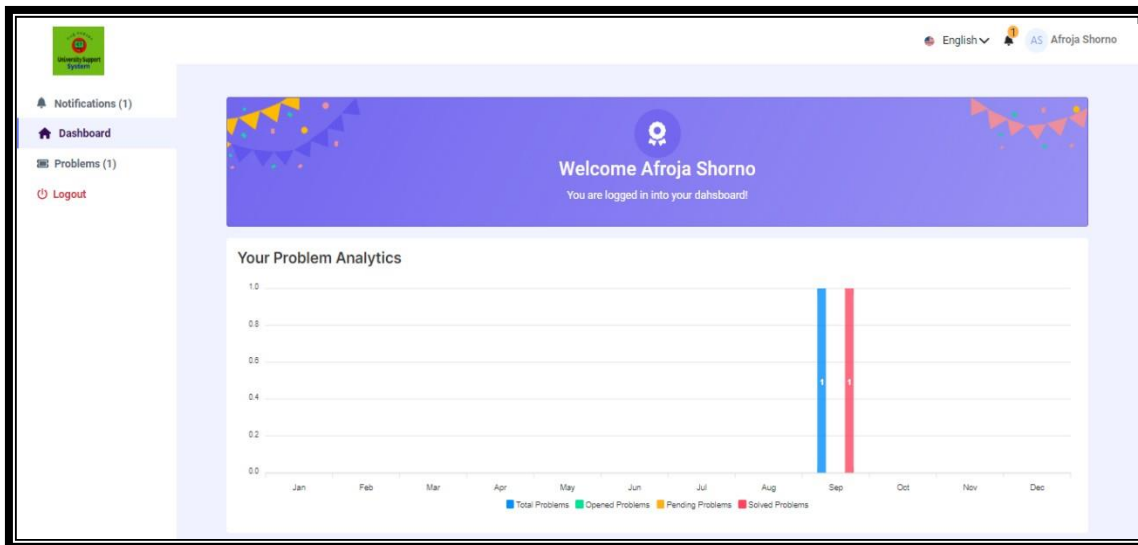


figure: 4.2.1 User Dashboard Panel

Task Identification

There are number of principles in interface design. We need to identify and think carefully about the tasks that the user must carry out. This has been achieved in the requirement analysis above. However, we have to go further by monitoring the frequency at which users carry out the tasks. They also explain how tasks with a high frequency will benefit by assigning shortcut keys to them thus aiding the user in quick navigation. Although no formal frequency observation has been carried out, it is clear, due to the limited number of tasks they carry out, that the administrators' task of managing a support call will be most frequent, with student, contract and system management being used much less frequently. In terms of the support technicians and administrator, because they only have one high-level task to carry out, this form of analysis is not relevant and breaking these functions into smaller tasks is not feasible beyond those tasks discussed in the requirements analysis.

Short Term Memory

The short-term memory of a human is used to remember details to carry out everyday tasks. For Example, a simple multiplication may be divided up into separate blocks, each block requiring its short-term storage in memory. The capacity of this memory is very limited, and people can usually remember facts. It is important to note that a system should not rely on the user's short-term memory very much at all. For example, they should not have to remember information from one page to complete the next. However, it is important that a page does not become so cluttered with information that the user cannot possibly store enough of the information in short term memory to be able to realistically complete their task.

Execution-Evaluation Cycle

This is a model of user behavior designed around the execution of a command on a computer system. If the user moves the mouse, the system then moves the cursor on the screen, and the user then evaluates the response the system had to their command. There are several stages in this process which starts off by the user establishing their goal. The user then forms an intention to carry out actions and then decides on the actual actions they are going to take. The user then executes those actions and perceives the changes they have on the system. They then interpret and evaluate the system state after the action has been carried out. This shows that user inputs should be intuitive and that the outputs should not give unexpected results. It is important to ensure that the system adheres to common standards that users have grown to expect and to ensure that it does behave in a manner that the particular users at Southtech Limited expect.

Form Filling

One area of the system where usability will be fundamentally important is the data entry of new support calls. Often the calls will come in on the phone and so the page must be easy to navigate and to enter data into, possibly using just one hand. The literature read on this subject has not elaborated on how one should make a form usable. Most form filling interfaces allow easy movement around the form and allow some fields to be left blank. They also require correction facilities, as users may change their minds or make a mistake about the value that belongs in each field. However, we must go further if we are to pay real

attention to the data entry form. As mentioned, the form must allow for easy movement around it. In theory it should be as easy as moving a pen to the relevant area on a paper form. Though this may not be possible in practice, simply ensuring that the 'Tab' key results in the cursor being moved to the next field and not some random space is one step towards a usable system.

Menu's

A system that has many different parts will, by definition, require some form of navigation to those parts. Because menus rely on recognition, rather than recall, it is important to ensure that they are meaningful and by clicking on them have the desired effect. Menus can be hierarchical and as such grouping of tasks can be done. This aids the navigation process and an important factor in the helpdesk system will be to ensure that, where applicable, suitable measures are taken to make hierarchical menu structure effective rather than hinder the user.

System Options

Having now discussed the environment within which the new system must work, it is possible to examine the different development technologies available.

Student / Server Architecture

The infrastructure at Southtech Limited provides an environment that allows for a student / server architecture. The decision that must be made is whether to implement a thin-student or fat-student architecture. A thin-student architecture ensures that all of the business logic is controlled and the information is received from the server is in its final state and as such has no further processing to do before it displays the information. A good example of a thin-student architecture is that of a webpage where processing may be done on the server to obtain data to be displayed, but the student receives only the information in its final state – the HTML. In contrast, with a fat-student, much of the processing of data is done on the student's computer and as such, requires more powerful machines. However, the latter results in much of the load being taken away from the server. There are many examples of fat-student applications, but perhaps the most relevant example is the helpdesk system currently employed by Southtech Limited. This is a Windows application that must be installed onto each of the individual machines that require access to the system.

There are a number of issues encountered with this architecture. Installation of the application is nontrivial; it takes approximately ten minutes to install onto each machine and requires manual ODBC configuration. Any updates made to the application also result in a requirement to redeploy throughout the university – far from ideal. Whilst the current architecture takes load away from the server, Muchmore data will be transferred between the student and the server than with a thin-student. Consequently, it is felt that moving to a thin-student architecture would lead to a number of benefits that must be exploited in the new system.

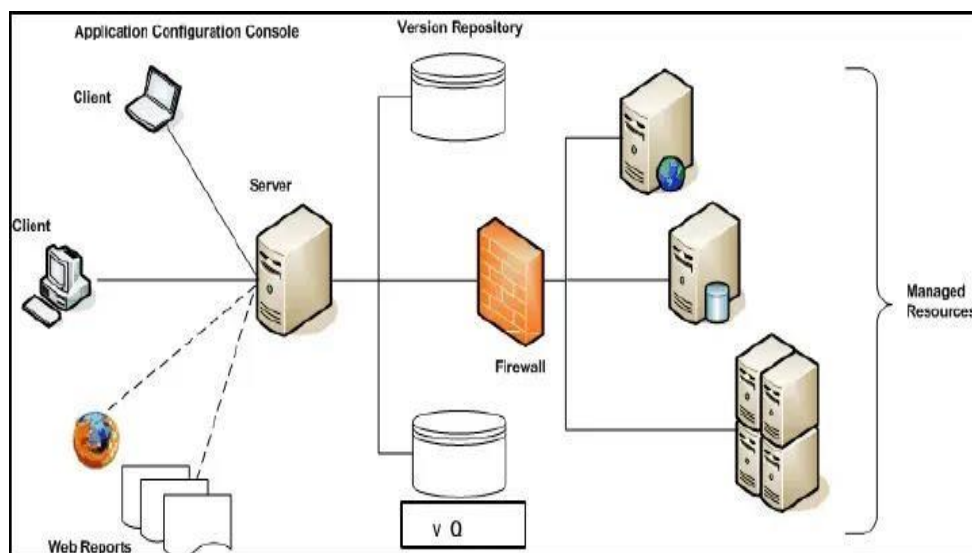


Figure 4.9: Current and proposed architectures

All the servers within the university are connected via a Gigabit network; workstations only benefit from tenth of this speed. In recognition of this it is advantageous to have most data transferred between the servers and not the workstations (Figure 5.3). By transferring only the presentation of the application, the bandwidth restrictions between the server and the workstation are less of an issue. It is also useful to be aware that the power provided by a dedicated server is more appropriate use of processor power than relying on a PC tasked with doing a number of jobs.

This leads to two options. Either the new helpdesk system is developed using web based technology or an architecture is sought after through the deployment of technology such as the Citrix Meta Frame Server. As Southtech Limited do not have the infrastructure in place to support this style of thin-student architecture, a student-server approach will be adopted in the form of a locally hosted Intranet site using the existing infrastructure.

Web Technologies

Since the birth of the World Wide Web technology has moved on considerably from static HTML pages to dynamic, data-driven solutions that have the capacity to support entire business processes. Having an understanding of the technologies available and their limitations is fundamental to the correct implementation of the help desk system. Three technologies are to be discussed: PHP, Active Server Pages (ASP) and PHP. It must be decided if, and how, each of them would fit into the infrastructure Adsotech Limited and an understanding of how the technology would integrate with the requirements outlined above must be gained.

Active Server Pages

Much like PHP, Active Server Pages (ASP) offers a server side scripting environment but it is possible to write pages using a number of different programming languages. Again, like PHP, ASP development provides a server-side scripting environment in which pages are rendered 'on-the fly' to return dynamic, data-driven content in an HTML page. PHP may be a more appropriate and advanced technology for application development.

PHP

PHP is a server-side scripting language that provides database access, form processing, user authentication and many other tasks. In short, it has the capabilities for developing the help desk system with all of the functionality described in the requirements' analysis. Often it is used in conjunction with the Apache web server that is run on approximately 69% of the Worlds Web servers. It will also run on both Microsoft Windows and on the Linux platform. However, Southtech Limited use both Apache and have IIS enabled on their internal facing Intranet Server. It is possible to install PHP on an IIS installation but no-one in the university would be able to support this as well as the Microsoft technologies they have already adopted. Despite the fact the PHP would provide all the functionality required in the new system, it would be unwise to endorse its use in this environment. Trying to incorporate existing Microsoft application data into a PHP run

Web site would require starting from scratch with a great deal of headaches including purchasing new programs.

This, coupled with the issue of no direct support for Windows, leads to an alternative that offers similar functionality but integrates more transparently into the infrastructure to be sought after. On the face of it, PHP appears to be a suitable environment for the helpdesk system. However, Microsoft's latest web development tool.

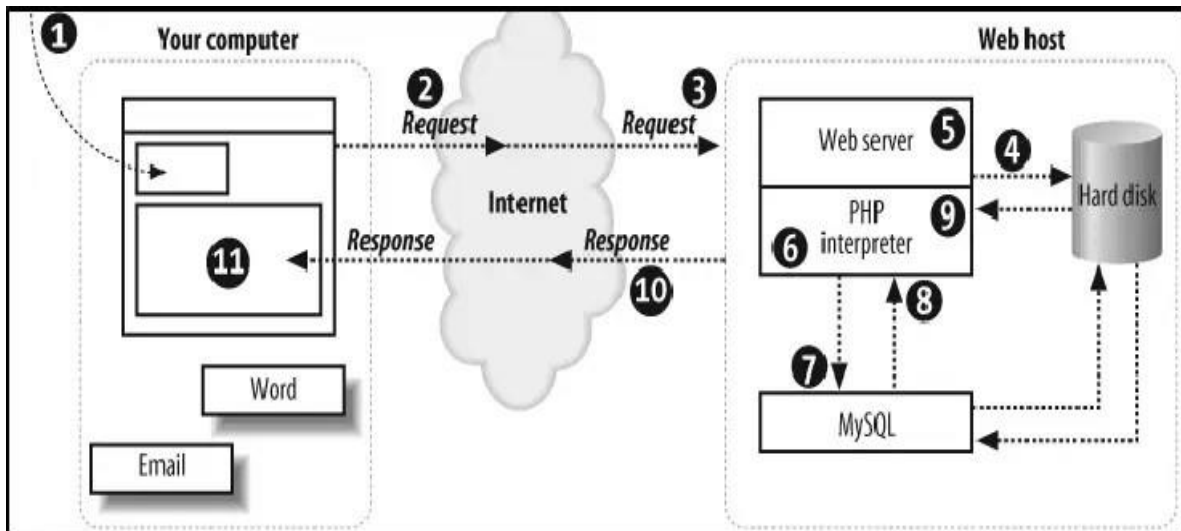


Figure 4.12: The execution model for Web architecture

Summary

Throughout this section a number of options have been considered and the decision to embrace the PHP& MySQL platform has been taken. Although this is not the preferred method of Southtech Limited, because of their expertise in the area and their ability to support it is goes to Microsoft technologies, the decision has been taken on the merits of the technology itself as it is a smaller part of our Tiny Coders group. This technology provides the most advanced functionality with better flexibility from the alternative approaches discussed. The choice of technology has not been a trivial one. It is accepted that whichever tool was chosen there would be somewhat of a learning curve during the implementation. However, whilst the author has experienced both ASP and PHP, very little experience of PHP has been encountered. It would, however, be naïve to think that the areas of experience will help with this project. It is clear that whichever method chosen would result in a learning curve but the choice has been made because of the qualities the technology brings to the project not because of an understanding of syntax– this can be easily learned.

CHAPTER 5

System Design

It has already been decided that an evolutionary prototyping approach is to be taken, so why concentration the design of the system? The answer to this is simple, without a suitable system design the issues discussed in Section 2.2 will not be overcome. Whilst the prototyping method is considered the best approach for this project, it is also felt that documentation and efficient programming practices must be realized. As a result, the analysis from the previous chapter will aid the design of the system before any implementation is to take place.

5.1 System Block Diagram

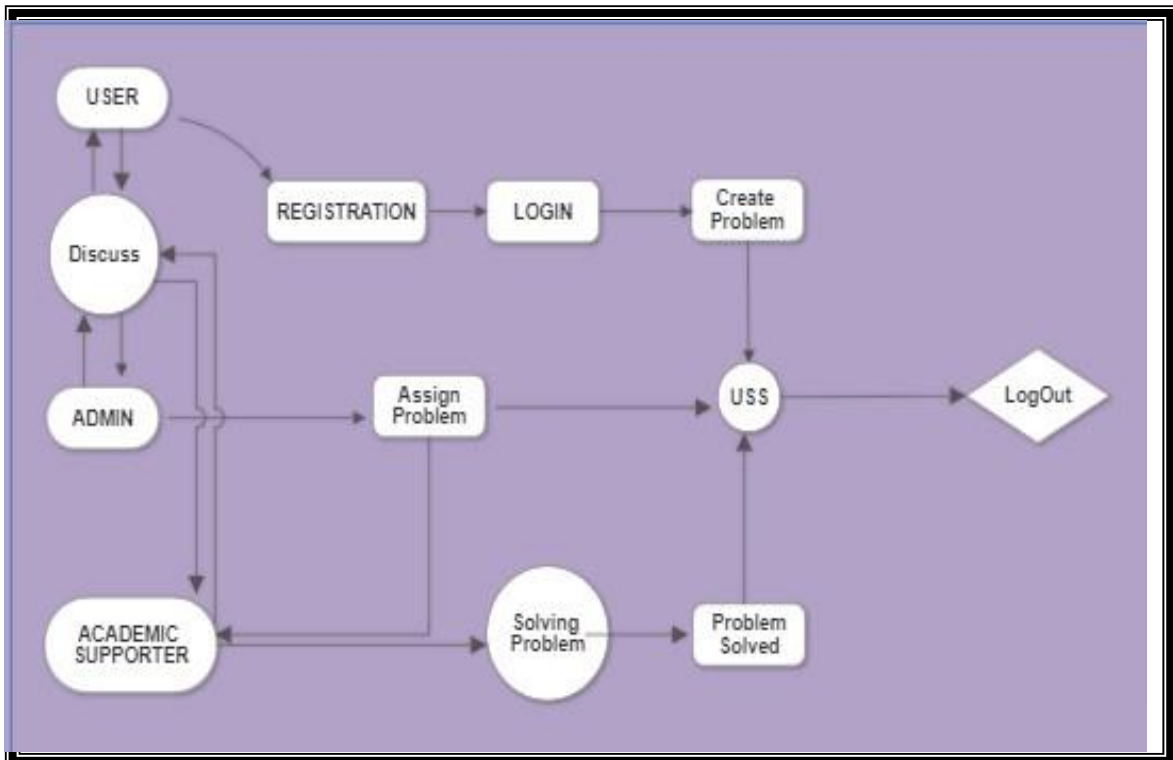


Figure 5.2.1: System Block Diagram

5.2.1 Application Architecture

The architecture of the application is designed to fit into the infrastructure at Southtech Limited which leads to a distributed n-tier application. This is defined as a model that helps developers to create flexible and reusable applications by breaking it up into different tiers. As a result, it is likely that if changes are made to a single tier, it is possible that the entire application will not need updating when those changes are made. Taking the n-tier path also leads to a number of benefits created by the integration of the internal infrastructure. The helpdesk application itself will not deal with the security aspects – Active Directory will handle this. The solution will use the current MySQL Server and will utilize email facilities provided by the Exchange server (see Figure 6.1 for the architecture summary or Appendix M for entire architecture). Finally, the student communicates with just one server, which brings together each tier in a single point of entry for the student and references other tiers when they are required.

Though the system architecture is reasonably complex it leads to a number of benefits for Southtech Limited that will simplify the management of the new system. Because IIS will not have to deal with security, password management remains centralized in the organization, and a single logon is retained. By including a tier for dealing with email the system will be capable of fulfilling one of the minimum requirements of sending email when a call is close to breaking its SLA.

Finally, by keeping data separate from the application it provides a much more flexible framework. The application could be easily designed without having to redesign the data structures. Likewise, in theory at least, the data storage mechanism could be altered without having to make changes to the actual application. In addition, unless the application itself changes, significant improvements could be made to the implementation in the future without the users of the system being aware of the details of these changes (for example, that database could be moved to a faster server; users would not know the details of the change apart from the fact that their queries had been executed in less time).

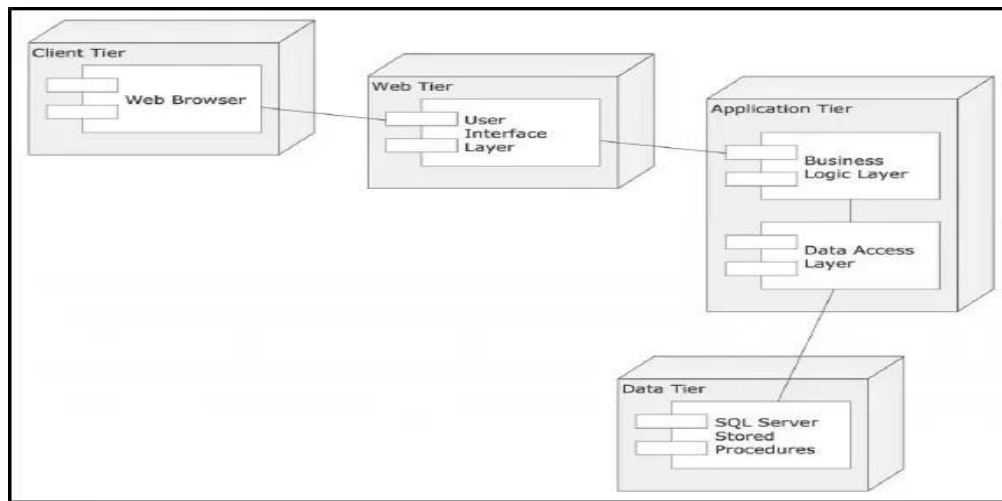


Figure 5.1: Application Architecture - UML Deployment diagram

Database Modelling and Design

A sound database design will be the key to the success of this system. By ensuring that, at this stage, all the relevant information is incorporated into the system, the rest of the system design will fall easily around the database. Database design is generally thought to involve the modeling of different Entities, Relationships and Attributes. Breaking down the design of the database is broken down into different stages:

- Conceptual database design
- Logical database design
- Physical database design

The conceptual design stage is used to build an understanding of each of the entities, relationships and attributes that have been identified. This is then translated to form a logical design by creating valid relations. The physical design must then be created and will be dependent on the Database Management System in use.

Conceptual Database Design

The Entity-Relationship (ER) diagram allows the database designer to get a clear picture of how different entities relate to one-another. Requirements analysis is the most important step (step I) of the database life cycle and is typically the most intensive and as such the previous analysis is extremely important.

In order to determine the relevant entities, relationships and attributes, the internal documentation is the most useful (see Appendix D and Appendix E). These documents outline the entire call logging process and also give an insight into the data that has to be captured with each call.

Logical Database Design

With a conceptual model completed, the logical database design which incorporates the tasks of deriving relations, validating the relations using normalization and ensuring that all constraints are checked. From the conceptual model (the E-R diagram) Connolly and Beg state that the following must be carried out:

- For each strong entity in the model, create a relation that includes each attribute
- For one-to-many relationships a copy of the primary key attributes is passed from the parent to the child relation as a foreign key.

In addition to this, some of the attributes associated with the ‘Support call’ entity may themselves need to be promoted to entities. The ‘Action’ attribute is multi-valued and requires promotion to an entity with the ‘Supporter’ primary key attribute of the ‘Support Call’ entity passed as a foreign key attribute. The ‘Problem category’, ‘Status’ and ‘Priority’ attributes must contain values held within an attribute domain which requires some form of integrity constraint. There are a number of ways that this can be accomplished. Firstly, within the DBMS, in this case it will be Microsoft MYSQL Server, check constraints can be utilized to ensure that the values entered are from the required domain.

Another method is to promote the attribute to an entity in the design and pass the primary key of the parent entity into the new entity as a foreign key attribute. For the helpdesk system this method provides significant advantage over check constraints. Because the constraints are stored in a lookup table, the onus to maintain these constraints can be removed from the developer and passed onto the user. This will create an environment in which the system can be updated dynamically instead of requiring the skills of a database programmer. However, by choosing this method, the database design is made more complex.

Data Dictionary/Database Files

A Data Dictionary is simply a record of data about data. It may be manually compiled, or it may be a fully automated package. All definitions of elements in the system— data flows, processes, and data stores—

are stored in Data Dictionary. Data Dictionary is an integral component of structured analysis. Data Dictionary provides additional information about the system. A Data Dictionary is a catalog- a repository --- of the elements in a system. These elements center on data and the way they are structured to meet user requirements and organization needs. The major elements are data flows, data stores and processes. The data Dictionary stores details and descriptions of these elements The Data Dictionary is the only common source of definitions for users and investigators alive. It is the single source of answer of answers to all questions regarding the format and context of the data sets used in the system.

Data Elements

The most fundamental level of data is the data element. Data elements are building blocks for all other data in the system like Data Names, Data Description, Aliases, Length, and Data Values.

Functional Design

Modelling of the data structures has specified exactly what information is stored in the new system. This section details the processes that must be undertaken in order to store that information. It is important to get a good understanding of how processes interact with one another but because prototypes are being utilized, the application design should be considered only as a guide.

System Maintenance

The system maintenance screen will simply allow the core functionality of the system to be maintained and configured. The global settings such as default SLA response times, default priorities will be configured here as well as adding and removing authorization for certain users' networks.

Update Problem
✕

ID

Status

Priority

Department

Assign (If do not have Agent, then create an agent) | Create Agent

Expire Date

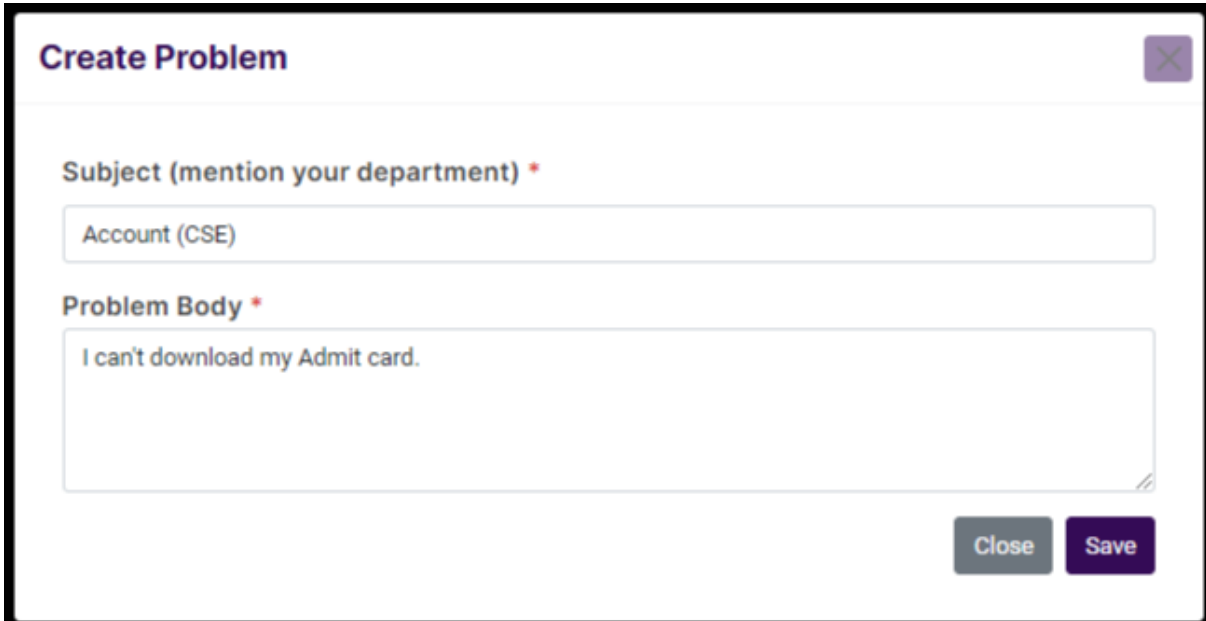
Figure:5.8 Update Problem

User Interface

With the aim of ensuring an acceptable user interface that addresses the HCI issues discussed in Section 5.1 a page template has been developed using the popular CSS and JavaScript.

This design has not included the use of any HTML mark-up but instead simply positions components on the page where they are considered to be most suitable. Figure 6.9 outlines the main page template and where the page details are a little more complex, further images have been created to aid the implementation of the pages. The page template is divided into five distinctive areas, the header, menu bar, sub menu, page options menu and the main content.

The header and the menu bar will remain the same throughout the site and will not change. The sub-menu and page-options sections will be dependent upon the selected page and the content will clearly be the content of the page selected.



The screenshot shows a web form titled "Create Problem". At the top left is the title "Create Problem" and at the top right is a close button (an 'X' in a square). Below the title is a section labeled "Subject (mention your department) *" with a text input field containing the text "Account (CSE)". Below that is a section labeled "Problem Body *" with a text area containing the text "I can't download my Admit card.". At the bottom right of the form are two buttons: "Close" and "Save".

Figure:5.9 User Problem creation

System Implementation

Development Environment

Before the implementation of the system, and in addition to the minimum requirements of the project, a development environment has been created, instead of having to use the internal infrastructure at Southtech Limited. There are several reasons behind this decision. Firstly, if the infrastructure at Southtech Limited was utilized either the development would have to take place in the office or a connection through their VPN would have to be made. This would not be satisfactory because the bandwidth of the connection restricts what can be done and it would not be practical to visit the office on a daily basis. The second reason for this decision was that in implementing the system, it is not necessarily known what the impacts on other systems might be. It is extremely dangerous to make changes on business-critical systems unless full testing has been undertaken and this is a risk that the author was not prepared to take.

It is not entirely necessary for the environment at Southtech Limited to be replicated for the development of the first prototype. However, by undertaking this task now as opposed to future iterations of development, it is possible to build a higher fidelity prototype which encompasses more functionality such as email and the integration of Active Directory. This is beneficial for evaluation of the new system because, from as early a stage as possible, users will have an understanding of the functionality that will exist in the final product. Possibly more compelling than this is that by creating the environment that the system will sit as soon as possible ensures that the final product will work within the infrastructure at Southtech Limited without the need for excessive reconfiguration or coding.

The development environment consisted of a Personal Computer of reasonable specification connected to a broad band connection through a firewall and router. In order to configure Active Directory and the email server, Microsoft Exchange, a domain name was required. For the purposes of the project, the author already had an unused domain name, skiint.com, and decided to use it in this environment. In doing so it is, in essence, the equivalent to Southtech Limited using their domain name, southtechgroup.com.

As discussed in Section 6.1 the system relies on many different server components. At Southtech Limited these are generally hosted on different servers, however, in this development environment, they are all packed onto the same machine. Whilst this may have adverse effects on the performance of the system, it is not a release environment and only has one person interacting with it. As such the performance issues are not apparent, although it is not a true reflection of the final use of the application where many users may interact concurrently with the system.

Data Access

The underlying requirement of this system is to store and retrieve information captured at different stages of the helpdesk procedure. In order to achieve this access to the database must be provided to the web front end of the system. Having chosen the technology for this, ADO.NET, an easy way of calling common methods is to create a data class. By including the class in the namespace of the application, it is possible to call data access functions at any time which significantly decreases the development time. And code duplication. The access to the database enables MySQL queries to be executed and a Dataset returned to the

application. Stored procedures are used in MySQL server so that query strings are not required in the code. This creates two advantages. Firstly, less data is transferred between the student and the server so the execution is more efficient. Secondly, the detail of the query is hidden from the application as so another layer of abstraction is formed. This deskills the MySQL required but ensures that any developer can still work with the database.

Page Template

he implementation of the system requires the page design from Section 6.4 to be translated from a simple two-dimensional image into a fully rendered PHP page. All the images are sliced into relevant sizes and placed into a generic HTML page which is then used as the basic template every time a New php file is required. The newly created .php file can then have its main content inserted within the development environment depending on which page is being implemented. The Object Orientated approach of PHP allows for inheritance of classes that lends itself to the creation of these page templates. Figure 7.2 illustrates how the page template structure works. As was previously mentioned, all standard pages in a PHP application inherit methods and attributes from the Page (System.Web.UI.Page) class. Instead of the pages of this application inheriting from the Page class, they inherit from the Page Base class that includes additional cross-site functionality.

CHAPTER 6

System Function & Outcome

Before an evaluation of the system can be carried out it is vital that system testing is performed so that the requirements can be measured against the achievements made. For Southtech Limited, testing represents a more vital stage that ensures the development carried out fulfils their requirements and that the system is adept enough to deal with every user's requests. Two forms of assessment must therefore be carried out; firstly, internal validation testing to ensure data integrity and that the software runs without any adverse issues being apparent. Secondly, acceptance testing to ensure that the users are satisfied that the functional and non-functional requirements captured at the beginning of the project have been implemented appropriately.

Functional Testing

The sole aim of the functional testing carried out is to ensure that the user cannot 'break' the system by entering erroneous data. Creating a list of every function implemented in the system, and considering the different approaches users may take that could break the system is a good way of achieving this. Appendix P provides a comprehensive list of what is and what is not implemented in the system. Where certain functionality has not been implemented Southtech Limited have agreed that future developments will address this absence. Because of the prototyping approach, coupled with the fact that a second iteration now needs to be developed, it is felt that the tests provide some very positive feedback about the system. One of the most significant outcomes of the functional testing is the uncovering of the fact that field validation needs to be implemented on each of the forms. It was felt through the implementation of the system that, because of the amount of work required to incorporate the functionality requested, time would be better spent implementing the functionality that the users would recognize to be more beneficial to them. As a result, field validation has been left to be more of a tidying up exercise after the system has passed through another iteration of development. The forms implemented successfully provide the functionality required with just a few omissions that do not form the minimum requirements (discussed in the project evaluation).

User Acceptance Testing

At the beginning of this project, after all the requirements capture and analysis had been undertaken, it was necessary for our group to sign-off the requirements list and for them to have an understanding of what would be included in the first prototype. More important than the validation testing in Section 8.1 is the acceptance from the Users that it works as they expect it to. The last column in each of the tables in shows whether or not the implementation has been formally accepted by all the users questioned. This was determined by holding demonstrations with a number of users across each of the functional areas and allowing them to use the system for a number of test cases prepared beforehand.

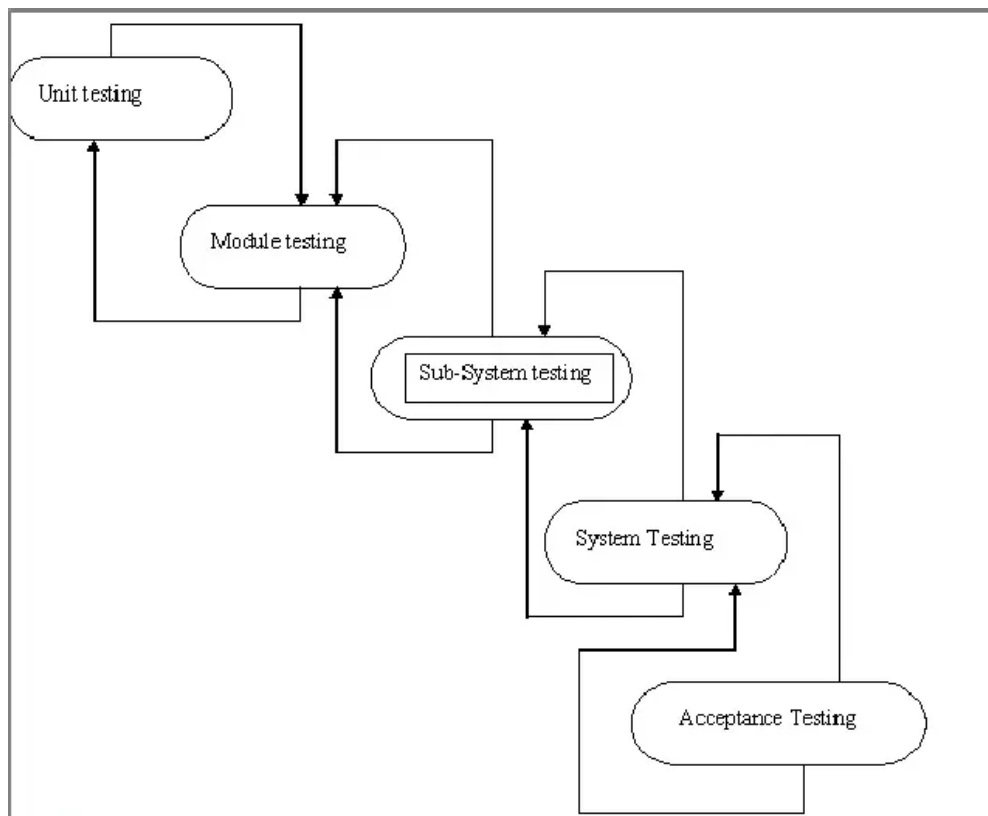


Figure: 6.2 testing

Usability Testing

In response to the poor efforts made in the previous system to address HCI issues, much emphasis has been put on usability throughout this project. There are many defects that can be encountered with usability including the navigation, screen design, terminology, and feedback.

It is hoped that the careful planning and design of the user interface will pay dividends with a system that is easy to learn an intuitive to the user.

However, it is acknowledged that interface design is an iterative process and that addressing the issues of being able to learn the system quickly and creating an intuitive user interface in the first iteration of development is unlikely. The test plan developed uses a task based approach to determine how well the user interacts with the system; both with navigability and their comprehension of responses they receive from the system. The test does not explore every area of the system but ensures that a good cross-section of the functionality is encountered. By measuring the time it takes to complete a task, comparisons may be made to the old system. However, by running the tests more than once, it is possible to make judgments on how easy the system is to learn. Accordingly, the tests were repeated three time. The tasks performed were:

1. Change the SLA for a specified student from the default agreement to a customized agreement
2. Add a specified employee to a specified student

The testing was performed with a sample of four Administration employees as it is they who require the most functionality and use the system most frequently. It is also arguable that they do not have as deepen understanding with computer systems as the support technicians and as such provide a better sample for testing the learning aspects of the system.

| ID | Name | Assignee | Department | Subjects | Status | Priority | Created Date | Message | Actions |
|----|---------------|---------------|------------|-----------------|---------|----------|--------------|---------|---------|
| #5 | NASRIN NIJHUM | Not appointee | NULL | IT Problem(CSE) | Pending | NULL | 05-Aug-2022 | 0 | ⋮ |
| #4 | NASRIN NIJHUM | Agent | BBA | Retake (BBA) | Closed | High | 06-Sep-2022 | 0 | ⋮ |
| #3 | AFROJA SHORNO | Agent | Account | Account | Open | High | 06-Sep-2022 | 2 | ⋮ |
| #2 | Customer | Agent | CSE | design issue | Pending | Medium | 03-Sep-2022 | 0 | ⋮ |
| #1 | Customer | Agent | Account | seeder problem | Closed | High | 03-Sep-2022 | 3 | ⋮ |

Figure:6.3 All Problems list

6. 4 System Outcome

Outcome of the data structures has specified exactly what information is stored in the new system. This section details the processes that must be undertaken in order to store that information. It is important to get a good understanding of how processes interact with one another but because prototypes are being utilized, the application design should be considered only as a guide.

Outcome Interface

Log in

In Login Panel user, admin & academic supporter need to log in with there ID & password.

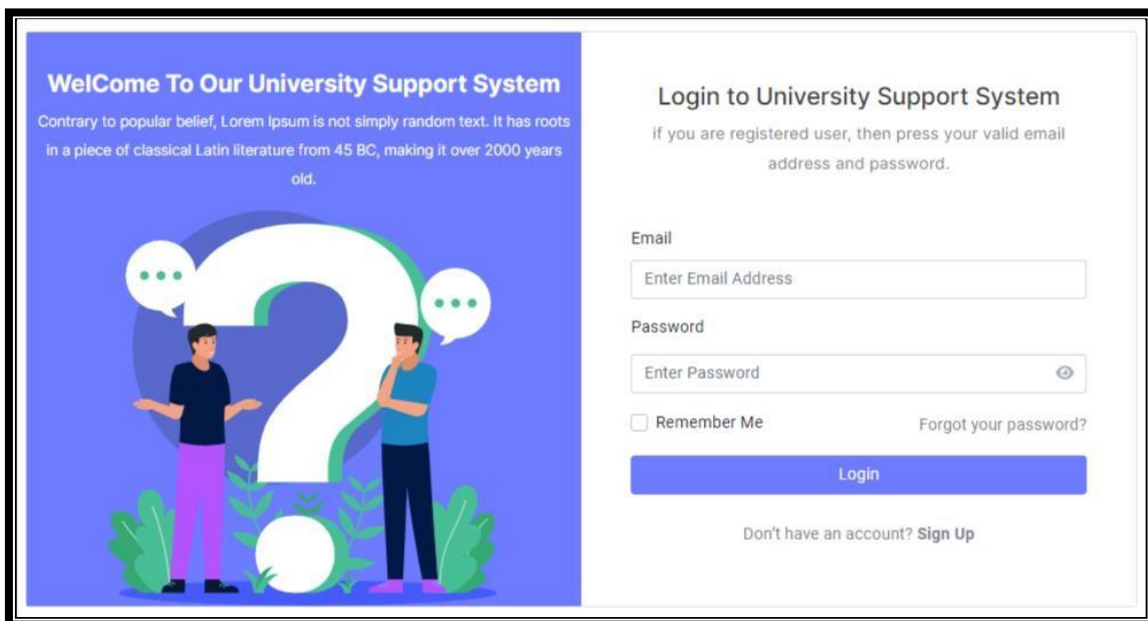


Figure:6.5.1 Login Interface

Registration

Users have to register their Name / ID / Password for log in, Once registration done, students can place their problems & record their problem.

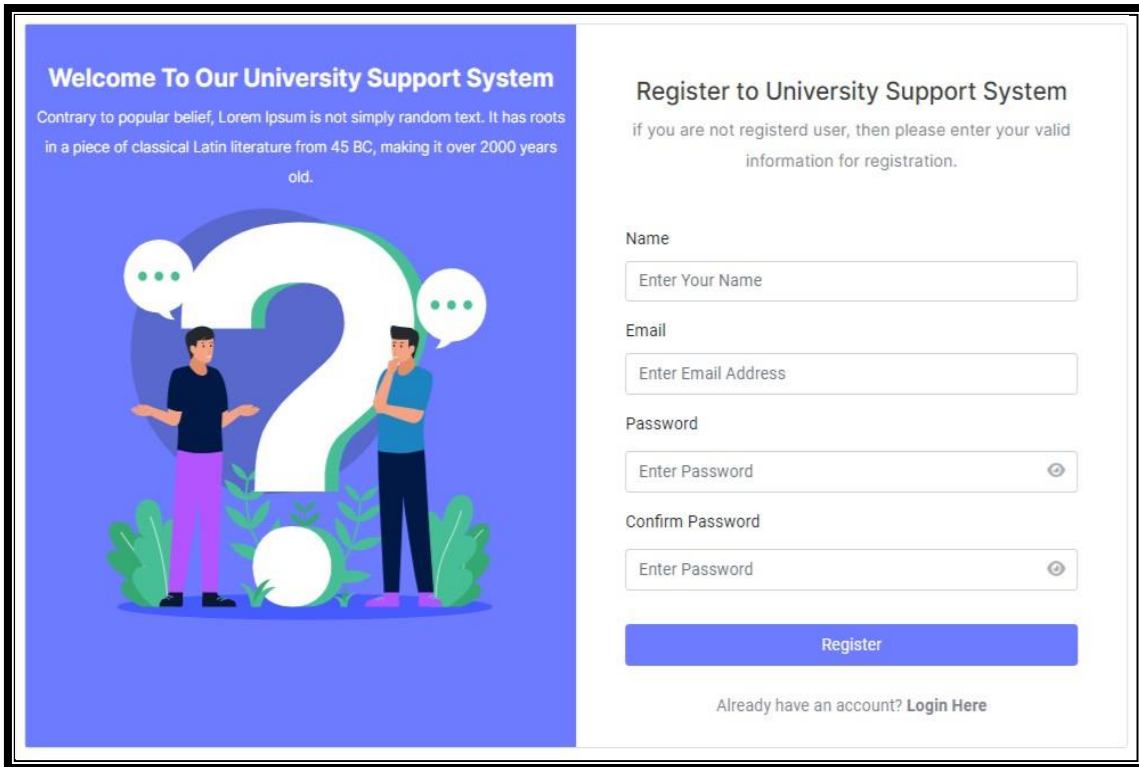


Figure:6.5.2 Registration Panel

User Panel

User can log in there panel by using their name & password, after that they can easily share their problems to our Admin. Admin will assign the correct desk or person as per students requirements.

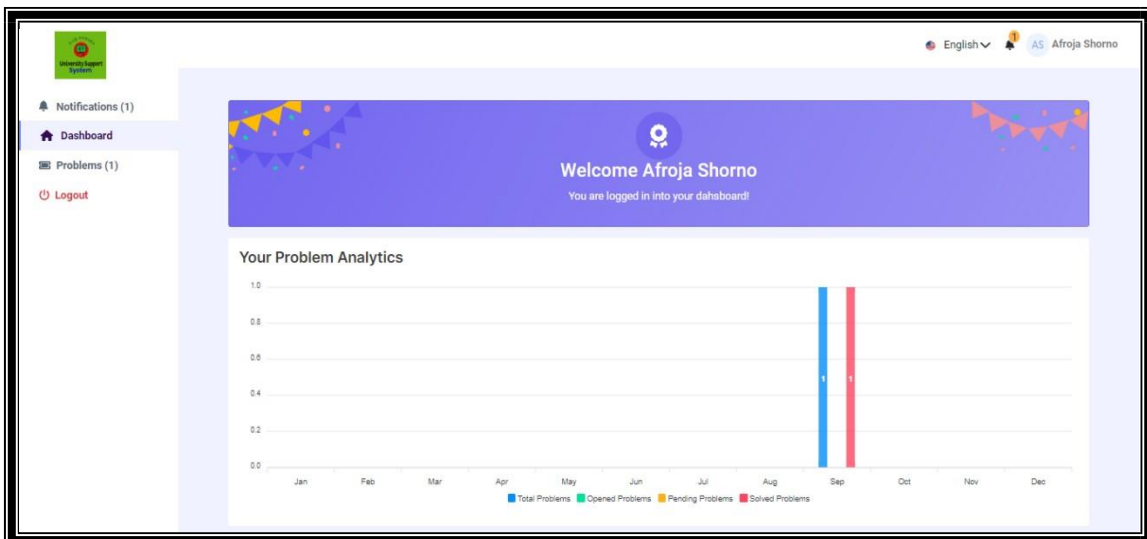


Figure:6.5.3.1 User Dashboard Interface

Create Problem

Subject (mention your department) *

Account (CSE)

Problem Body *

I can't download my Admit card.

Close Save

Figure:6.5.3. 2 User Problem creation

Admin Panel

A Students service administrator holds a Admin, handling students inquiries and overseeing the needs and requirements of the student service representatives working in their department. This sector will be quick response that requires equal parts clerical skill and communications savvy.

Admin Task

- Monitoring
- Assigning
- Tacking
- Set Priority (High, Medium, Low)
- Set Department
- Check notifications

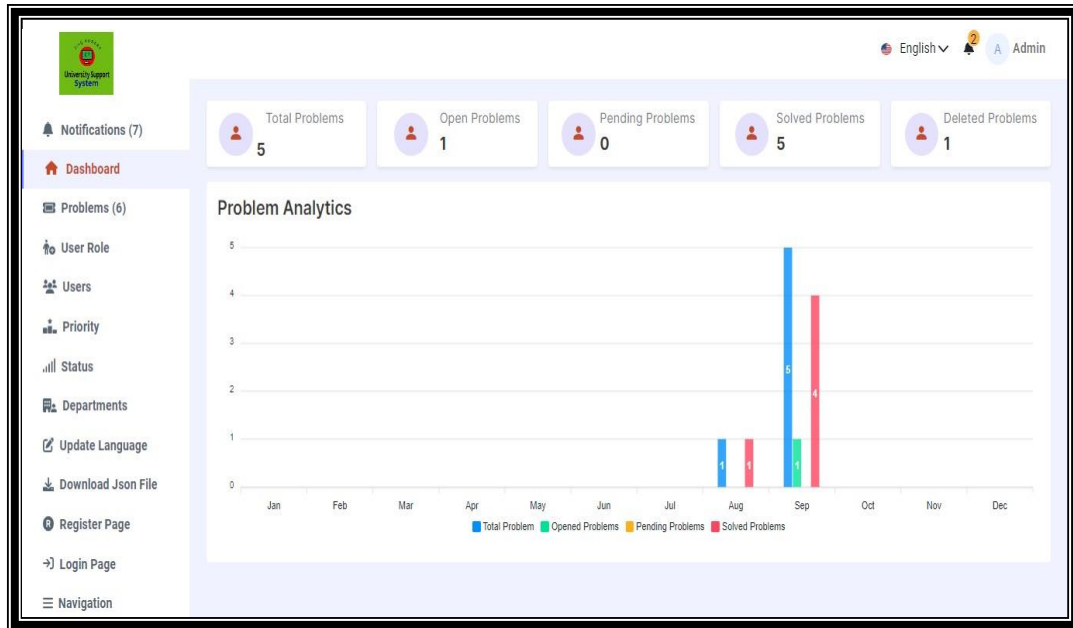


Figure:6.5.6 Admin Homepage



Figure:6.5.7 Admin Notification Panel

Show 10 entries
Create Department

| Serial | Name | Created Date | Action |
|--------|---------|--------------|--------|
| 1 | Account | 03-09-2022 | ⋮ |
| 2 | CSE | 03-09-2022 | ⋮ |
| 3 | EEE | 03-09-2022 | ⋮ |
| 4 | CIVIL | 03-09-2022 | ⋮ |
| 5 | BBA | 03-09-2022 | ⋮ |
| 6 | IT | 03-09-2022 | ⋮ |

Showing 1 to 6 of 6 entries
Previous 1 Next

Figure:6.5.8 Admin Department assigning

< Back

Status
 Solved

Priority: High **Assign**: Agent **Expire Date**: 2022-09-08

Customer Name: NASRIN NIJHUM Department: Account Priority: High ID:#5

Subject: IT Problem(CSE)
Problem: Tomorrow is my final xm, but somehow i ca download my problem, can u help me?

Agent
 Send me your Id , I'll Fix it.
 Wed At 10:44 PM

NASRIN NIJHUM
 My ID is -CSE1801013013
 Wed At 10:48 PM

NASRIN NIJHUM
 Fixed it as soon as possible
 Wed At 10:48 PM

NASRIN NIJHUM
 Fixed it as soon as possible
 Wed At 10:48 PM

Agent
 NOW You can Download your Admit. Your problem has been solved.
 Wed At 10:47 PM

NASRIN NIJHUM
 Thank you.
 Wed At 10:47 PM

Profile
 Designation: Student
 Phone: +Not Found
 Email: nasrinijhum@gmail.com
IT Problem(CSE)
 05 Aug 2022 at 01:33 AM

Figure:6.5.9 Admin Message Panel

Update Problem ✕

ID
#2

Status
Pending

Priority
Medium

Department
CSE

Assign (If do not have Academic Supporter, then create an Academic Supporter) | [Create Academic Supporter](#)

× Academic Supporter

Expire Date
09/08/2022

Close Update

Figure:6.5.9 Admin Assigning Problem

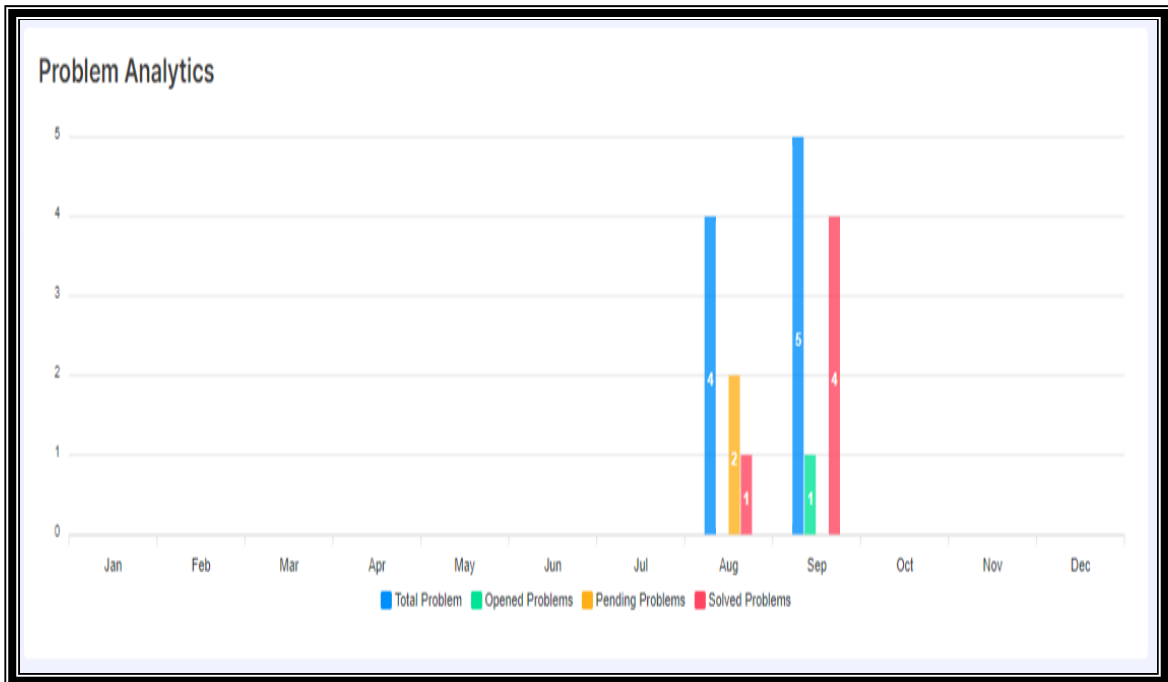


Figure:6.5.10 Admin Problem Analytic chart

Academic Supporter

An academic support work will be start from the assigning by admin. His / Her work will be depending on the problem has come.

Academic Supporter TASK

- Analysis the problem
- Will response for each student
- Clarifying the exact problems
- Open a case & start working as per date priority

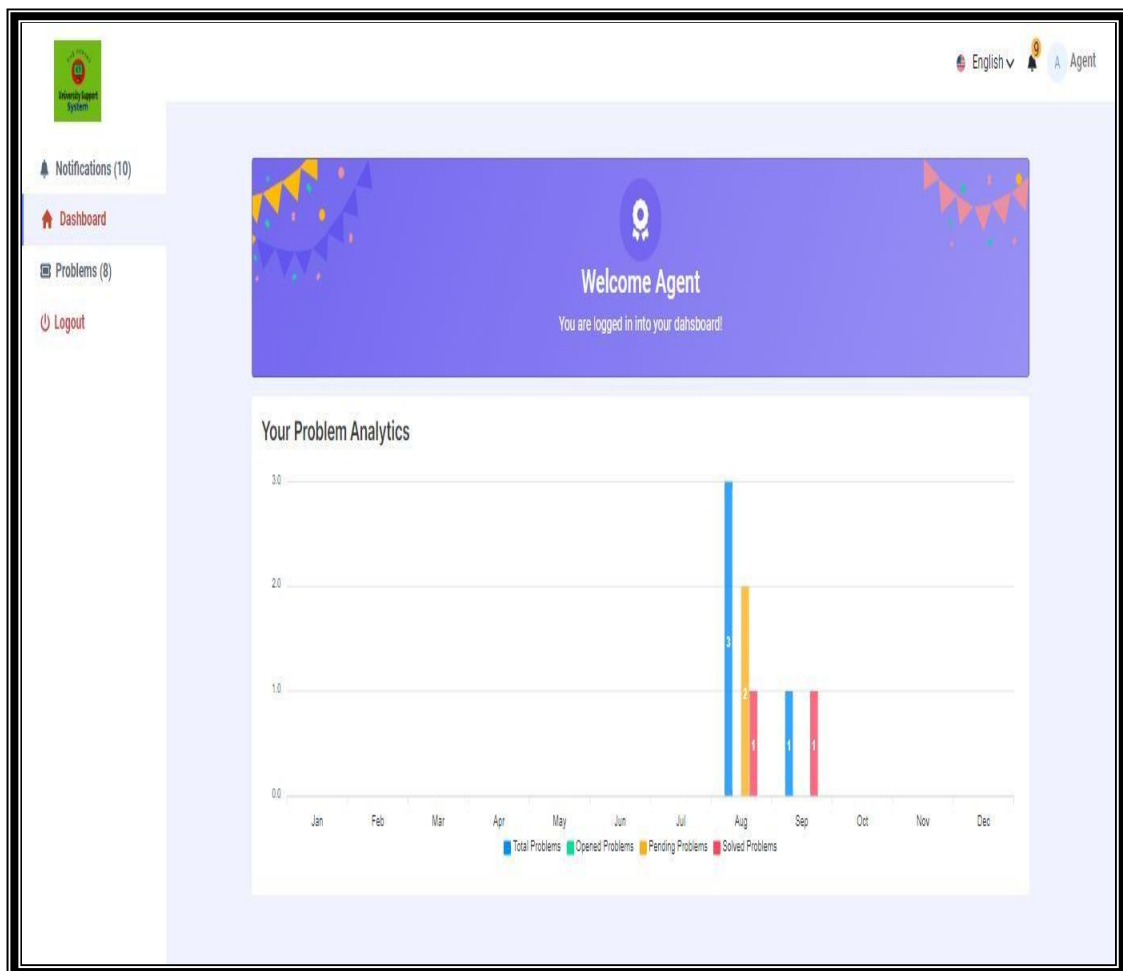


Figure:6.5.7 Academic support Problem Analytic chart

Problem Details

| | |
|--------------|---------------------|
| ID | #2 |
| Name | Customer |
| Department | CSE |
| Subject | design issue |
| Status | Pending |
| Priority | Medium |
| Assign | Academic Supporter, |
| Problem Body | solve this issue |
| Expire Date | NULL |

[Return Back](#) [Send Message](#)

Figure:6.5.8 Academic support Problem Details

CHAPTER 7

CONCLUSION AND FUTURE WORK

Conclusion

In short, a University Support System solution makes things easier for both internal and external student service activities. When students are satisfied, they are likely to stick around longer. Therefore, a University Support System is the ideal platform for inducing student loyalty as it simplifies student service tasks, organizes relevant data, and streamlines multiple processes.

In the long term, a helpdesk platform may also help organizations save money over time since it can prompt more efficient student service operations. Even if it is always possible to improve the ticket support and make it more user friendly. The system is ready to be used and the basic functionality has been achieved. The system reliability is high and enough security has been provided the system is very simple in design and to implement.

Future work

When the project was originally conceived there was a sense that it may be a little over simplistic for final year project. After a small amount of market research opinion was quickly altered as it became apparent how a new system could bring numerous business benefits. Having now developed the first prototype of the system, the future vision has to be discussed in order to understand the authors' excitement with the project and the frustration that more could not be achieved in the time spent so far. The qualitative feedback in the previous chapter provides a good basis to start this section from. The issues raised must be the first improvement to be made to the system. They don't form enhancements but fundamental changes that will provide adequate functionality with a well-established user interface. However, beyond this there are changes that can be truly regarded as 'enhancements'. The final point to make about improvements that should be made to the implementation methods relate to data access. The system, in its first prototype form, would not support concurrent usage because of the data access implementation. Because the Web technology used leads to a disconnected front-end, controlling data becomes much more complex when many users are to use the same data source.

Some Form of record locking needs to be implemented in order to ensure that when one person makes a change no one else makes a change while they are editing the data. This is a complex area of research and will not be discussed any further, though, in the final version it is imperative that some method is implemented. Taking functionality from these items of software, such as the Statistical Dashboard would ensure that the system provided User can get a solution that meets their requirements entirely.

However, other features that have not been seen on the market could be added. Discussed in the best practice guidelines was the inclusion of configuration management. This is something that is seen to be relevant and should really be implemented at a future date. Once this is implemented the system could go a stage further than any other package on the market by incorporating Remote Control Students such as PC Anywhere, VNC or Microsoft Remote Desktop. This would not create just a call tracking package but would result in an entire support management where support technicians could receive a call and instead of having to lookup separate documentation, simply click a button and be connected to the student's system. Finally, it is felt that, although email is a powerful tool, a different method of notifying employees when they have a new call logged to them or when they are close to breaching an SLA. It has been considered that a Java application could be used to notify the users. Another option could be integrating the system with Instant Messaging technology to send messages over the Internet or even via SMS. The possibilities are exciting and endless. The prototype is a distance away from the vision but is a good start to a product that was not available on the open software market.

REFERENCES

- [1] E. Cambria, et al., "New avenues in opinion mining and sentiment Analysis," IEEE Intelligent Systems, vol. 28, no. 2, 2013, pp.15-21.
- [2] E. Cambria, "Affective computing and sentiment Analysis," IEEE Intelligent Systems 31.2 (2016): pp. 102-107.
- [3] P. Anantharam, K. Thirunarayan, and A. Sheath, "Topical anomaly detection from twitter stream," Proceedings of the 4th Annual ACM Web Science Conference, ACM, 2012.
- [4] L. Chen, et al., "Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter," Sixth International AAAI Conference on Weblogs and social media, 2012.
- [5] L. Chen, W. Wang, and A. Sheth, "Are Twitter users equal in predicting elections? A study of user groups in predicting 2012 US Republican Presidential Primaries," International Conference.
- [6] Avinash Navlani, "Understanding Random Forests Classifiers in Python", May 16, 2018, [Online]. Available: <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>
- [7] D. Mediana, "Rancang Bangun Aplikasi Helpdesk (A-Desk) Berbasis Web Menggunakanframework Laravel (Studi Kasus Di Pdam Surya Sembada Kota Surabaya)
Andi Iwan Nurhidayat Abstrak
- [8] P. Jaringan and P. T. Indosatm, "Otomatisasi Trouble Ticket Untuk Peningkatan Performance System," no. March, pp. 51– 58, 2017.
- [9] M. Harcenko, P. Dorogovs, and A. Romanovs, "IT Service Desk Implementation Solutions,"Sci. J. Riga Tech. Univ. Computer. Science., vol. 42, no. 1, pp. 68 – 73, 2010.
- [10] H. H. Thimm, "Supporting Environmental Compliance Managers by Ticket Systems- An Industry Survey, Deployment Optionsand a Reference Business Process," Int. J. Environ. Sustain., vol. 6, no. 1, 2017.