



Faculty of Computer Science and Information Technology

*Benadigital Ticketing System for Issue
Management*

Nelysa Nurain Zaik

Bachelor of Software Engineering with Honors

2025

Benadigital Ticketing System For Issue Management

NELYSA NURAIN ZAIK

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Kejuruteraan Perisian dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

Acknowledgement

First and foremost, I would like to express my deepest gratitude to Allah SWT for granting me the strength and perseverance throughout the completion of this final year project.

I would also like to extend my heartfelt thanks to my supervisor, Madam Earzaqilla Phang, for her continuous guidance, support, and insightful feedback throughout this project. Her encouragement played a crucial role in helping me stay focused and motivated. Additionally, I would like to express my gratitude to Profesor Madya Dr. Sze San Nah, my examiner, for her comments on this project.

A special thanks to my mom, Manis Binti Yusak who is my main support pillar through this ongoing journey and family members who provided endless support and prayers during this journey. I would also like to acknowledge my friends who is a blessing in disguise and have helped me technically and morally during the development of this system. Thank you da gang of Kuikui for lifting the burden on my shoulder and making my heart a little lighter, this journey would not have come to an end without your endless support.

Furthermore, I am also profoundly thankful to my one and only manager, Tuan Colin Nizam and my colleagues, Madam Farhana and Azka Basyiri who serves as my support for the final year project whose words of wisdom and trust in my abilities helped to keep me motivated to finish my final year project.

Lastly, I want to thank me for believing in me, I want to thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting. I wanna thank me for always being a giver and trying to give more than I receive. I wanna thank me for trying to do more right than wrong. I wanna thank me for being me at all times, alongside with uncountable amount of kopi peng and eclipse.

Hence, I would like to appreciate the assistance of everyone involved directly or indirectly in the success of this project.

Abstract

This project report presents the development of the “Benadigital Ticketing System for Issue Management”, a web-based support platform designed to handle technical issues raised by users of the Benadigital CRM system. The system allows users to submit tickets, while administrators and developers can assign, track, and respond to those issues efficiently. The core technologies used include Flask as the backend framework, MySQL for database management, Bootstrap for responsive UI, and PythonAnywhere for cloud deployment.

Additional features such as public ticket submission without login, role selection page (User/Admin/Developer), email notifications to admins, developer workload monitoring, and activity logging have been integrated to enhance the overall functionality. The target users include general users, developers, and system administrators within the Benadigital CRM ecosystem. The system aims to streamline support workflows, improve response times, and centralize issue tracking for better transparency and resolution management.

Abstrak

Laporan projek ini membentangkan pembangunan “Sistem Tiket Benadigital untuk Pengurusan Isu”, iaitu sebuah platform sokongan berasaskan web yang direka untuk mengendalikan isu teknikal yang dibangkitkan oleh pengguna sistem CRM Benadigital. Sistem ini membolehkan pengguna menghantar tiket, manakala pentadbir dan pembangun dapat menetapkan, menjejaki, dan membalas isu tersebut dengan cekap. Teknologi utama yang digunakan termasuk Flask sebagai rangka kerja backend, MySQL untuk pengurusan pangkalan data, Bootstrap untuk antara muka responsif, dan PythonAnywhere untuk pelancaran di awan.

Ciri tambahan seperti penyerahan tiket awam tanpa log masuk, halaman pemilihan peranan (Pengguna/Pentadbir/Pembangun), pemberitahuan emel kepada pentadbir, pemantauan beban kerja pembangun, dan log aktiviti telah diintegrasikan bagi meningkatkan fungsi sistem secara menyeluruh. Pengguna sasaran merangkumi pengguna umum, pembangun, dan pentadbir dalam ekosistem CRM Benadigital. Sistem ini bertujuan untuk mempermudah aliran kerja sokongan, mempercepat masa respons, dan memusatkan penjejakan isu untuk pengurusan penyelesaian yang lebih telus.

Table of Contents

Chapter 1: Introduction	14
1.1 Introduction	14
1.2 Problem Statement	15
1.3 Scope.....	16
1.4 Aims and Objectives	16
1.5 Brief Methodology	17
1. Requirement Gathering and Analysis:.....	17
2. System Design:.....	17
3. Natural Language Processing (NLP) Implementation:	18
4. Testing and Evaluation:	18
5. System Development:	18
6. Deployment and Maintenance:	18
1.6 Significance of Project	19
1.7 Project Schedule	19
1.8 Expected Outcome	19
Chapter 2 : Literature Review	21
2.1 Introduction	21
2.2 Review of the similar existing system	23
2.2.1 Exabytes Live Chat & Ticket System	23
2.2.2 Extendly – My CRM Support.....	29
2.3 Comparison Between the Existing System and Proposed System	32
2.4 Summary	33
Chapter 3: Requirement Analysis and Design	35
3.1 Introduction	35
3.2 Agile Methodology	35
3.3 Analysis Phase and User Requirements	37
3.3.1 Analysis of Online Survey	38
3.3.2 Analysis of Interview Question	46
3.3.3 Summary of Interview.....	48
3.4 System Design	49
3.4.1 Use Case Diagram	49
3.4.2 Activity Diagram.....	50
3.5 System Requirement	60
3.6 Software Requirement	61
3.7 System Functionalities	61
3.8 Summary	63
Chapter 4: System Design and Implementation.....	64
4.1 System Architecture Overview	64
4.2 System Roles and Functionalities.....	64
4.2.1 General User (Public Ticket Submitter)	64

4.2.2 Admin	64
4.2.3 Developer	65
4.3 System Architecture	66
4.3.1 Installation and Configuration of System Components	66
4.3.2 Architecture Overview	68
4.4 Interface Design	69
4.4.1 Role Selection Page	69
4.4.2 Admin and Developer Login System	70
4.4.3 Public Ticket Submission	71
4.4.4 Admin Dashboard	72
4.4.5 Developer Dashboard	74
4.4.6 Developer workload summary	75
4.4.7 Developer and Admin Ticket Page	75
4.4.8 Developer Ticket Detail Page (unassigned ticket)	76
4.4.9 Developer Ticket Detail Page (assigned ticket)	77
4.4.10 Developer Ticket Detail page (own assigned ticket)	78
4.4.11 Admin Ticket Detail Page	78
4.5 Database Design	79
4.6 Security Considerations	80
4.7 Tools Used	80
4.8 Summary	80
Chapter 5: System Testing and Evaluation	81
5.1 Introduction	81
5.2 Functional Testing	81
5.3 Test Cases and Results	82
5.4 Test Environment	84
5.5 User Feedback and Evaluation	85
5.6 Conclusion	90
Chapter 6: Conclusion and Future Work	91
6.1 Conclusion	91
6.2 Project Strengths	92
6.3 Limitations	92
6.4 Future Enhancements	92
References	93
Appendix A	94
Appendix B	94

List of Tables

Table 2.1 Comparison of features between similar existing systems and proposed system

Table 3.1 Data Dictionary Table

Table 3.2 Hardware Requirement

Table 3.3 Software Requirement

Table 3.4 System Functionalities for user

Table 3.5 System Functionalities for Admin and Support Team

Table 5.1 Test Case Table

Table 5.2 Test Case Table (admin)

Table 5.3 Test Case Table (developer)

List of Figures

Figure 1.0: Agile Methodology Cycles

Figure 1.1 Schedule for FYP 1

Figure 1.1 Schedule for FYP 2

Figure 2.1 Exabyte Login Page

Figure 2.2: Main Page of Exabyte Support Portal for Ticket Creation

Figure 2.3: Exabytes Live Chat Pre-Chat Form

Figure 2.4: Ticket Details for Domain Inquiry

Figure 2.5: List or Dashboard of Open Tickets

Figure 2.6: Details of a Specific Ticket

Figure 2.7: Ticket Resolution History

Figure 2.8 Main Page of Extendly – The CRM Support Portal

Figure 2.9 Support Ticket Submission Page

Figure 2.10: Ticket Confirmation Page

Figure 2.11 Ticket Overview Page

Figure 3.1 Age group of respondents

Figure 3.2 Respondent field of work

Figure 3.3 Ticketing system or CRM software experience

Figure 3.4 Respondent experience in experiencing delays in resolving issues

Figure 3.5 Respondent satisfaction with the current issue resolution process

Figure 3.6 Common challenges face by the respondent with the current system

Figure 3.7 Respondent expectation towards the proposed system

Figure 3.8 Respondent view on how important the real-time system update

Figure 3.9 Respondent preference towards system that support real-time notification

Figure 3.10 Respondent expected features for the proposed system

Figure 3.11 Use Case Diagram

Figure 3.12 Activity Diagram for the user

Figure 3.13 Activity Diagram for support team/administrator

Figure 3.14 User login sequence diagram

Figure 3.15 Support Team Login Sequence Diagram

Figure 3.16 Admin Login Sequence Diagram

Figure 3.17 Logout Sequence Diagram

Figure 3.18 User Registration Sequence Diagram

Figure 3.19 View All Ticket/ View Ticket Sequence Diagram

Figure 3.20 Ticket Submission Sequence Diagram

Figure 4.1: Role Selection Page

Figure 4.2: Admin and Developer Login System

Figure 4.3 Public Ticket Submission

Figure 4.4: Admin Dashboard

Figure 4.5 Developer Workload Summary

Figure 4.6 Developer Dashboard

Figure 4.7 Developer Workload Summary

Figure 4.8 Developer Workload Summary

Figure 4.9 Developer and Admin Ticket Page

Figure 4.10 Developer Ticket Detail Page (unassigned ticket)

Figure 4.11 Developer Ticket Detail Page (assigned ticket)

Figure 4.12 Developer Ticket Detail Page (own assigned ticket)

Figure 4.13 Admin Ticket Detail Page

Figure 5.1 Role

Figure 5.2 System Login Functionality Rate

Figure 5.3 Role Selection Page Navigation Functionality

Figure 5.4 Ticket Submission Process Functionality

Figure 5.5 View Ticket Page Functionality

Figure 5.6 System Behaviour Functionality

Figure 5.7 User Interface Design Rate

Figure 5.8 Developer Workload Summary and Activity Log Usability

Figure 5.9 Ticket Filtering and Sorting Usability

Figure 5.10 Suggestion For Improvement

Figure 5.11 System Readiness for Real-World Deployment

Figure 5.12 System Recommendation To Public User

List of Abbreviation

CRM Customer Relationship Management

UNIMAS Universiti Malaysia Sarawak

IT Information Technology

AI Artificial Intelligence

NLP Natural Language Processing

Chapter 1: Introduction

1.1 Introduction

In today's era of fast-paced digitalization, businesses are becoming more reliant on technology to deliver the effectiveness in providing customer support. For Benadigital, who have been focusing on digital marketing and marketing software as well as IT-services, maintaining customer support requests is very crucial to ensure customer satisfaction.

However, Benadigital currently faces problems in addressing customer support request due to direct, manual approach that they have been offering, which often resulting in delays in providing the solution to their customer and unsatisfactory outcome. As the demand for timely responses continues to increase over time, the urges to streamline and improve the management process have come to light.

The “Benadigital Ticketing System for Issue Management” focuses on addressing the inefficiencies that have been identified by providing a managerial platform to handle customer support inquiries by creating tickets. Mainly leveraging Natural Language Processing (NLP), this system helps to automate the classification and categorization of tickets and offered AI-powered solutions to analyze customer sentiments to prioritize critical issues. With aim to improve the long-term support experience by decreasing the response times and ensuring customer receive the best solutions for the issues they encountered.

As Benadigital grows, customer service plays a big role in maintaining customer user experience and trust. With the integration of NLP technologies, this system will help both customers and the support team to better engage with each other, allowing them to resolve the issues more effectively.

1.2 Problem Statement

Benadigital faces significant challenges while maintaining customer satisfaction and managing customer support issues as they offer direct approach for inquiries regarding their marketing software and other IT-related issues resulting in operational inefficiencies. These support issues have leads to delays in solution delivery time as per customer service report and impacting customer experience and trust in the service.

These issues affect various operational aspects such as follows:

1. Efficiency: Lack of an organized support management system for tracking and managing support requests resulting in longer response time and resolution times. This inefficiency inconveniences the productivity of the support team and spawn barrier in issue resolution.
2. Clarity: When there is not a systematic and automated method in place, it becomes challenging to ensure effective communication and transparency between the support team and clients. This hinders accountability and diminishes the level of service provided.
3. Customer Satisfaction: Delayed replies and solutions lead to customer frustration and dissatisfaction, undermining the company's ability to foster strong, positive relationships with its clients.
4. Scalability: With Benadigital's ongoing growth in its customer base, the existing strategy will face challenges in addressing rising demands, which will worsen delays and inefficiencies.

Intervention in these challenges is crucial for boosting the overall efficiency of customer support operations. By introducing an automated ticketing system for managing issues, processes can be streamlined, response times can be improved, and customer satisfaction can be enhanced, all in line with Benadigital's dedication to maintain the high service standards.

1.3 Scope

Benadigital Ticketing System for Issue Management is designed to create a proper automated solution to handle the demand of customer support inquiries over time. The system will allow users to submit support tickets, categorizing the issues automatically using Natural Language Processing (NLP). It will also prioritize tickets based on urgency, analyze user sentiment to detect emotions, and offer suggestions for common issues to reduce repetitive inquiries. The system will enable the support team to track and manage tickets efficiently, providing real-time updates to users regarding the status of their issues.

1.4 Aims and Objectives

1. To provide a structured platform for users to raise tickets, automate ticket classification using natural language processing (NLP), and streamline the issue resolution process.
2. To improve the efficiency and effectiveness of support operations, ultimately enhancing user experience.
3. To keep users informed by providing real-time updates and notifications for better communication during the issue resolution process.

1.5 Brief Methodology

The development of the Benadigital Ticketing System for Issue Management will follow an agile software development methodology to ensure flexibility and iterative progress.

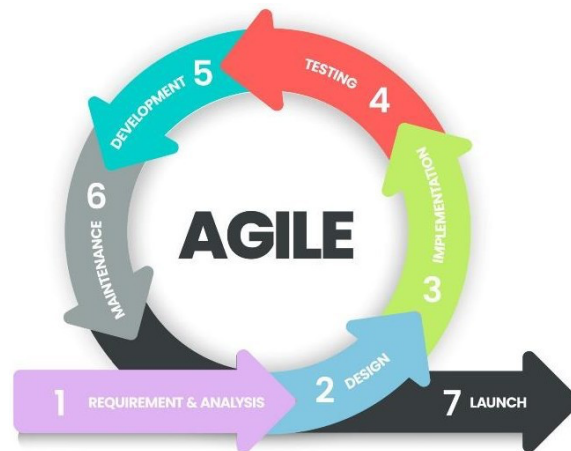


Figure 1.0: Agile Methodology Cycles

The project will be divided into several stages:

1. Requirement Gathering and Analysis:

- Conduct interviews with stakeholders (support team and users) to understand current challenges and requirements.
- Analyze existing systems and identify gaps that need to be addressed in the new ticketing system.

2. System Design:

- Design a user-friendly web-based interface for submitting and managing support tickets.
- Plan the architecture of the system, ensuring scalability and security.
- Design the backend structure to support automated ticket classification, NLP processing and suggestions.

3. Natural Language Processing (NLP) Implementation:

- Implement an NLP model to automatically classify issues based on the content of user submissions.
- Integrate sentiment analysis to assess user emotions and prioritize tickets accordingly.
- Develop algorithms that analyze historical ticket data to provide automated responses and suggest potential solutions for common issues.
- Use machine learning to continuously improve the accuracy of suggestions.

4. Testing and Evaluation:

- Perform unit testing, integration testing, and user acceptance testing (UAT) to ensure the system meets the specified requirements.
- Evaluate the system's performance by comparing ticket response times and user satisfaction before and after implementation.

5. System Development:

- Develop the web interface and backend functionality.
- Implement the NLP components using suitable programming languages (e.g., Python for NLP).
- Test all features to ensure the system functions as expected.

6. Deployment and Maintenance:

- Deploy the system on the company's server, ensuring it is accessible to users and support staff.
- Provide training to users and support staff to ensure smooth adoption.

- Continuously monitor the system for any issues, and provide updates and maintenance as needed.

1.6 Significance of Project

Benadigital Ticketing System for Issue Management project is significant as its main objective is to enhance the efficiency of the customer support inquiries regarding the marketing software and IT-related issues by automating tickets classifications. Leveraging the NLP and a structured platform for ticket creation, the system will improve the time response, streamline operation and enhance the support efficiency and will be able to results in faster deliveries and more organized support process as well as contributing to the growth and success of Benadigital's customer service operations.

1.7 Project Schedule

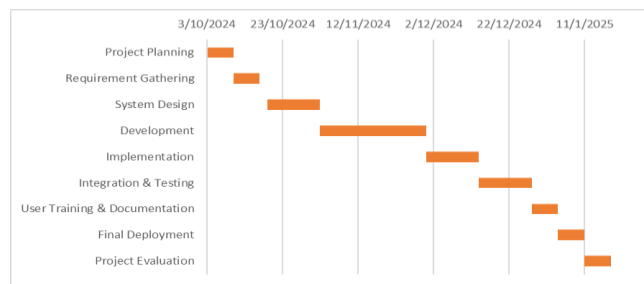


Figure 1.1 Schedule for FYP 1

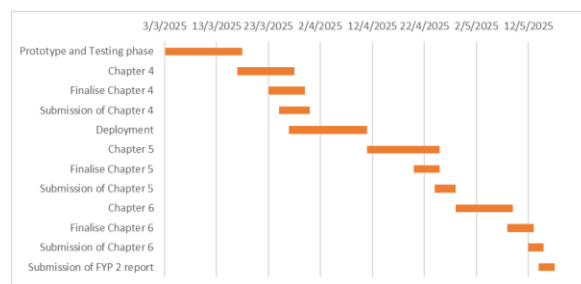


Figure 1.2: Schedule for FYP 2

1.8 Expected Outcome

This project is expected to improve issue management for Benadigital's customer support team and provide the following benefits:

- I. Efficient Issue Resolution: A more structured and automated ticketing system will lead to quicker issue classification and faster resolution times, improving overall support efficiency.
- II. Enhanced Customer Satisfaction: With faster response times and more accurate handling of requests, customer satisfaction will increase, strengthening Benadigital's reputation for excellent customer support.
- III. Streamlined Support Operations: Automation and improved ticket management will reduce the workload of support staff, enabling them to focus on solving high-priority issues.

This system will lead to better customer experiences and more efficient operations for Benadigital's customer support team. Efficient Issue Resolution: A more structured and automated ticketing system will lead to quicker issue classification and faster resolution times, improving overall support efficiency.

- I. Enhanced Customer Satisfaction: With faster response times and more accurate handling of requests, customer satisfaction will increase, strengthening Benadigital's reputation for excellent customer support.
- II. Streamlined Support Operations: Automation and improved ticket management will reduce the workload of support staff, enabling them to focus on solving high-priority issues.

This system will lead to better customer experiences and more efficient operations for Benadigital's customer support

Chapter 2 : Literature Review

2.1 Introduction

Customer support systems play a vital role in ensuring the satisfaction and retention of clients, particularly in industries that rely heavily on technology and digital marketing. Despite technological advancements, many support systems face inefficiencies in addressing customer issues, leading to delays and dissatisfaction. Natural Language Processing (NLP)-based chatbots have emerged as a solution, providing real-time assistance, automated classification, and issue resolution. This literature review explores relevant research on NLP-driven chatbots, their role in customer support, and their applicability to issue management systems like the proposed Benadigital Ticketing System.

Advancements in NLP-based Chatbots

The evolution of chatbots began with rule-based systems like Eliza, which relied on keyword recognition without true understanding (Castro et al., 2023). Subsequent advancements introduced conversational agents like Siri, Alexa, and Google Assistant, which integrated more sophisticated AI and NLP techniques for better interaction capabilities. However, limitations remain, particularly in emotion recognition and user intent classification, as highlighted by (Castro et al., 2023). The integration of affective recognition capabilities and Speech Act Theory (SAT) is a growing area of research aimed at enhancing conversational relevance and emotional understanding in chatbots.

The study by Gowda (et al., 2021) emphasized the educational potential of chatbots, illustrating how rule-based and AI-driven models can address logistical challenges in real-time interactions. By leveraging NLP, their proposed chatbot successfully facilitated information retrieval and improved engagement through structured databases and tokenization techniques.

NLP in Customer Feedback Management

Chhibber (et al., 2023) demonstrated the effectiveness of NLP for classifying customer feedback into multiple categories, achieving a high accuracy rate of 91.35% using Gradient Boosting Classification. Their chatbot was designed for a paper manufacturing company to categorize customer comments into actionable insights, aiding in decision-making and service improvement. The methodology incorporated data preprocessing, SMOTE for balancing datasets, and model evaluation using precision and recall metrics. This approach aligns closely with the requirements of issue management systems, where feedback categorization and resolution are paramount.

The integration of machine learning models like Random Forest and Support Vector Machines further enhances the capabilities of chatbots in real-time analysis. Chhibber noted the potential limitations of chatbots in handling colloquial language and the importance of regular updates to improve performance.

Chatbots for Issue Management

Existing research highlights the potential of NLP-driven chatbots for streamlining issue management processes. For instance, Gowda (et al. 2021) developed a rule-based chatbot capable of addressing predefined queries and escalating unresolved issues to administrators. This hierarchical approach ensures that unresolved tickets are logged and addressed systematically. Similarly, Chhibber (et al., 2023) emphasized the scalability of chatbots for handling customer interactions, categorizing issues, and facilitating resolution through automated feedback loops.

Moreover, the use of hybrid models combining rule-based and AI-driven techniques, as suggested by (Castro et al., 2023), offers a promising pathway for addressing complex queries in issue management systems. The incorporation of Speech Act Theory and emotional analysis further ensures that responses are contextually relevant and empathetic, thereby improving user satisfaction.

Relevance to the Benadigital Ticketing System

The proposed Benadigital Ticketing System aims to address inefficiencies in customer support by providing a structured platform for ticket creation and resolution. By leveraging NLP, the system can automate ticket classification, prioritize issues based on urgency, and route them to appropriate departments. Insights from Chhibber (et al., 2023) on data preprocessing and classification models can be directly applied to ensure high accuracy in ticket categorization.

Additionally, the integration of affective recognition techniques, as discussed by Castro (et al., 2023), can enhance the system's ability to provide personalized and empathetic responses. This aligns with the goal of improving user satisfaction and streamlining support operation.

Conclusion

The reviewed literature underscores the transformative potential of NLP-driven chatbots in customer support and issue management. By combining advancements in machine learning, NLP, and emotion recognition, systems like the Benadigital Ticketing System can address existing inefficiencies and deliver enhanced user experiences. Future research should focus on refining emotion recognition techniques and exploring hybrid models to address the challenges of real-time issue management in diverse industries.

2.2 Review of the similar existing system

2.2.1 Exabytes Live Chat & Ticket System

Exabytes Network Sdn Bhd (Exabytes) is a leading provider of cloud-based services and digital solutions in Malaysia, offering a variety of hosting, website, and digital marketing services (Exabytes, n.d.).

Founded in 2001, Exabytes has grown to become a major player in the industry, providing comprehensive IT solutions to businesses of all sizes (Exabytes, n.d.). The company initially focused on web hosting and domain registration, but over the years, it expanded its services to include cloud hosting, e-commerce solutions, and digital marketing services (Exabytes, n.d.).

Exabytes has successfully transitioned from a web hosting service provider to a full-service IT company offering a range of products that include cloud solutions, web development, and digital marketing services (Exabytes, n.d.). Their strategic shift aims to cater to businesses looking for more integrated digital solutions (Exabytes, n.d.). Exabytes Support Portal (Live Chat & Ticket System) is a cloud-based infrastructure system that Exabytes staff uses to manage client hosting services, domain registrations, and e-commerce operations. The login page for the system is shown in Figure 2.1.

GROW Digital English

exabytes® Products & Services Submit a ticket My Tickets Login Sign up

Login to the support portal

Enter the details below

Your e-mail address

Password

☒ Remember me on this computer

[Forgot your password?](#) [Login](#)

[Are you an agent? Login here](#)

...or login using

[G Google](#) [f Facebook](#)

Sign up

[Sign up with us](#)

Once you sign up, you will have complete access to our self service portal and you can use your account to raise support tickets and track their status.

Figure 2.1 Exabyte Login Page

GROW Digital

English



[Products & Services](#)[Submit a ticket](#)[My Tickets](#)[Login](#)[Sign up](#)

Submit a ticket

Requester *

Subject *

Department *

Priority *

Description *

B

I

U











+ Attach a file

☐ I'm not a robot


reCAPTCHA
[Privacy](#) - [Terms](#)

Submit

Cancel

Figure 2.2: Main Page of Exabyte Support Portal for Ticket Creation

Figure 2.2 below shows the main page of the Exabytes Ticket System for ticket creation. The main page provides a simplified form for creating new support tickets, where users can specify the department, subject, priority, and description of the issue to ensure streamlined ticket management and resolution.

PreChat Form

X

We can't wait to talk to you. But first, please take a couple of moments to tell us a bit about yourself.

Name *

Email *

Mobile Phone *

Start Chat

Figure 2.3: Exabytes Live Chat Pre-Chat Form

Figure 2.3 shows the Exabytes Live Chat Pre-Chat Form, which allows users to initiate a live chat session with customer support. The form requires users to fill in essential information, including their name, email, and mobile phone number, before proceeding. This ensures that support agents have the necessary details to provide personalized and efficient assistance. The "Start Chat" button at the bottom enables users to begin their conversation with a support representative.

Your ticket has been created.

GROW DigitalEnglish

exabytes®

Products & ServicesSubmit a ticketMy TicketsEdit profileSign out

Home > Tickets list

#10611194

Domain for benadig.com

Please check the domain whether it is secure or not from your end.
#

+ Attach a file

Reply

Cancel

Ticket details

Department *
Domain

Status
Being Processed

Priority *
Low

Department
Exabytes.MY - Domains

Update

Figure 2.4: Ticket Details for Domain Inquiry

The Exabytes Ticket System provides a detailed interface for managing customer inquiries. Figure 2.4 highlights the ticket details page, where users can view ticket status, priority, and associated departments. The page also enables users to update ticket information or communicate directly with support staff, streamlining the resolution process.

GROW DigitalEnglish

exabytes®

Products & ServicesSubmit a ticketMy TicketsEdit profileSign out

All Tickets

Sorted by Date Created

Export tickets

Domain for benadig.com #10611194
Created on Thu, 16 Jan at 1:43 AM

BEING PROCESSED

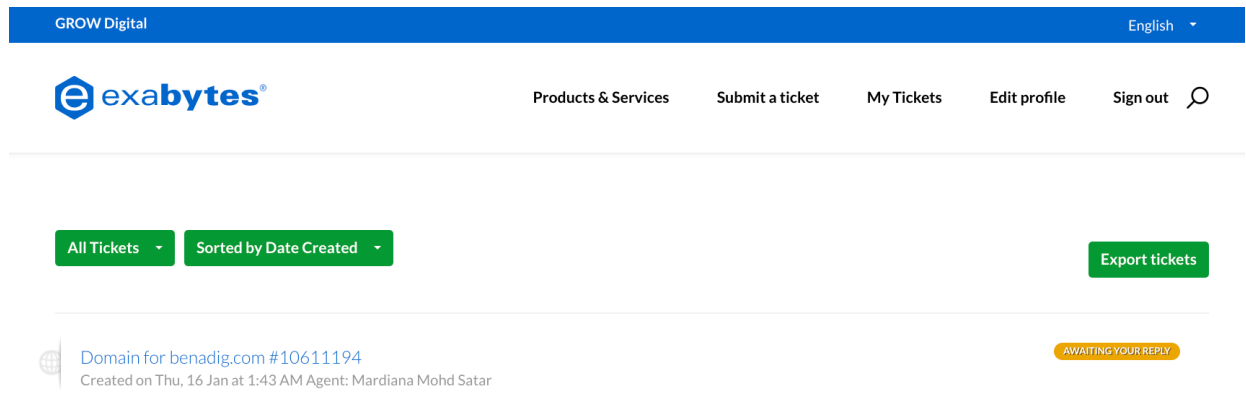


Figure 2.5: List or Dashboard of Open Tickets

Figure 2.5 shows the Open Tickets Dashboard in the Exabytes Ticket System. The dashboard displays all tickets from being processed and waiting for user action. Columns include the ticket ID, date created, subject of the issue, status, and the assigned agent.

Submit a ticket

Requester *

Subject *

Department *

Priority *

Description *

B *I* U

Please check the domain.

[+ Attach a file](#)

I'm not a robot



Related articles...

[Managing domains in the domain manager](#)

the Domain Manager, the general domain list, and how to manage your domains. ... manage the domains for a website? 1. Manage domains either in the Domain

[How To Add An IDN Domain As Domain Alias / Parked Domain](#)

guide user who owns an IDN domain and wish to make it as domain alias with their ... their host domain. First users are required to hold an ... an IDN domain and

[What Is The Difference Between A Sub-Domain, Parked Domain And An Add-on Domain?](#)

Sub domain Let's say your domain is domain.com. You install a E-Commerce and ... and put in a directory called domain.com/commerce/. You can turn the

[Why was my domain deleted from Cloudflare?](#)

Domain deletion commonly occurs for two reasons: A user with access to the domain ... nameservers. Cloudflare continuously monitors domain registration

[Domain Alias](#)

What is Domain Alias Domain aliases is allowing additional domain names to point ... example: XYZ Corporation registers two domain names, A and B. It

[Show all results](#)

Figure 2.6: Details of a Specific Ticket

Figure 2.6 illustrates the details of a specific ticket. The system provides comprehensive information such as the requester email address, subject, department assigned, issue description, priority level, and allow user to attach file if necessary.

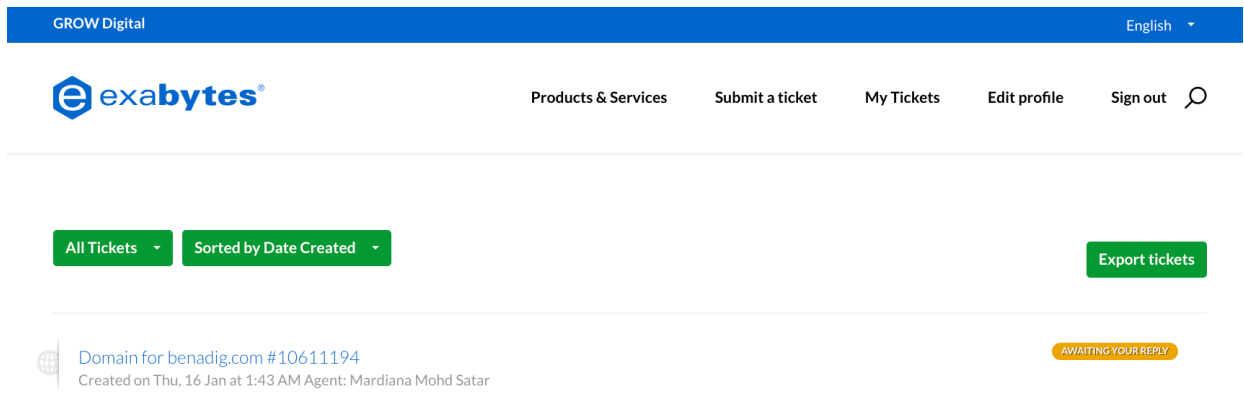


Figure 2.7: Ticket Resolution History

Figure 2.7 shows the Resolution History Page, where all ticket updates are recorded. Each entry includes the timestamp, ticket ID, actions taken such as status update, and the agent name that are assigned to assist further.

2.2.2 Extendly – My CRM Support

Extendly or better known as The CRM Support Portal provides an intuitive and user-friendly interface for addressing customer support needs. The portal allows users to search for help articles, view ticket status, and submit new support tickets for any issues they face with the CRM system. These features aim to streamline the support process, ensuring a seamless user experience and faster resolution times.

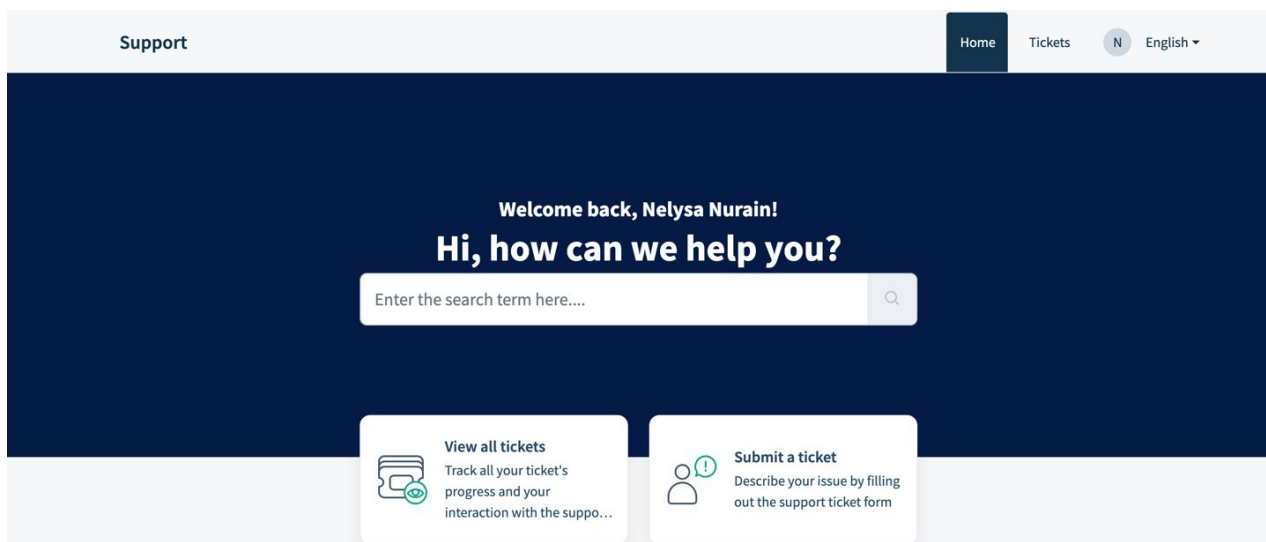


Figure 2.8 Main Page of Extendly – The CRM Support Portal

Figure 2.6 shows the main page of the CRM Support Portal. The navigation bar at the top includes options for navigating to the home page, viewing submitted tickets, and accessing account settings. The page also greets users with a personalized welcome message and features a prominent search bar for users to quickly find relevant information. Below the search bar, two primary options are displayed:

1. View all tickets: Allows users to track the progress of their support tickets and view interactions with the support team.
2. Submit a ticket: Enables users to describe their issue and submit a new support request via a straightforward ticket form.

This layout is designed to provide clarity and ease of use, improving the overall efficiency of the support system and user satisfaction.

The screenshot displays the 'Support' page of a CRM system. At the top, a navigation bar contains links for 'Home', 'Tickets', and a user profile icon labeled 'N' with 'English' as a language selector. Below the navigation bar is a dark blue header with a 'Home' link on the left and a search bar on the right with the placeholder text 'Enter the search term here...'. The main content area features a light gray box containing the ticket submission form. The form includes a text input for 'Your Platform Email Address' (pre-filled with 'nelysanurain@gmail.com'), an 'Add CC' button, a text input for 'A Brief Description of the Issue', and a rich text editor for 'What's your question?'. The rich text editor has a toolbar with icons for bold, italic, underline, link, list, image, and code. Below the text input is a checkbox with the text 'I acknowledge that submitting this ticket without sufficient information may delay the response and resolution.' and an 'Attachment' link. At the bottom of the form are 'Cancel' and 'Submit' buttons.

Figure 2.9 Support Ticket Submission Page

Figure 2.7 shows the ticket submission form on the support page. The form allows users to provide their platform email address, a brief description of the issue, and a detailed question or concern. Users can also attach files and acknowledge terms before submitting the ticket.

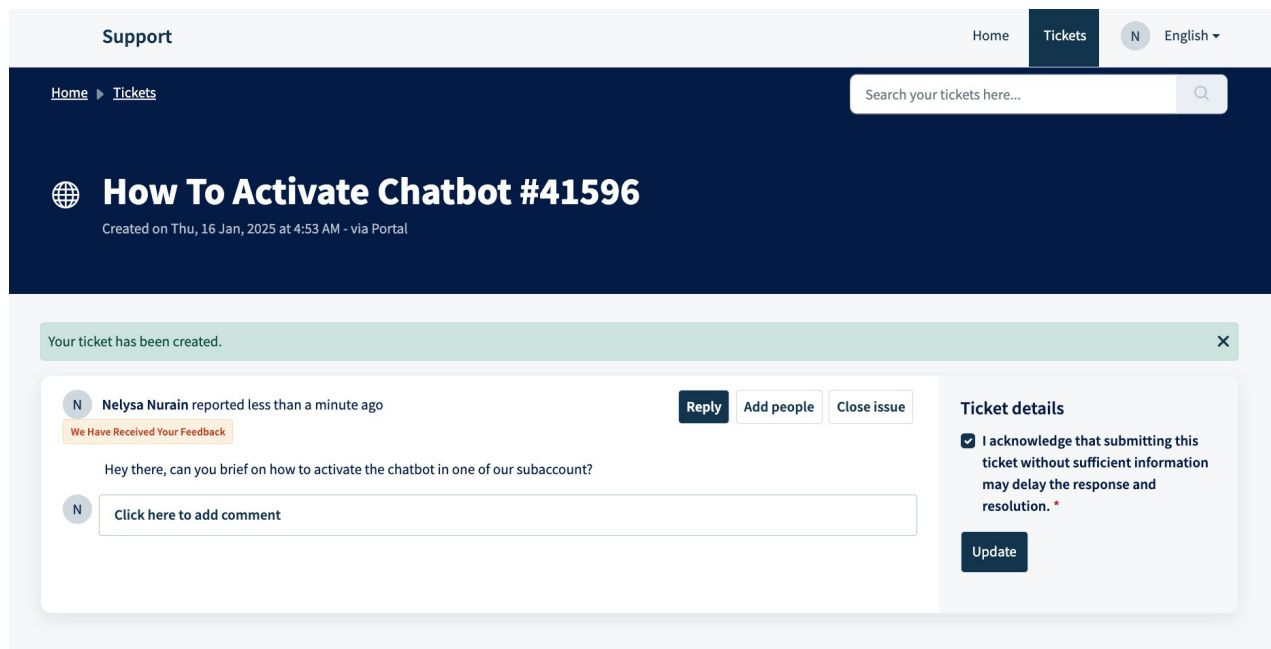


Figure 2.10: Ticket Confirmation Page

Figure 2.8 shows that the ticket has been created. The page displays the creation date, ticket ID and time alongside with the message "Your ticket has been created" and below it, the user can see the confirmation that their feedback has been received. The page allows users to reply directly to support, add people or close issue if needed.

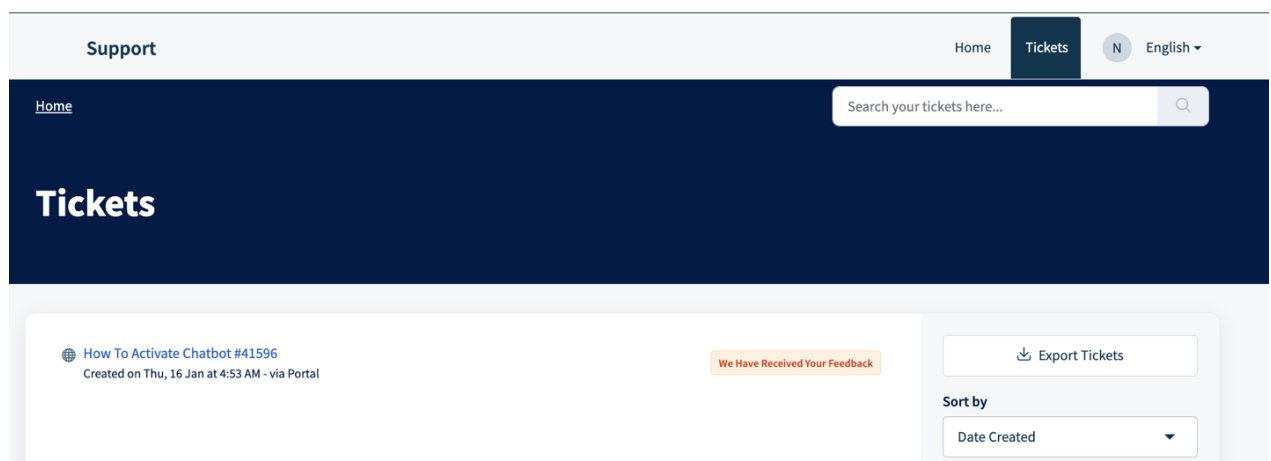


Figure 2.11 Ticket Overview Page

Figure 2.9 displays the Ticket Overview Page in the My CRM Support Portal. The page features a list of user tickets, including the title, creation date, and status notification that feedback has been received. It also provides sorting options for users to manage their tickets effectively.

2.3 Comparison Between the Existing System and Proposed System

The comparison of features between similar existing systems and the proposed system is in Table 2.1.

Features	Exabyte	Extendly	Benadigital Ticketing System
Sign up and log in to the page	The system has a log-in and sign-up page for clients. The form design is simple, and data validation works.	The login and signup forms are simple but lack advanced security features. Data validation works correctly.	The system has a log-in page and sign-up page. The login page design is simple with proper data validation.
Ticket creation	No automated ticket classification feature.	Limited automation available for ticket categorization; relies on predefined tags.	Uses NLP to automatically classify and categorize tickets for faster and more accurate processing.
Real-time status updates	Users can check ticket status via email notifications.	Status updates are provided but delayed for certain tickets.	Provides real-time ticket status updates through a dashboard and notifications to improve response times.

User interface and navigation	The user interface is outdated and non-intuitive, making navigation difficult for users.	The interface is user-friendly but lacks customization options for advanced users.	The system features a unified, modern, and responsive design for easy navigation and enhanced user experience.
Integration with other systems	Integrates with some third-party tools, but the setup is not-user friendly.	Provides seamless integration with CRM tools, but additional costs apply for premium integrations.	Designed to integrate with existing CRM systems at Benadigital for a cohesive workflow with minimal setup.
Customization options	Customization is minimal, with few configurable settings.	Offers moderate customization, such as branding and workflow tweaks.	Highly customizable, including custom workflows, ticket fields, and automated responses tailored to user needs.
Cost-effectiveness	Relatively affordable but lacks advanced features for modern support systems.	Costs are higher due to integration and advanced automation tools.	Cost-effective as it is developed in-house with features tailored to Benadigital's operational needs.

Table 2.1 Comparison of features between similar existing systems and proposed system

2.4 Summary

Chapter 2 begins with an introduction to the literature review, focusing on ticketing and issue management systems. Three existing systems similar to the proposed Benadigital Ticketing System are

reviewed in this chapter: Exabytes, Extendly, and Benadigital Ticketing System (proposed system). The history, services, and features provided by these systems are further explored to establish their relevance.

The features, user interfaces, and functionalities of these systems are analyzed to guide the development of the proposed system. A comparison table highlights key differences between the three systems and the proposed system. These comparisons help identify the limitations of existing systems and provide insights into features the proposed system can enhance or introduce. By analyzing the strengths and weaknesses of the existing systems, the proposed Benadigital Ticketing System aims to offer improved efficiency, user experience, and automation compared to current solutions.

Chapter 3: Requirement Analysis and Design

3.1 Introduction

Benadigital Ticketing System for Issue Management is a web-based system designed to improve the management of customer support requests at Benadigital Sdn. Bhd. This chapter provides details for the first two agile phases, which are the analysis and design phases. User requirements are gathered through surveys and interviews with user who have use CRM software before. The findings are analyzed to define the design, features, and functionalities of the Benadigital Ticketing System.

Hardware and software specifications are outlined to ensure system compatibility and maximise performance. The development phase of this methodology will be covered in the next chapter, focusing on the implementation of the system.

3.2 Agile Methodology

Agile methodology is a flexible and iterative software development approach based on the principles outlined in the Agile Manifesto. Introduced in 2001, the Agile Manifesto emphasizes collaboration, adaptability, and responsiveness to change, making it a widely adopted methodology in modern software engineering (Beck et al., 2001). Agile methodologies focus on delivering small, functional increments of a product through continuous iterations, ensuring that user feedback is integrated into the development process.

Agile development is characterized by phased deliverables, frequent testing, and iterative cycles that allow teams to adapt quickly to changing user requirements. Each phase of development produces a functional component of the final product, which is shared with users for feedback. This iterative approach reduces the risks of late-stage modifications by addressing issues early in the development lifecycle (Sinha & Das, 2021). Agile methodology ensures that changes in user needs or business requirements can be seamlessly incorporated in subsequent iterations, enhancing the product's overall adaptability and relevance.

User involvement is a core tenet of Agile, with regular collaboration between developers and stakeholders. According to Tokdemir and Uğuz (2022), Agile emphasizes relationships, functioning software, and adaptability to ensure that development aligns with user expectations. Frequent user feedback ensures that the final product delivers the desired functionality and quality.

The Agile process consists of five key phases: brainstorming, design, development, quality assurance, and deployment. These phases are iterative, allowing for thorough testing and improvement before progressing to the next stage. During the brainstorming phase, requirements are identified, and the project scope is defined. The design phase involves creating prototypes or wireframes to outline the software's structure. Development focuses on implementing features incrementally, while quality assurance involves rigorous testing to ensure functionality and reliability. Finally, deployment delivers the completed increments to users in a timely manner.

The iterative nature of Agile methodology enhances productivity, minimizes risks, and ensures the timely delivery of functional software that meets user requirements. By promoting continuous improvement and active collaboration, Agile has become a cornerstone of efficient and user-centric software development practices.

3.3 Analysis Phase and User Requirements

The analysis phase, also referred to as the requirement analysis or brainstorming phase, involves gathering and analyzing information from users to identify their needs. This phase is crucial in developing a system that incorporates features and functionalities aligned with user expectations. By understanding user requirements, the development team ensures that the final outcome effectively addresses user needs and delivers the expected value.

One effective method for collecting user requirements is conducting an online survey.

For this project, the survey is distributed via Google Forms using the opportunity sampling method to gather insights from the target community which is user who have use CRM software or ticketing system before. The survey focuses on obtaining their feedback and experience with customer support issue while resolving issues and their expectations for a better system.

To maximize participation, the survey is shared through multiple channels such as WhatsApp, Instagram and Telegram. Respondents are reminded that responses are use for academic purposes only before proceeding with the questionnaire.

In addition to the survey, an interview with user who have been using CRM software on daily basis is conducted to gather qualitative insights. The interview focuses on understanding the current challenges in issue management by Benadigital and gathering feedback on the proposed system. The interview ensures that the system design incorporates

the user practical experience and operational knowledge, contributing to a solution that addresses both user and operational needs.

This dual approach of surveys and interviews ensures that the analysis phase captures diverse perspectives, enabling the development of a comprehensive and user-centric issue management system.

3.3.1 Analysis of Online Survey

The online survey consists of three sections designed to gather comprehensive insights into the requirements for the Benadigital Ticketing System for Issue Management. The link for the online survey is <https://forms.gle/62QyVWsP6x25haTe9>. The survey received responses from 21 participants.

Section A: Demographic Profile

Section A focuses on collecting demographic information about the respondents, including their age, occupation, and experience in using CRM or ticketing systems. These details help in understanding the diversity of the respondent pool and ensuring the system addresses the needs of various user groups.

Section B: Current Issue Management Practices

Section B gathers data on the respondents' experiences with the current issue management practices at Benadigital. Questions in this section aim to identify challenges such as delays in ticket resolution, user satisfaction levels, and frequency of unresolved issues.

Section C: Opinions on the Proposed Ticketing System

Section C captures respondents' opinions and expectations regarding the proposed Benadigital Ticketing System. Questions focus on desired features, such as automated ticket classification, real-time updates, and user-friendly interfaces. Respondents are also asked about their priorities, including response time, ease of use, and integration with existing systems.

This survey analysis provides valuable insights that will guide the design and development of the ticketing system, ensuring it aligns with user expectations and resolves existing inefficiencies.

Section A: Demographic Profile

Question 1: What is your age group?

What is your age group?
21 responses

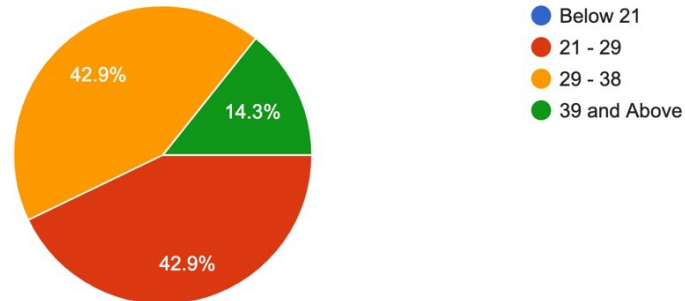


Figure 3.1 Age group of respondents

Figure 3.1 shows the pie chart for the respondents age group. The majority of the respondents are in age group of 21-29 as well as 29-38, and the rest are in the age group of 39 and above. In total, 9 respondents are in between the age of 21 to 29, 9 respondents are in between 29 to 38 and 3 respondent are 39 and above. The survey had a higher number of respondents below the age of 39 compared to respondent age above 39.

Question 2: Profession/ Field of Work

Profession/ Field of Work

21 responses

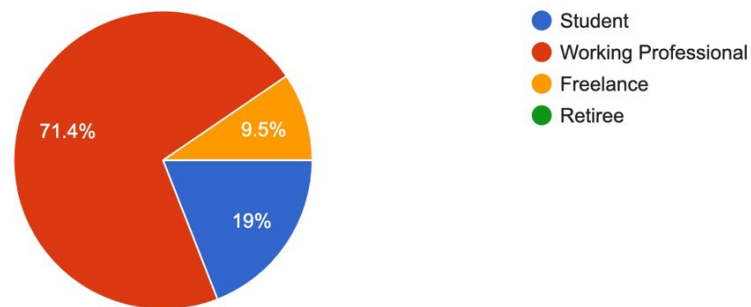


Figure 3.2 Respondent field of work

Figure 3.2 shows the pie chart for respondent field of work. Most respondent are working professional. In total, 9 respondent are student, 15 are working professional, 2 are doing freelance and no amount of respondent are retiree.

Question 3: Have you used a ticketing system or CRM software before?

Have you used a ticketing system or CRM software before?

21 responses

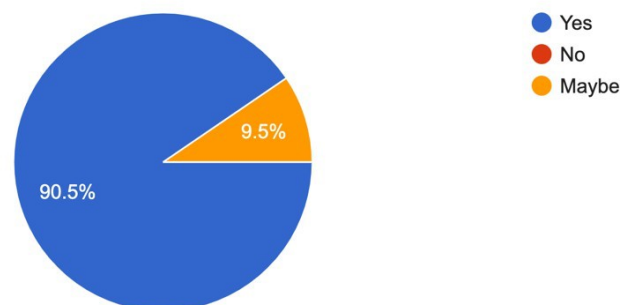


Figure 3.3 Ticketing system or CRM software experience

Figure 3.3 indicates 19 of the respondents have experience in using ticketing system or CRM before and only 2 of the respondent are not sure whether they have experience using CRM or

ticketing system before. No amount of respondent has never use ticketing system or CRM software before.

Question 4: How often do you experience delays in resolving issues?

How often do you experience delays in resolving issues?

21 responses

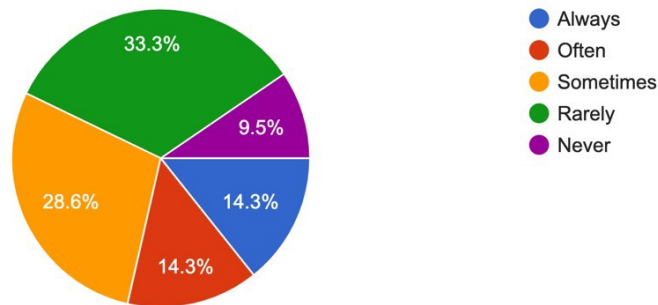


Figure 3.4 Respondent experience in experiencing delays in resolving issues

The pie chart shows that 7 of the respondents always experience delays in resolving issues while 3 of the respondents often experiencing delays in resolving issues as well as 6 of the respondents who have sometimes experience delays in resolving issues. However, we can see that 7 of the respondents rarely experience delays while resolving issues and 2 of the respondents never experience delays while resolving issues.

Question 5: On a scale of 1-5, how satisfied are you with the current issue resolution process?

On a scale of 1-5, how satisfied are you with the current issue resolution process?

21 responses

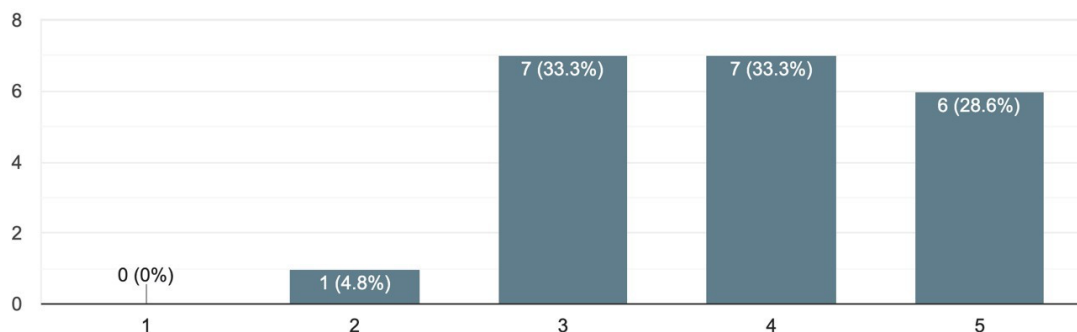


Figure 3.5 Respondent satisfaction with the current issue resolution process

Figure 3.5 shows that 5 of the respondent are very satisfied with the current issue resolution while no amount of respondent are not satisfied with the current issue resolution process.

Question 6: What are the most common challenges you face with the current system?

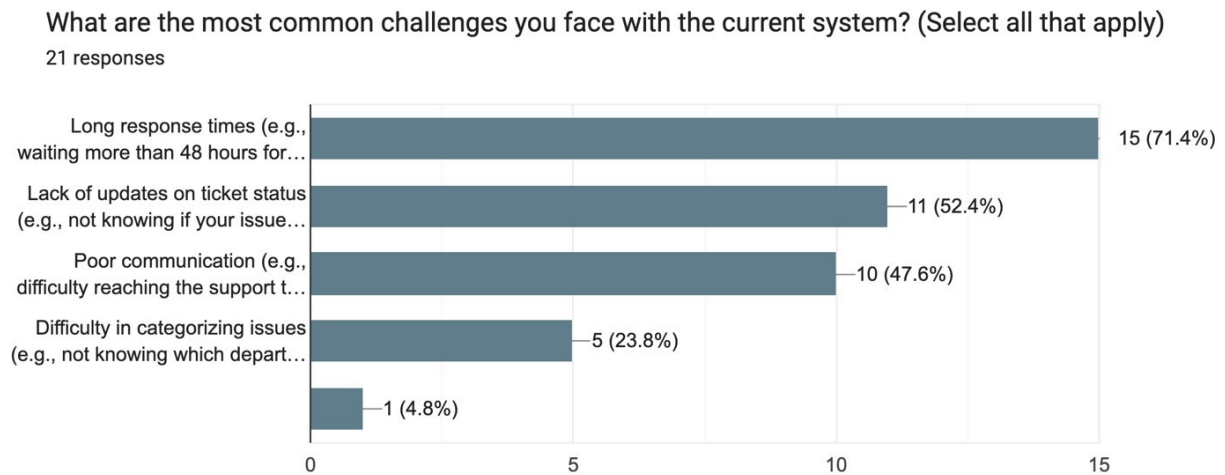


Figure 3.6 Common challenges face by the respondent with the current system

Figure 3.6 shows that 15 of the respondent face long responses time as common challenges. Next, 11 of the respondent dealt with lack of updates on ticket status as common challenges and 10 of them faces poor communication as common challenges. Difficulty incategorizing issues is also challenges that are faced by 5 respondent and one is categorized as others.

Question 7: Which feature would you prioritize in the new ticketing system?

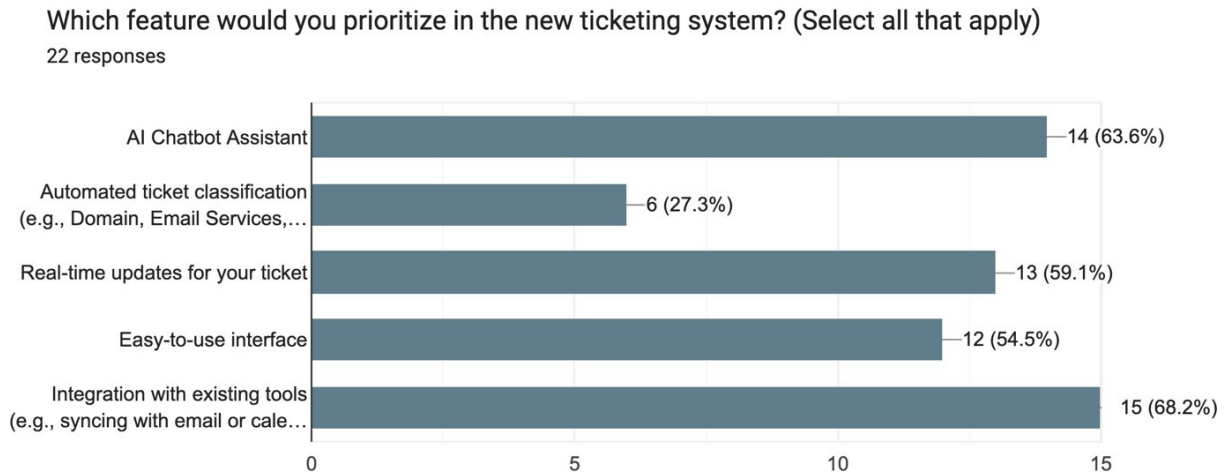


Figure 3.7 Respondent expectation towards the proposed system

Figure 3.7 shows that most of respondent priority are set to integration with existing tools, AI chatbot assistant, automated ticket classification and easy-to-use interface while marking the automated ticket classification as less priority.

Question 8: How important is it for the system to provide status updates?

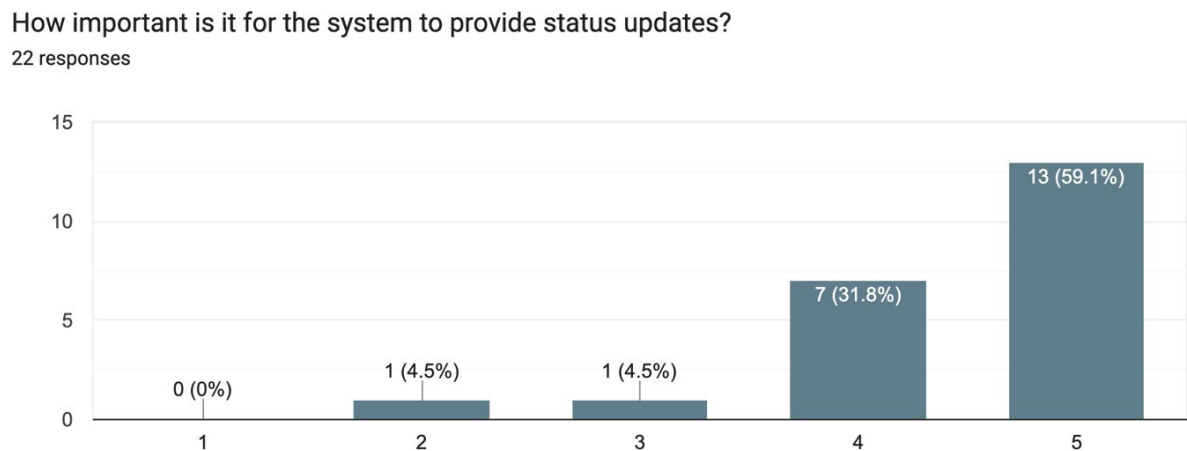


Figure 3.8 Respondent view on how important the real-time system update

Figure 3.8 indicates 5 out of 21 respondent thinks it is very important for the system to provide status update while only 2 of them shows that it is less important for the system to provide real-time update.

Question 9: Would you prefer a system that supports real-time notifications (e.g., via email or SMS)?

Would you prefer a system that supports real-time notifications (e.g., via email or SMS)?
22 responses

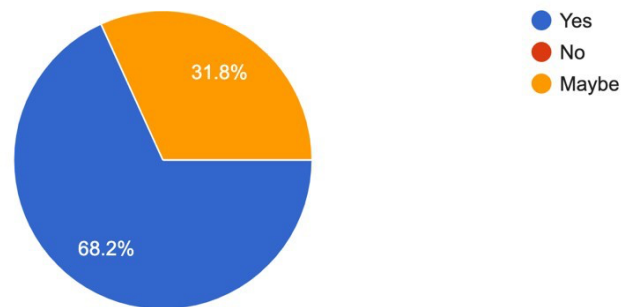


Figure 3.9 Respondent preference towards system that support real-time notification

Figure 3.9 shows that 15 out of 21 respondent prefers the system to provide real-time notification while the rest are not sure if they agree on the real-time notification.

Question 10: Any additional features you would like to see in the new ticketing system?

Any additional features you would like to see in the new ticketing system?

5 responses

Automation
agent that assist
24 hr real people
ticket id
No

Figure 3.10 Respondent expected features for the proposed system

Figure 3.10 shows respondent expectation on additional features for the proposed system. There are 5 responses to this open-ended question. One suggests to add ticket ID and another suggest that assigned agent to be show in the system. The rest is not related.

3.3.2 Analysis of Interview Question

Below is the interview question with one of the user that have been using Benadigital Software (CRM software) and dealing with some issues while using it.

Question 1: How long have you been using CRM software by Benadigital?

Answer: I've been using it for around 8 months.

Question 2: How does your work relate to using a CRM software?

Answer: My job involves managing leads and giving them consultation. Then, I need to update in the CRM.

Question 3: Do you encounter issues in regular basis with the CRM?

Answer: Not really but I do sometimes.

Question 4: Can you specify what issues usually are you dealing with?

Answer: Server problem most of the time.

Question 5: Are you having problem contacting customer service to deal with the issues?

Answer: Most of the time, yes. I need to manually WhatsApp them to inform about the issue and sometimes they replied late since it's outside office hour.

Question 6: How long do they usually resolve issues that arise after you contacted?

Answer: Depending on the issues, sometimes on the spot or sometimes can take overnight or even a week.

Question 7: Are you familiar with the flow of resolving issues within the CRM?

Answer: Not really since I usually contact them directly.

Question 8: Do you find it inconvenience not to have proper system to submit ticket about the arising issues?

Answer: Not really but it would be nice to have one. I don't like direct contact cause I find it uncomfortable. It is better if just notification from email or the system itself.

Question 9: Do you think it will be hard for you to switch from directly contacting the support team to submitting a ticket for issues arise?

Answer: Maybe but I think by some time all will be oka

3.3.3 Summary of Interview

Interview summary with Benadigital CRM software user concludes that the user has been using the Benadigital CRM software for around 8 months and is primarily involved in managing leads and providing consultations, with regular updates entered into the CRM system.

While the user doesn't experience frequent issues, they occasionally face server-related problems. A key challenge is the difficulty in contacting customer service, as the user needs to manually message via WhatsApp, which can result in delayed responses, especially outside of office hours. The time taken to resolve issues varies, sometimes being resolved on the spot, but other times taking from overnight to a week.

The user is not fully familiar with the CRM's issue resolution process, as they typically handle issues by directly contacting support. Although they don't find the lack of a formal ticket submission system to be a major inconvenience, they would prefer a more streamlined method, such as automated notifications via email or the system itself. The user believes they could adapt to a ticketing system, although the transition might be challenging at first.

This feedback suggests that introducing a more efficient, automated ticketing system could improve the user's experience with the CRM software, offering a less intrusive and more timely resolution process.

3.4 System Design

3.4.1 Use Case Diagram

Figure 3.11 shows the use case diagram for the proposed web-based system, Benadigital Ticketing System for Issue Management

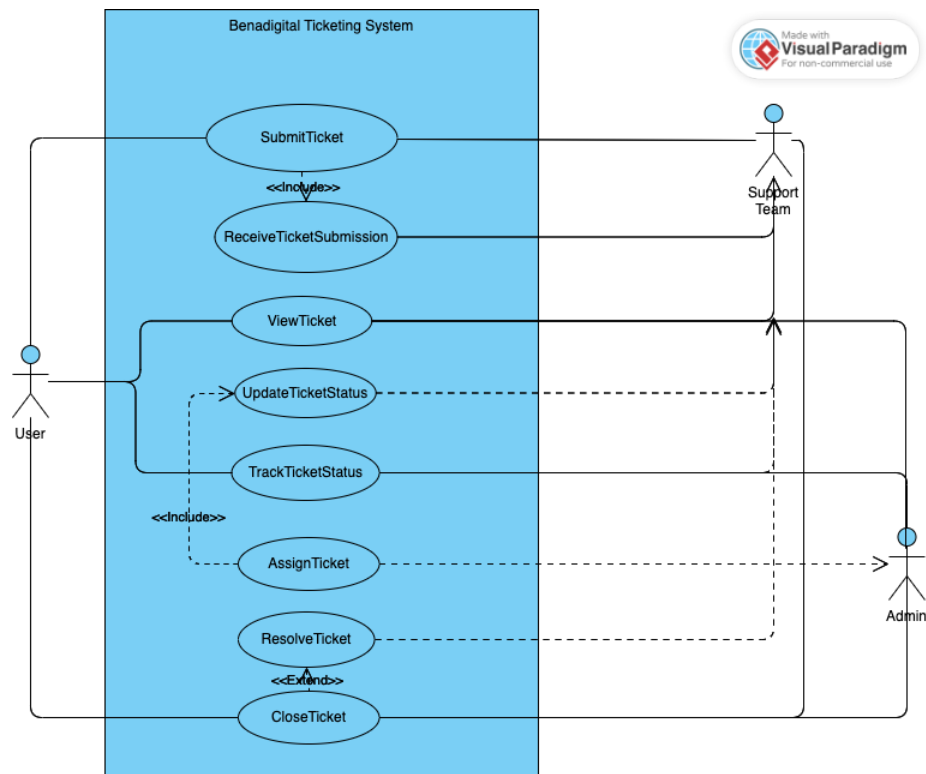


Figure 3.11 Use Case Diagram

3.4.2 Activity Diagram

Figure 3.12 and 3.13 shows the activity diagram for the user, support team and administrator.

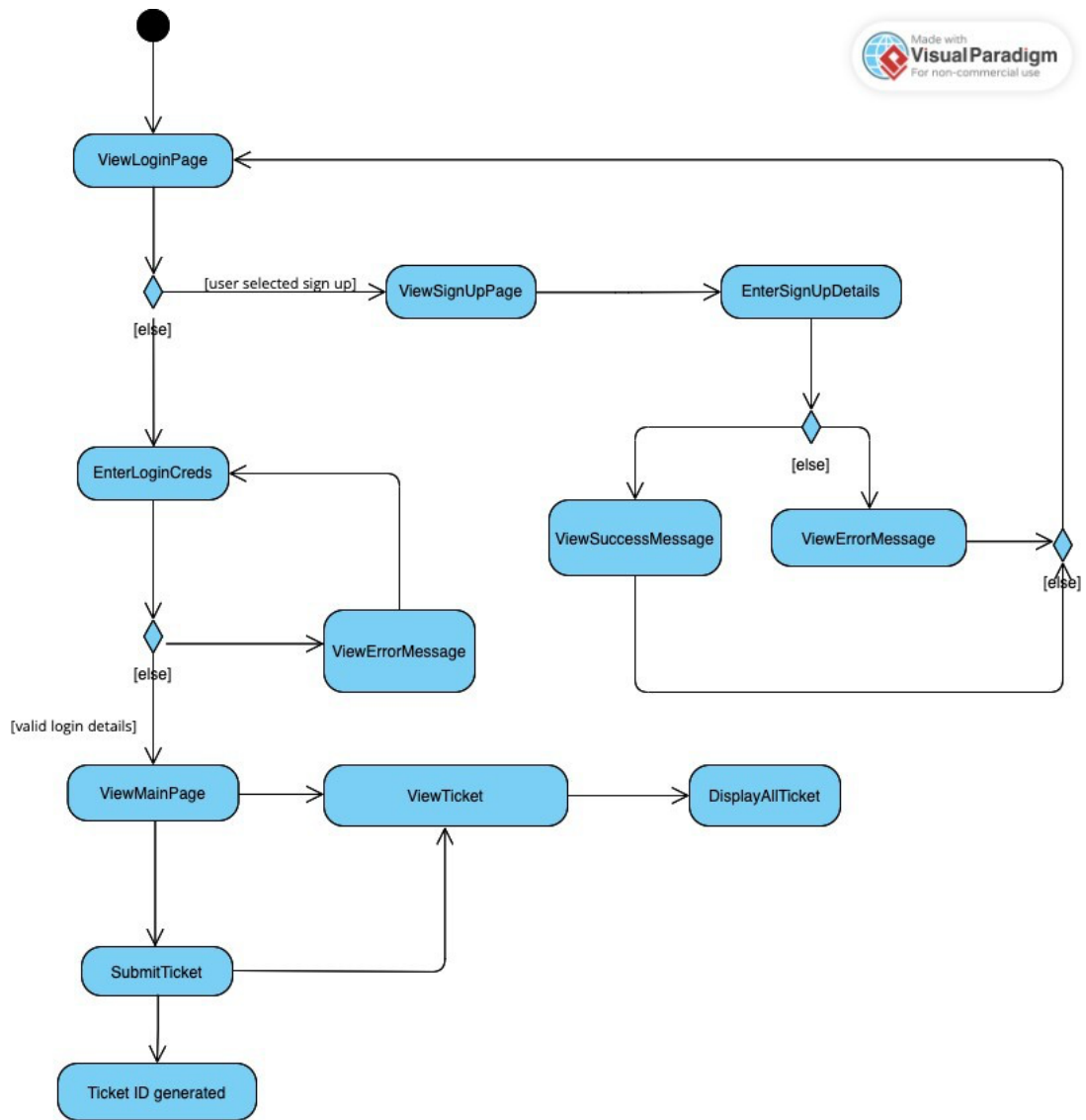


Figure 3.12 Activity Diagram for the user

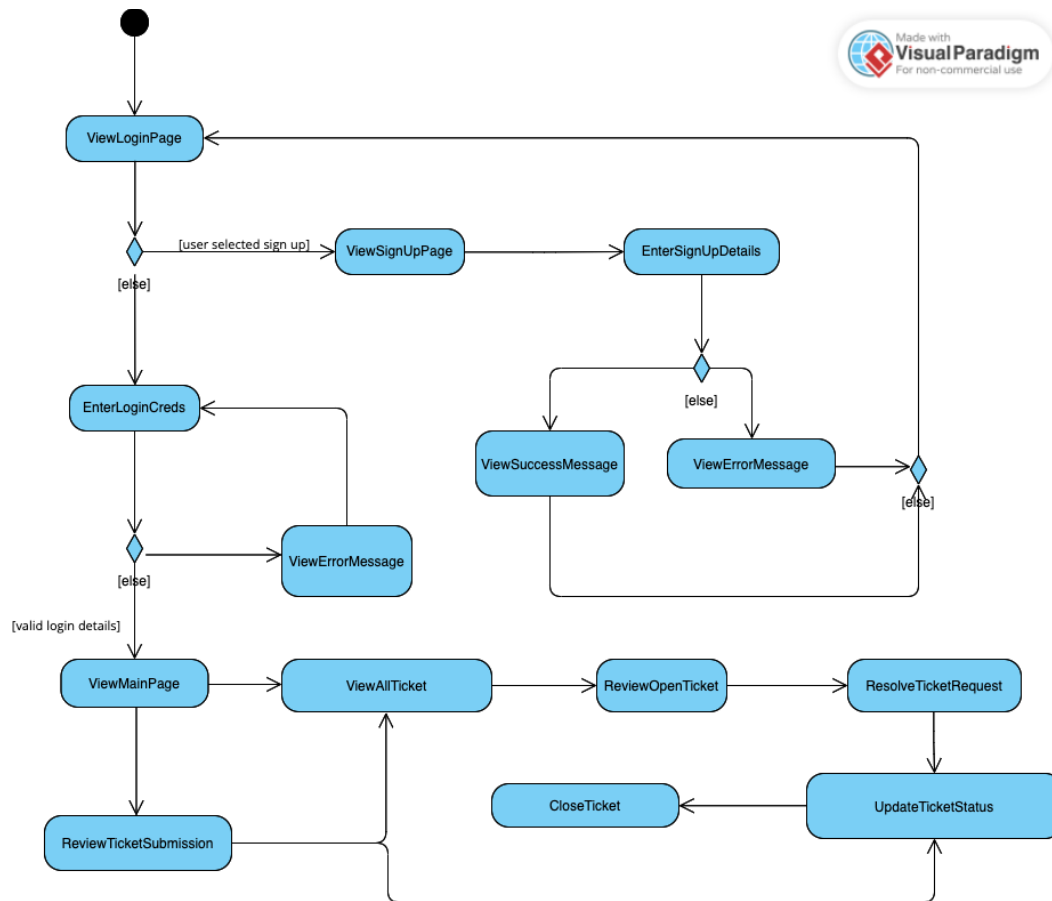


Figure 3.13 Activity Diagram for support team/administrator

3.3.3.1 Data Dictionary table

Entity	Field	Data Type	Description	Constraints
User	userID	Integer	Unique identifier for the user.	Primary key, not null
	name	String	Name of the user.	Not null
	email	String	Email address of the user.	Unique, not null
Ticket	ticketID	Integer	Unique identifier for the ticket.	Primary key, not null

	description	String	Details of the issue reported.	Not null
	status	String	Current status of the ticket (e.g., Open, Closed).	Default: "Open"
	priority	String	Priority of the ticket (e.g., High, Medium, Low).	Default: "Medium"
	assignedTo	Integer	Identifier of the support team or admin handling it.	Foreign key to Support/Admin
Admin	adminID	Integer	Unique identifier for the admin.	Primary key, not null
	name	String	Name of the admin.	Not null
Support Team	supportID	Integer	Unique identifier for the support team member.	Primary key, not null
	name	String	Name of the support team member.	Not null
System	systemID	Integer	Unique identifier for the system.	Primary key, not null
	logDetails	String	Logs for system activities or notifications.	Optional

Table 3.1 Data Dictionary Table

3.3.4 System Sequence Diagram

3.3.4.1 Login for user

Figure 3.15 shows the sequence diagram for user login process. The user needs to input correct login details to successfully login as user into the system. The system will show an error message if the user login details are invalid.

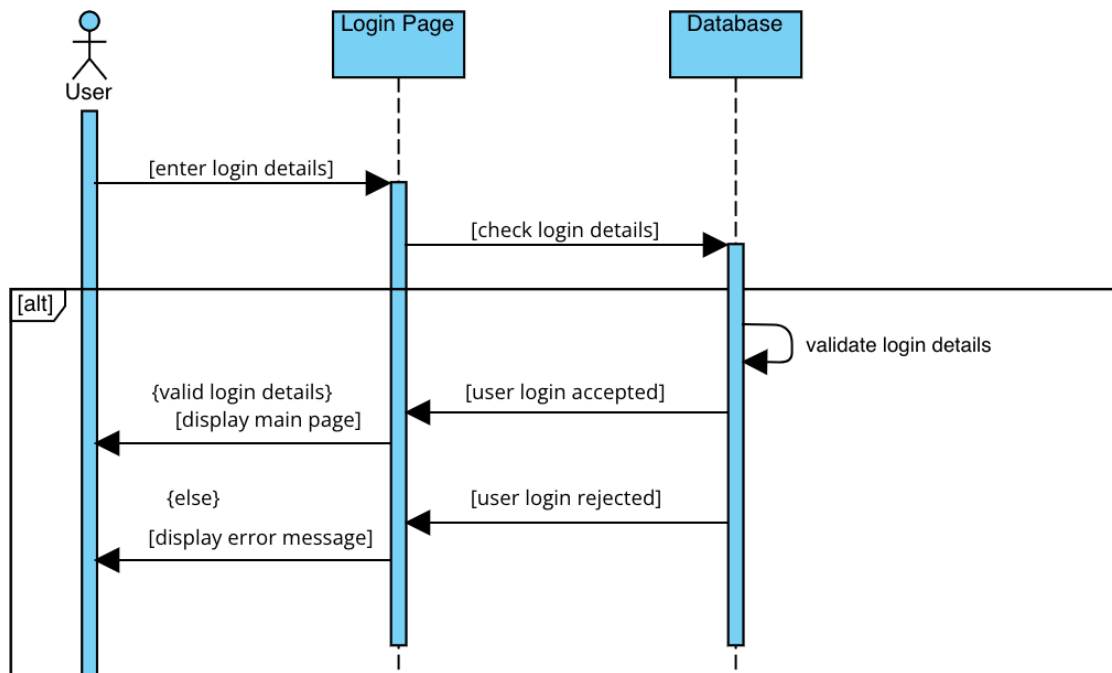


Figure 3.14 User login sequence diagram

3.3.4.2 Login for support team

Figure 3.16 shows the sequence diagram for support team login process. The support team needs to input their login credentials correctly to successfully login into the system. The system will show an error message if the login credentials are invalid.

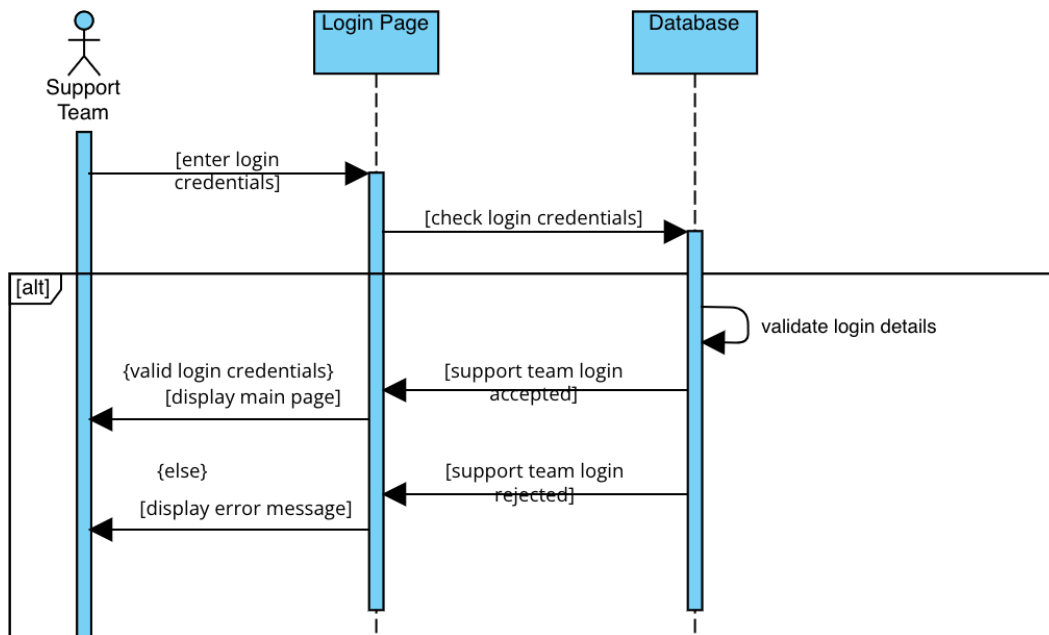


Figure 3.15 Support Team Login Sequence Diagram

3.3.4.3 Login for administrator

Figure 3.17 shows the sequence diagram for admin login process. The admin needs to input correct login details to successfully login as admin into the system. The system will show an error message if the admin login details are invalid.

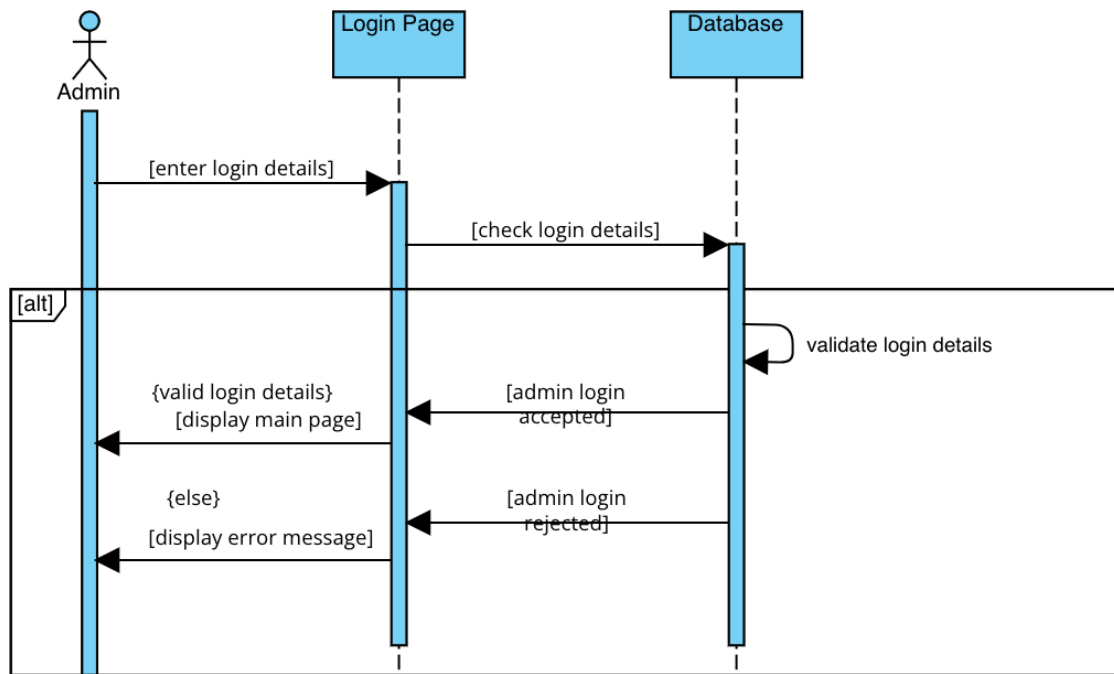


Figure 3.16 Admin Login Sequence Diagram

3.3.4.4 Logout

Figure 3.18 shows the sequence diagram for logout. The admin, support team or user can logout from the system after confirming to logout.

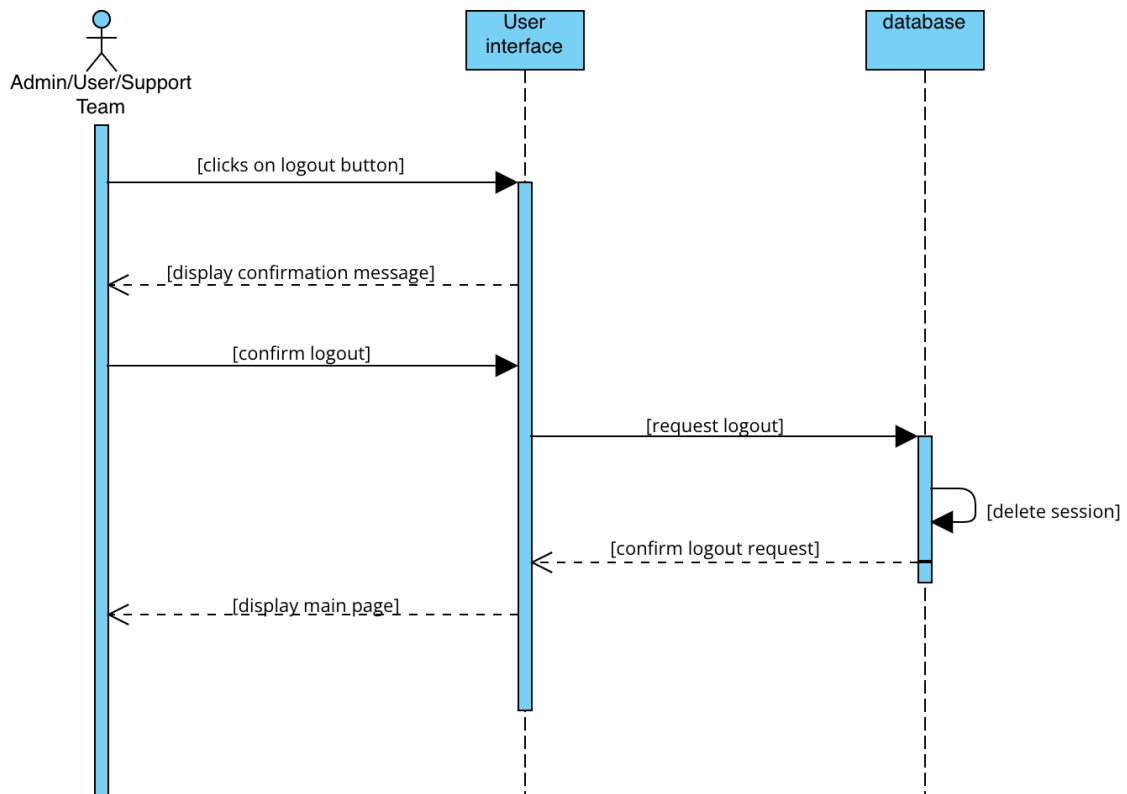


Figure 3.17 Logout Sequence Diagram

3.3.4.5 User Signup / Registration

Figure 3.19 shows the sequence diagram for the user signup or registration process. The user needs to enter their details to register. The registration process is successful if the user details are valid.

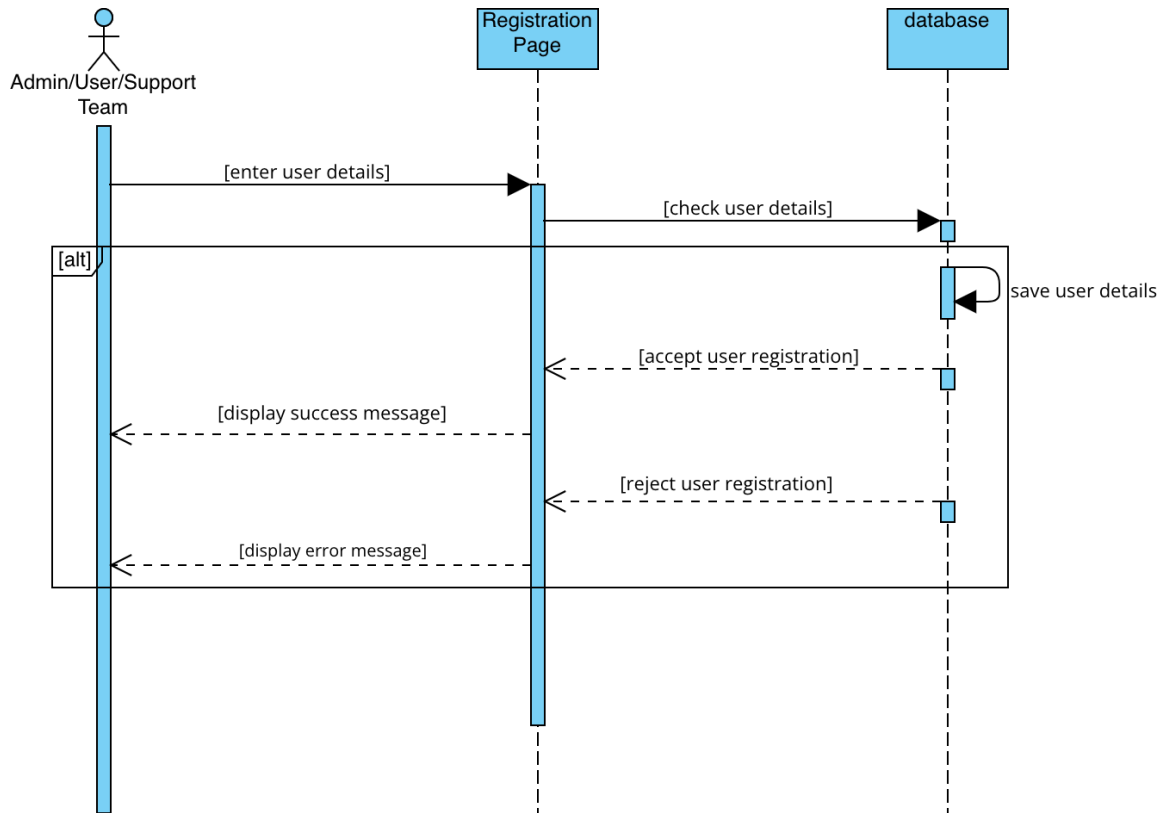


Figure 3.18 User Registration Sequence Diagram

3.3.4.6 View All Ticket

Figure 3.20 shows the sequence diagram to view all ticket. The user/support team/admin can click on the “View All Ticket” button to view all the ticket. The system will display all the ticket according to the status and date created.

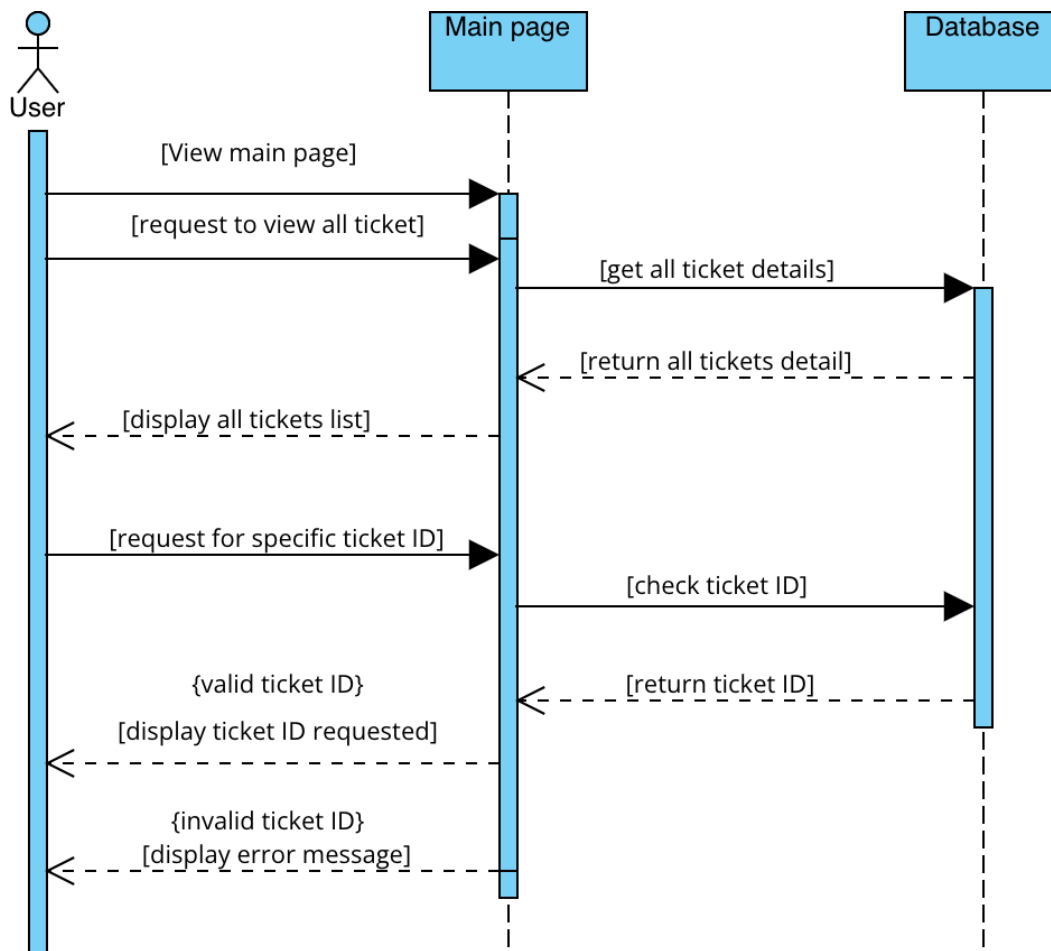


Figure 3.19 View All Ticket/ View Ticket Sequence Diagram

3.3.4.7 Submit Ticket

Figure 3.21 shows the sequence diagram to make a ticket submission. The user can click on the “Submit” button to submit the ticket. The system will display the status and date created after ticket is successfully submitted.

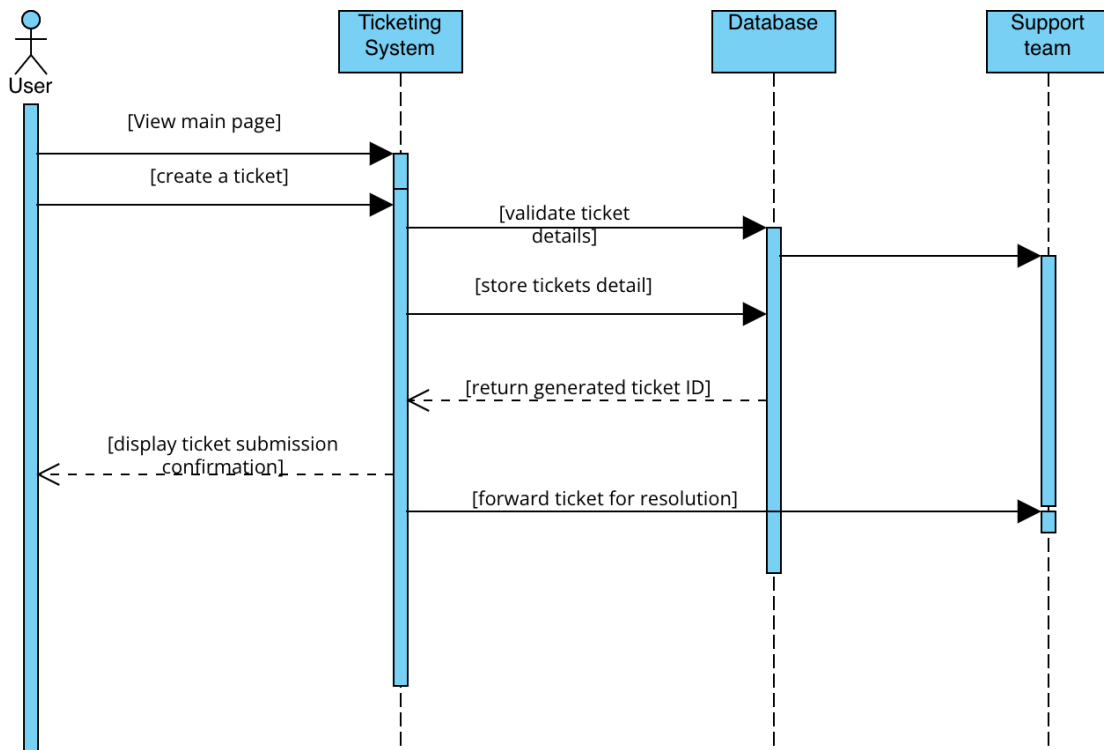


Figure 3.20 Ticket Submission Sequence Diagram

3.5 System Requirement

3.1.1 Hardware Requirement

Table 3.2 shows the recommended specifications of hardware for the proposed system.

Hardware Specifications	Requirements

Computer	Processor (Central Processing Unit or CPU): Intel Core i3 or AMD Ryzen 3 processor (dual-core or higher) Minimum clock speed: 2.0 gigahertz (GHz)
	Storage: 128 Gigabytes (GB) solid-state drive (SSD) or hard disk drive (HDD) or higher
	Memory (random-access memory or RAM): 4GB or higher
	Minimum clock speed: 2.5 gigahertz (GHz)
	Storage: 256 Gigabytes (GB) solid-state drive (SSD) or higher Memory (RAM): 16GB or higher
Web camera	Minimum resolution: 1080p (1920x1080 pixels)

Table 3.2 Hardware Requirement

3.6 Software Requirement

Table 3.3 shows the recommended specifications of software for the proposed system.

Software	Requirement
Code Editor	Visual Studio Code
Web Hosting and Domain	Namecheap
Programming Language	Hypertext Markup Language (HTML), Hypertext Preprocessor (PHP), Cascading Style Sheets (CSS), JavaScript, Python
Database	MySQL
Diagram Design Tool	Visual Paradigm
User Interface Design Tool	Figma
Documentation Tool	Microsoft Office 365, PDF

Table 3.3 Software Requirement

3.7 System Functionalities

The aim of the Benadigital Ticketing System is to streamline the management of customer support requests by providing functionalities for users, support team members, and administrators. Two tables below detail the system features. Table 3.4 outlines the system functionalities for the user, while Table 3.5 describes functionalities for the admin and support team.

System Functionalities	Description
Submit ticket	Users can submit a ticket by providing details such as issue description, priority level, and contact information.

View ticket status	Users can track the progress of their submitted tickets, including current status, updates, and assigned support staff.
Edit ticket information	Users can edit or update details of tickets that are still under review by the support team.
Receive notifications	Users receive notifications for updates on their tickets, including status changes and resolution updates.

Table 3.4 System Functionalities for user

System Functionalities	Description
Assign ticket	Admins can assign tickets to specific support team members based on workload or expertise.
Update ticket status	Support team members can update the status of tickets (e.g., "In Progress," "Resolved") and add comments for users.

Generate reports	Admins can generate reports on ticket resolution rates, average response times, and other system metrics.
Manage user accounts	Admins can manage user accounts, including granting permissions, suspending accounts, and resolving account-related issues.
Ticket escalation	Admins can escalate unresolved tickets to higher-level support or supervisors.

Table 3.5 System Functionalities for Admin and Support Team

3.8 Summary

The Benadigital Ticketing system aims to streamline ticket creation, automate classification using NLP, and improve overall support efficiency. The project involves analyzing existing systems, such as Exabytes and Extendly, to identify gaps and incorporate improvements. Key features include user-friendly interfaces, real-time ticket tracking, automated categorization, and data analytics dashboards. UML diagrams, including use case, activity, and sequence diagrams, are used to design and visualize the system's workflows. The system is expected to enhance response times, user satisfaction, and operational effectiveness. This project leverages advanced technologies to provide a scalable and efficient solution for issue management.

Chapter 4: System Design and Implementation

4.1 System Architecture Overview

This chapter discusses the architecture, interface designs, and functionalities of the Benadigital Ticketing System for Issue Management. The system is designed to provide a seamless and structured platform for users, administrators, and developers to manage support issues efficiently. The system was developed using Flask (Python framework) with MySQL for backend storage and Bootstrap for the user interface. The system supports both authenticated and unauthenticated ticket submission.

4.2 System Roles and Functionalities

The Benadigital Ticketing System is designed to serve three main user roles: General User, Admin, and Developer. Each role is associated with a specific set of functionalities to ensure efficient issue tracking, assignment, and resolution.

4.2.1 General User (Public Ticket Submitter)

General users refer to any user who wishes to submit a support ticket without logging into the system. This includes clients or CRM users of Benadigital. Their access is limited to the following functions:

1. Choose role (User)
2. Submit ticket through the public ticket submission form
3. Receive confirmation or notification upon successful ticket submission

4.2.2 Admin

Administrators have complete control over the ticketing workflow and developer assignments. They are responsible for overseeing the ticket progress and managing the team. Their functionalities include:

1. Choose role (Admin)
2. Access the admin dashboard
3. View developer activity logs
4. View developer workload summary

7. Update ticket status (e.g., Open, In Progress, Closed)
8. Add and post comments related to each ticket
9. Filter tickets based on:
 - Title
 - Creator
 - Status
 - PrioritySorting options include: Priority, Newest, Eldest, Status
10. Submit internal tickets
11. View detailed ticket information

4.2.3 Developer

Developers are responsible for resolving the issues submitted through the system. They share some administrative capabilities but are primarily focused on ticket resolution. Their access includes:

1. Choose role (Developer)
2. Access the developer dashboard
3. View own activity logs
4. View workload summary
5. Assign tickets to themselves (claim unassigned tickets)
6. Edit tickets that are assigned to them
7. Update ticket status during resolution
8. Add and post comments on tickets
9. Filter and sort tickets using the following parameters:
 - Title
 - Creator
 - Status
 - PrioritySorting options: Priority, Newest, Eldest, Status
10. Submit internal tickets (if needed)
11. View detailed ticket information
12. View tickets assigned to them
13. Claim and take ownership of unassigned tickets

4.3 System Architecture

4.3.1 Installation and Configuration of System Components

This guide outlines the steps to set up and run the Benadigital Ticketing System on a local machine using Python, Flask, and MySQL.

i. Prerequisites:

Ensure the following software is installed in the system:

- Python 3.10 or later
- MySQL Server (e.g., via XAMPP or MySQL Installer)
- Git (optional but useful)
- A modern web browser (Chrome, Firefox, etc.)

ii. Required Python Packages:

The following Python libraries is needed:

- Flask
- mysql-connector-python
- bcrypt
- Flask-Session

These can be installed later using pip.

1. Project Setup

1.1. Clone or download the project folder

Place the entire project folder (e.g., benadigital-ticketing-system) into your working directory.

1.2. Create a Python Virtual Environment (optional but recommended)

In your terminal or command prompt:

```
cd path/to/benadigital-ticketing-system  
python -m venv venv
```

Activate the environment:

- Windows:

- macOS/Linux:
- `source venv/bin/activate`

2. Install Required Packages

Run the following command to install all dependencies:

```
pip install -r requirements.txt
```

If requirements.txt is not available, install manually:

```
pip install flask mysql-connector-python flask-session bcrypt
```

3. MySQL Database Configuration

3.1. Start MySQL

If using XAMPP, open the XAMPP Control Panel and start the MySQL module.

3.2. Create a Database

Log into MySQL via terminal or phpMyAdmin and run:

```
CREATE DATABASE benadigital_ticketing;
```

3.3. Import Tables

Use the provided .sql file (if available) or create tables manually as per your data dictionary:

- users
- tickets
- activity_logs
- workload_summary

3.4. Set Credentials in app.py

Open app.py and locate this section:

```
mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="your_mysql_password",  
    database="benadigital_ticketing"  
)
```

Replace with your actual MySQL credentials.

4. Running the Flask App

In terminal (with virtual env activated), run:

```
flask run
```

You should see:

* Running on <http://127.0.0.1:5000/>

5. Access the System in Browser

Open your browser and visit:

<http://127.0.0.1:5000/>

The Role Selection Page will appear as system's entry point.

4.3.2 Architecture Overview

The Benadigital Ticketing System uses a three-tier architecture:

1. Presentation Layer – HTML, CSS (Bootstrap), hosted on PythonAnywhere.
2. Logic Layer – Flask routes handle user requests, logic processing, role-based rendering, and communication with the database.
3. Data Layer – MySQL stores user data, ticket data, comments, developer logs, and role-based permissions.

The system's context diagram describes the high-level interaction between external entities and the system. There are three main actors: General User, Admin, and Developer.

- General User submits support tickets publicly via a no-login form.
- Admin manages the ticket queue and assigns tickets to developers.
- Developers log in to view and manage their assigned tickets.

4.4 Interface Design

4.4.1 Role Selection Page

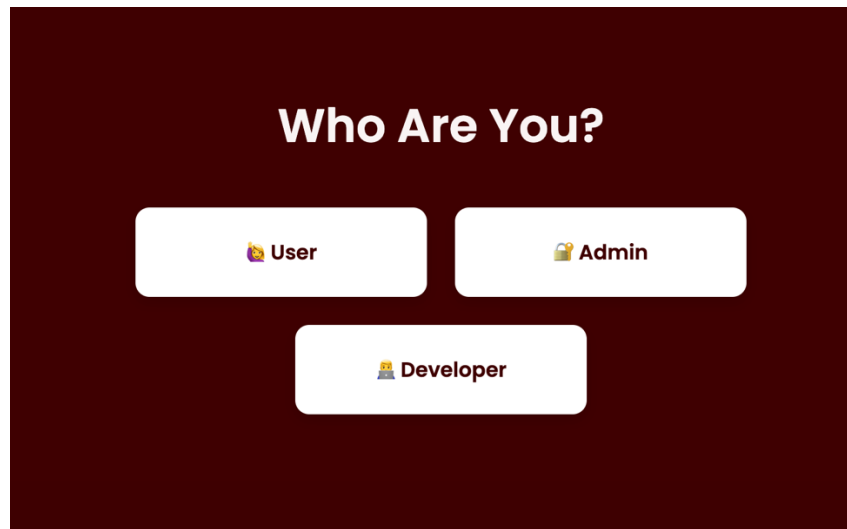


Figure 4.1: Role Selection Page

Upon visiting the platform, users are greeted with a “Who Are You?” role selection page. There are three buttons:

- 🧑 User – redirected to public ticket submission
- 🔑 Admin – redirected to admin login page
- 💻 Developer – redirected to the same login page

This role gateway ensures that the flow is intuitive for different user types.

Before accessing any other functionality, users are directed to a role selection interface. This page introduces a question, “Who Are You?”, followed by three interactive buttons representing the available user types: User, Admin, and Developer. If a user clicks the “User” button, they are taken directly to the public ticket submission form—no login required. Selecting either “Admin” or “Developer” redirects the user to the shared login page, where credentials must be provided to continue.

The purpose of the role selection page is to provide a clean and intuitive navigation experience. It guides users into their respective flows without confusion or accidental misdirection. This separation also enhances security by isolating login access only to relevant users.

4.4.2 Admin and Developer Login System

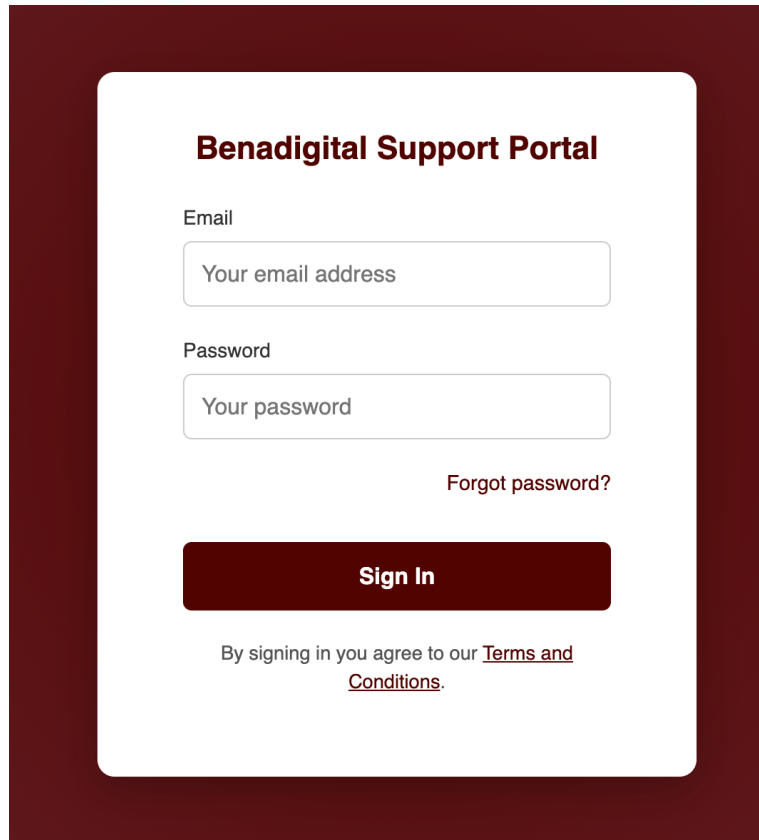
The image shows a login form for the 'Benadigital Support Portal'. The form is centered on a dark red background. It has a white background with rounded corners. At the top, the title 'Benadigital Support Portal' is displayed in a bold, dark red font. Below the title, there are two input fields: one for 'Email' with the placeholder text 'Your email address' and one for 'Password' with the placeholder text 'Your password'. To the right of the password field, there is a link that says 'Forgot password?'. Below these fields is a large, dark red button with the text 'Sign In' in white. At the bottom of the form, there is a line of text that says 'By signing in you agree to our [Terms and Conditions](#)'.

Figure 4.2: Admin and Developer Login System

Admins and Developers use the same login page. Upon successful login, the system checks the role of the user and redirects accordingly:

- Admin: Full access to all tickets and assignment functions.
- Developer: Full access to their own dashboard and tickets.

Authentication is handled securely through the Flask session and password hashing.

The login page serves as the secure entry point for both admin and developer roles. Each user is required to input a valid username and password. Upon submission, the system authenticates the credentials using bcrypt password hashing and Flask session handling to ensure secure login operations. Depending on the user's role retrieved from the database, the system redirects the user to the corresponding dashboard—admin or developer.

An error message is shown if incorrect credentials are provided. This helps prevent unauthorized access while offering user-friendly feedback. The login page is shared by both administrators and developers; however, general users are not required to log in to submit

4.4.3 Public Ticket Submission

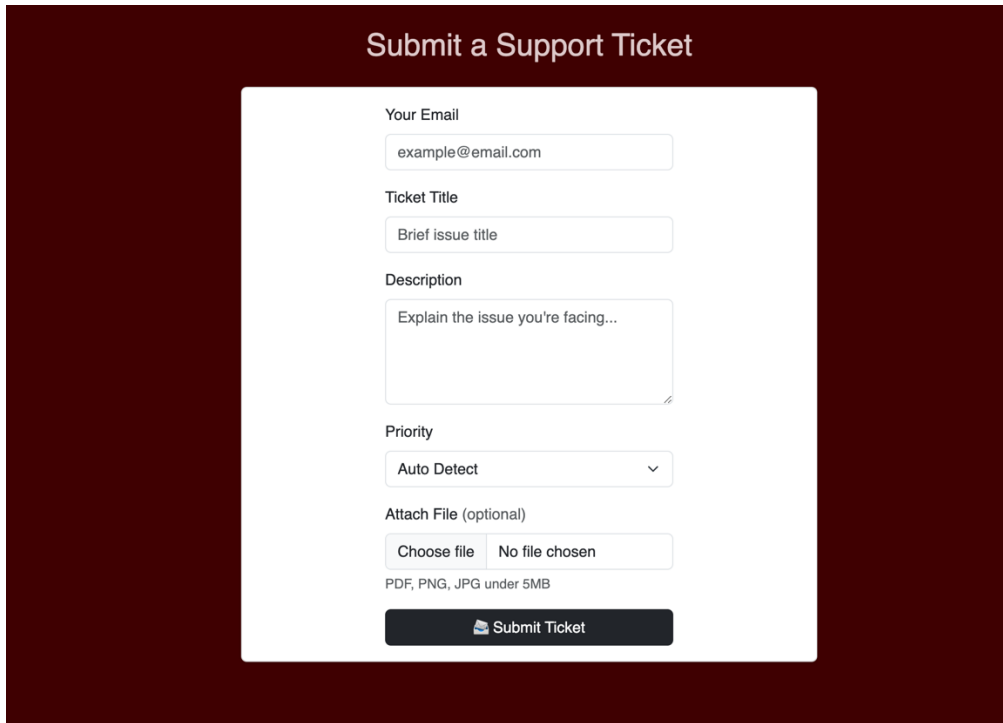
The image shows a web form titled "Submit a Support Ticket" set against a dark red background. The form itself is white and contains several input fields. At the top, it asks for "Your Email" with a text box containing "example@email.com". Below that is the "Ticket Title" field with the placeholder text "Brief issue title". The "Description" field is a larger text area with the placeholder "Explain the issue you're facing...". The "Priority" field is a dropdown menu currently showing "Auto Detect". Underneath is the "Attach File (optional)" section, which includes a "Choose file" button and a "No file chosen" button. Below these buttons, it specifies "PDF, PNG, JPG under 5MB". At the bottom of the form is a dark grey button with a paper icon and the text "Submit Ticket".

Figure 4.3 Public Ticket Submission Page

The Public Submit Ticket page is designed for general users of the Benadigital CRM system who do not require login access. This form allows users to submit an issue or inquiry directly to the support team. The form collects key information such as the user's email, ticket title, issue description, selected priority level (Low, Medium, High), and optional file attachments (e.g., screenshots or documents).

Upon submission, the system saves the ticket in the database and automatically triggers an email notification to the admin, informing them of the new request. The user is then presented with a confirmation message stating that the ticket was successfully submitted and will be reviewed shortly. This process allows for a seamless user experience, ensuring that anyone encountering an issue can easily request help without account registration.

Unauthenticated users can submit tickets using a clean form interface. Required fields include:

- Email
- Ticket Title
- Description
- Priority
- Optional File Upload

- Gets stored in the database
- Triggers an automatic email notification to the admin

4.4.4 Admin Dashboard

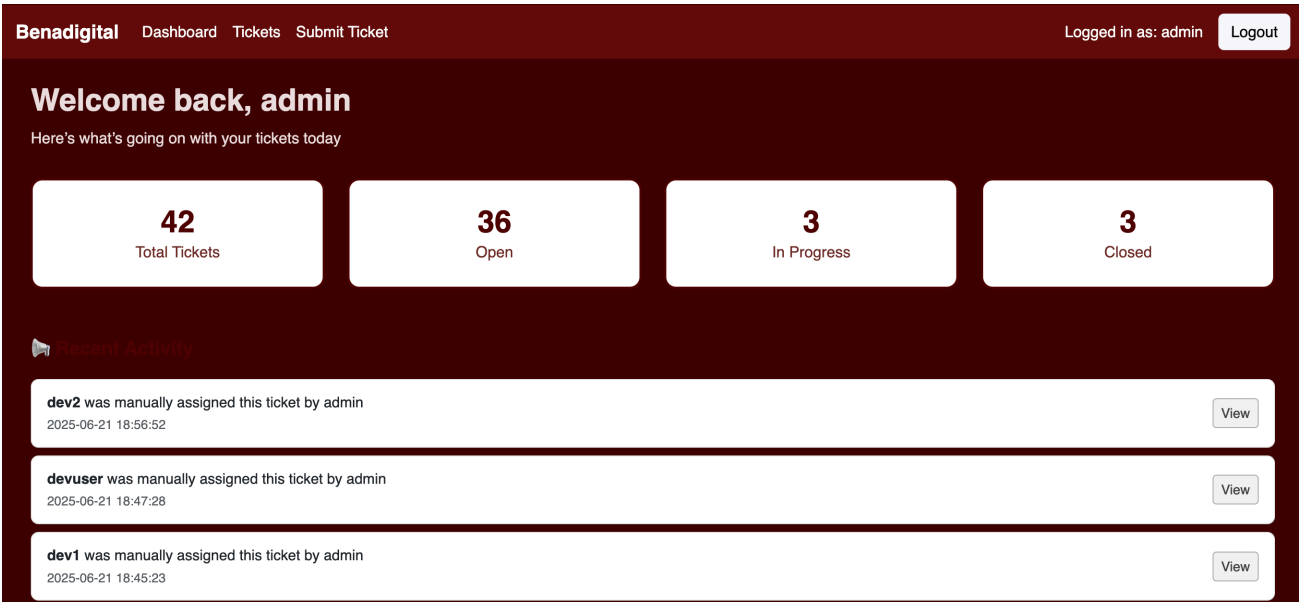


Figure 4.4: Admin Dashboard

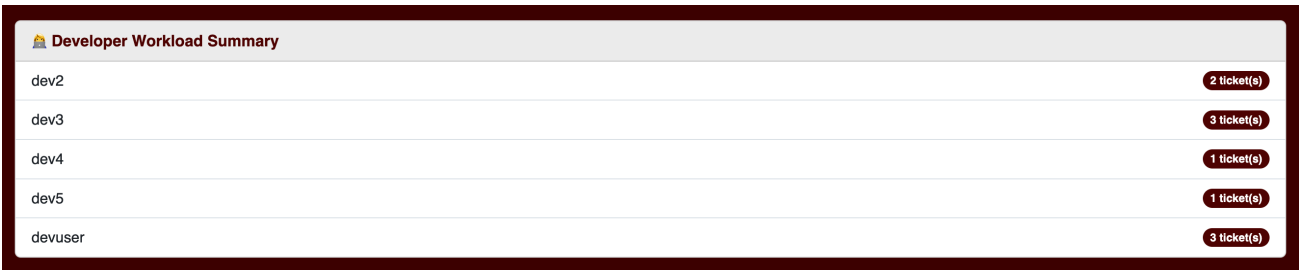


Figure 4.5 Developer Workload Summary

The dashboard page acts as the central hub for both administrators and developers once they log into the system. It displays essential summaries and navigation options relevant to their responsibilities.

For administrators, the dashboard provides a comprehensive view of the entire ticketing ecosystem. This includes status counters for all tickets categorized into Open, In Progress and Closed, offering immediate insight into ticket distribution.

Additionally, the admin dashboard features an “Activity Log” which records all significant events within the system. These include ticket submissions, status changes, developer assignments, and posted comments. Each log entry includes a timestamp and the

Below this, a section labelled “Developer Workload Summary” presents a list of developers along with the number of tickets currently assigned to each, helping admins assess and balance workloads efficiently.

Admin users can:

- View all tickets
 - Access dashboard
 - Search tickets by creator, title, priority, and status
 - Filter by status or priority
 - Assign tickets to developers
 - Track ticket logs
 - View developer activity logs
 - View developer workload summary
 - Edit assigned tickets
 - Update ticket status (e.g., Open, In Progress, Closed)
 - Add and post comments related to each ticket
 - Filter tickets based on:
 - Title
 - Creator
 - Status
 - Priority
- Sorting options include: Priority, Newest, Eldest, Status
- Submit internal tickets
 - View detailed ticket information

Each ticket also allows the admin to add notes for context.

4.4.5 Developer Dashboard

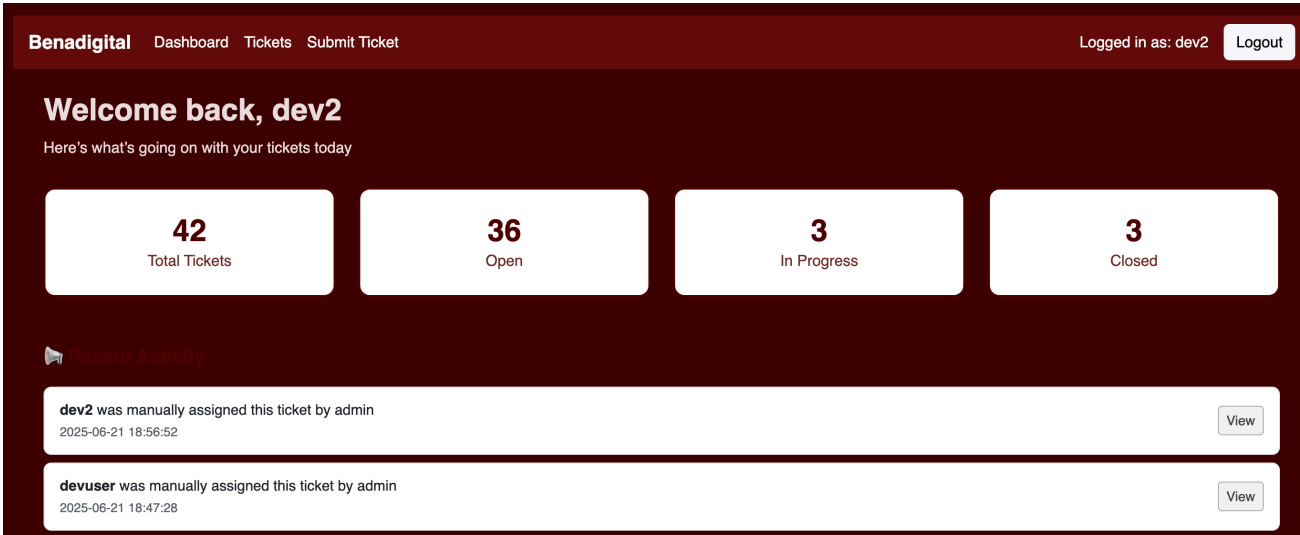


Figure 4.6 Developer Dashboard

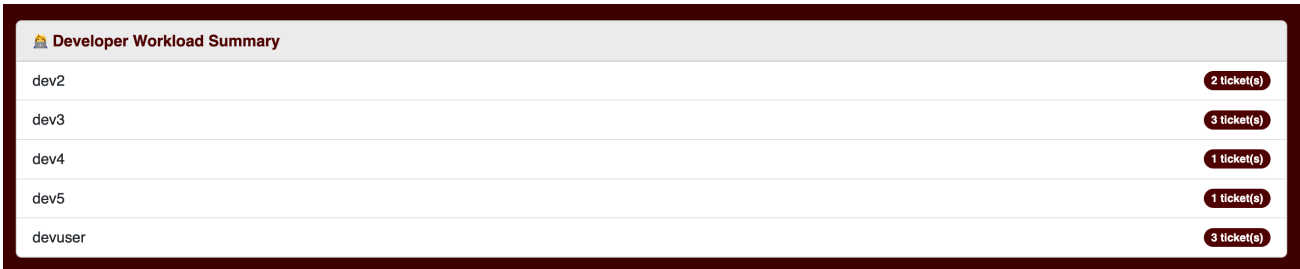


Figure 4.7 Developer Workload Summary

For developers, the dashboard displays their overall ticket statistics (e.g., in progress, open, closed) and a filtered activity log that shows only their interactions. This ensures that each developer stays updated on their current assignments and past actions.

Below this, a section labelled “Developer Workload Summary” presents a list of developers along with the number of tickets currently assigned to each, helping developer team members keeping track on the tickets assigned to them and claim unassigned ticket if they desired.

Developers see:

- A personal dashboard with total, open, in-progress, and closed ticket counters
- Recent activity logs
- Ticket viewing capabilities (View button)
- Ticket claiming functionality (Claim Ticket button)

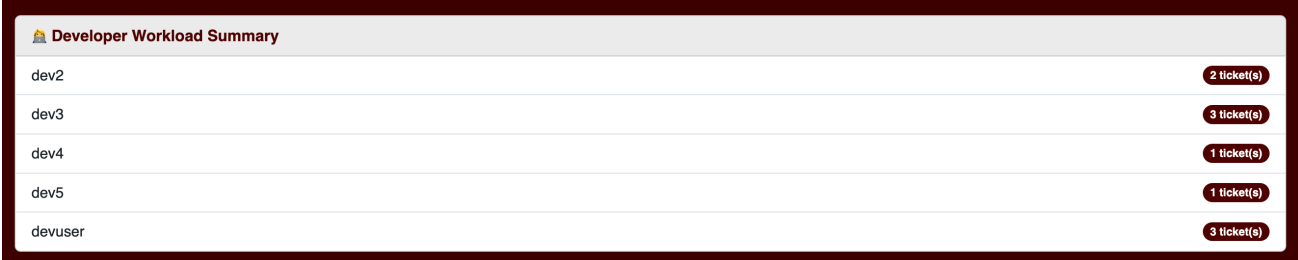
Developers can:

- Access the developer dashboard
- View own activity logs
- View workload summary
- Assign tickets to themselves (claim unassigned tickets)
- Edit tickets that are assigned to them
- Update ticket status during resolution
- Add and post comments on tickets
- Filter and sort tickets using the following parameters:
 - Title
 - Creator
 - Status
 - Priority

Sorting options: Priority, Newest, Eldest, Status

- Submit internal tickets (if needed)
- View detailed ticket information

4.4.6 Developer workload summary



Developer Workload Summary	
dev2	2 ticket(s)
dev3	3 ticket(s)
dev4	1 ticket(s)
dev5	1 ticket(s)
devuser	3 ticket(s)

Figure 4.8 Developer Workload Summary

A unique section titled “Developer Workload Summary” displays how many tickets are currently assigned to each developer.

4.4.7 Developer and Admin Ticket Page

Benadigital Dashboard Tickets Submit Ticket Logged in as: dev2 Logout							
All Tickets Search Title or creator Status All Priority All Sort by Newest Apply							
ID	Title	Status	Created By	Date	Note	Action	Priority
42	couldnt log in	Open	guest:adlin@test.com	2025-06-22 01:15:23		View	Low
41	domain prior engagement	Open	guest:nelysa@benadig	2025-06-22 00:33:46		View	Low
40	meovv	Open	admin	2025-06-21 17:02:58	Assigned to you	View	Low

Figure 4.9 Developer and Admin Ticket Page

The Ticket Page serves as a central workspace for both developer and admin users, providing them with a structured and interactive table that lists all tickets in the system. While both roles share a similar layout, the access level and functions slightly differ depending on the role.

For administrators, the ticket page displays every ticket submitted to the system, regardless of its status or assignment. The page includes filters and sorting options that allow admins to quickly narrow down the tickets based on title, creator email, status (e.g., Open, In Progress, Closed), priority (High, Medium, Low), and sort order (Newest, Eldest, Priority). Admins can view, edit, assign developers, update ticket status, and leave comments on each ticket.

Developers, on the other hand, are presented with a more focused view. They only see tickets that are unassigned, already assigned to them, or available for claiming. This helps minimize distractions and ensures developers only interact with relevant tasks. Developers can use the same filters and sort options to organize their workload. For unassigned tickets, developers can click “Claim Ticket” to take ownership. For assigned tickets, they can update the status, post internal comments, or view ticket history.

4.4.8 Developer Ticket Detail Page (unassigned ticket)

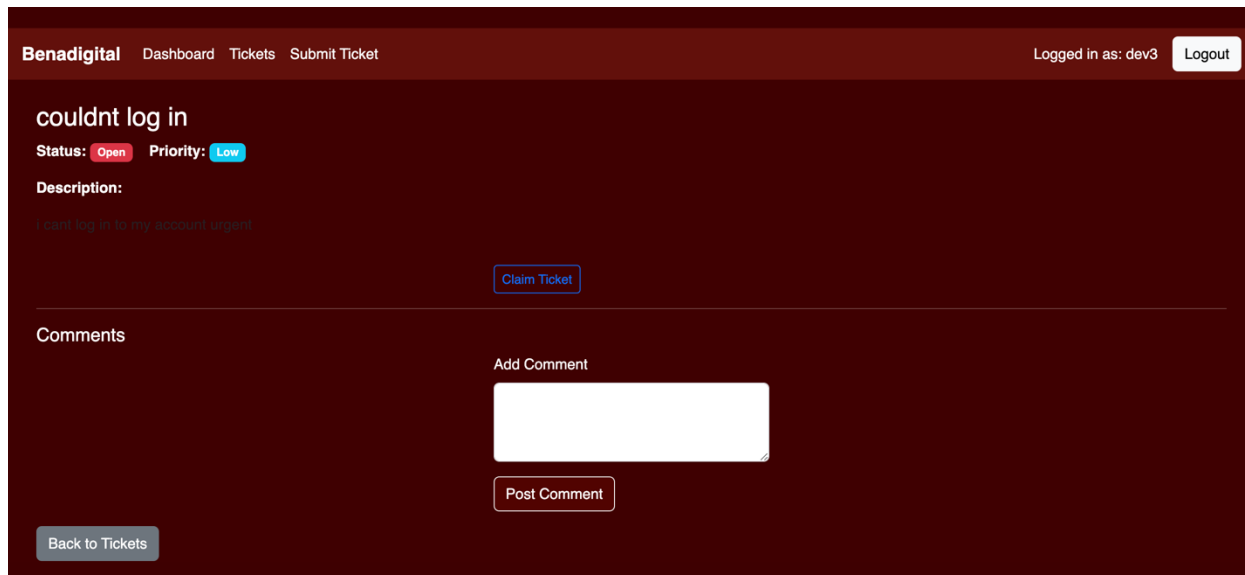


Figure 4.10 Developer Ticket Detail Page (unassigned ticket)

When a developer clicks on an unassigned ticket from their ticket list, they are brought to the Ticket Detail Page for that ticket. This page presents full details about the issue, including the title, issue description, priority, and current status (default: Open).

Since the ticket is still unassigned, the developer is given the option to claim it. A prominent “Claim Ticket” button is displayed at the top or bottom of the page. Once clicked, the system updates the assignedTo field in the database to reflect the developer’s ID, effectively assigning the ticket to them. After claiming, the ticket becomes visible under their assigned ticket list, and its status can then be updated.

4.4.9 Developer Ticket Detail Page (assigned ticket)

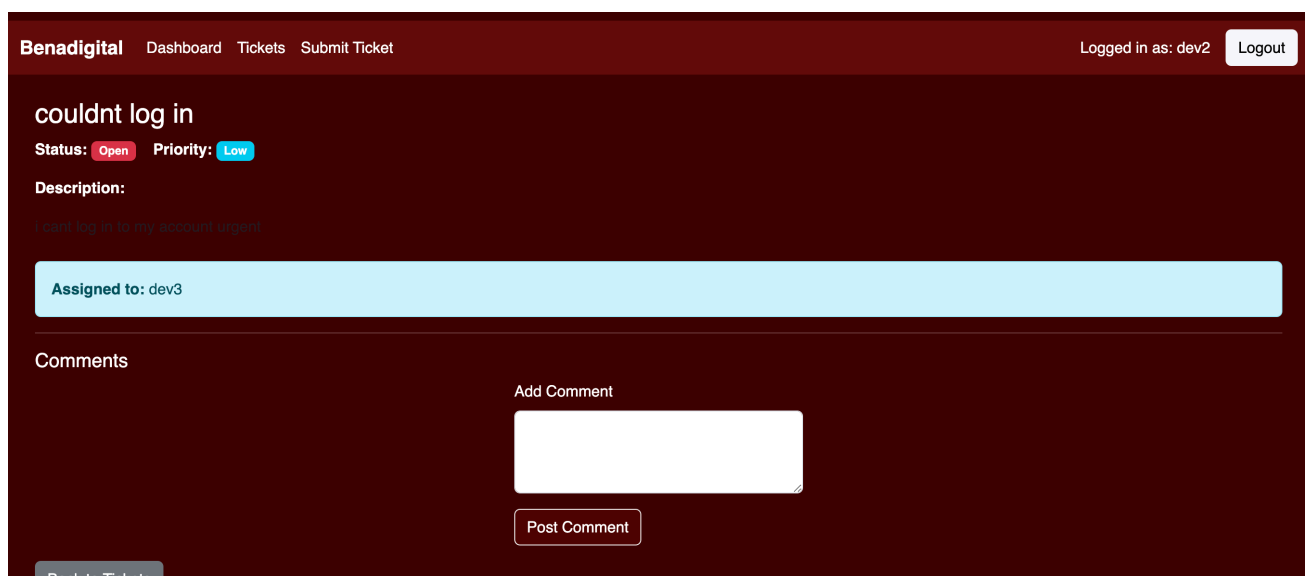


Figure 4.11 Developer Ticket Detail Page (assigned ticket)

This page appears when a developer opens a ticket that has already been assigned to another developer. If the ticket is assigned to another developer, the view is read-only. The system disables edit access but able the commenting features for developer to discuss insights which each other or mainly to see progress updates on the ticket. The developer can only view the ticket details such as title, description, priority, current status, and assignment.

4.4.10 Developer Ticket Detail page (own assigned ticket)

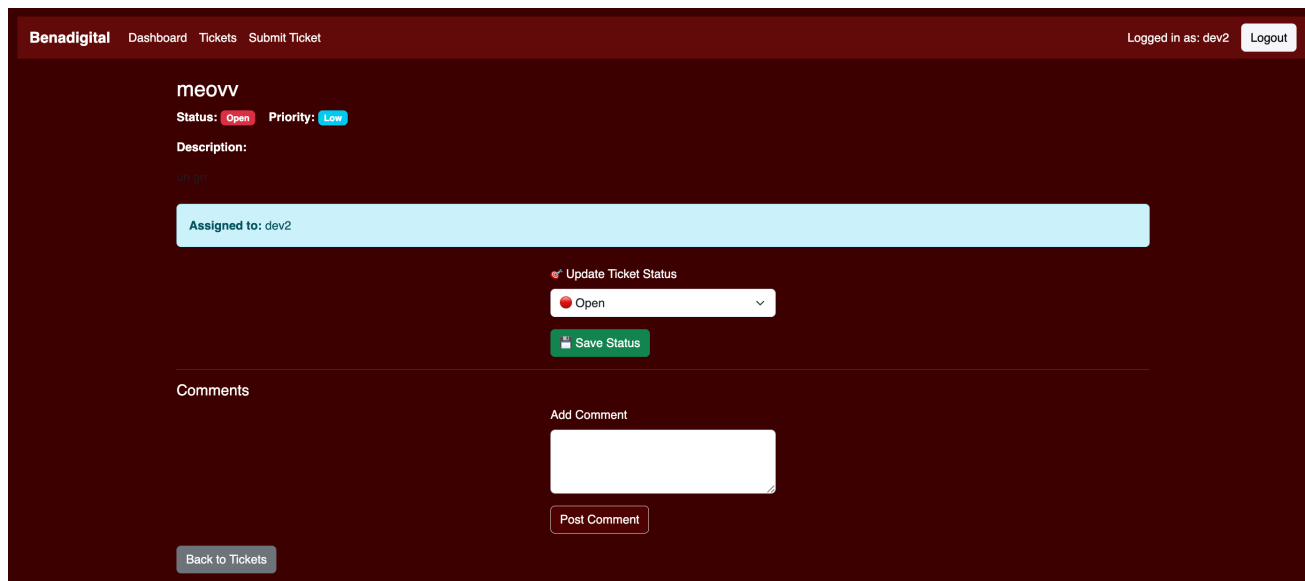


Figure 4.12 Developer Ticket Detail Page (own assigned ticket)

For tickets that have been specifically assigned to the logged-in developer, the detail page is fully interactive. Developers can see the ticket metadata and a comment history panel that tracks all internal communications and updates. Developers can post comments, add progress notes, and mark ticket stages (In Progress, Open, or Closed). This ensures that only the assigned developer can make changes, which improves data integrity and accountability in ticket resolution.

This page also includes a "Ticket History Log" section that outlines every status update or comment made on the ticket. This feature provides transparency and serves as a log of the resolution journey. Additionally, developers can escalate the ticket (if needed) or request reassignment through an internal note, depending on the admin's workflow rules.

4.4.11 Admin Ticket Detail Page

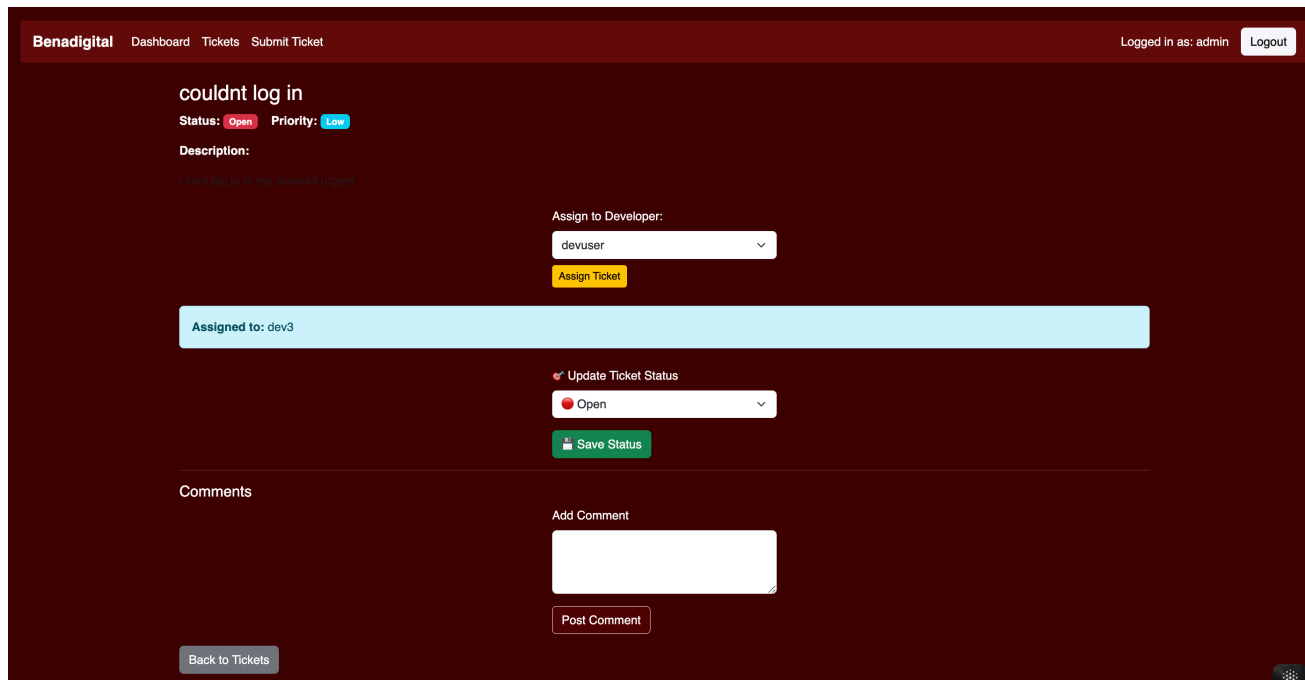


Figure 4.13 Admin Ticket Detail Page

The Admin Ticket Detail Page is one of the most feature-rich views in the system. When an administrator opens a ticket, they gain access to full ticket information along with actionable tools. Admins can reassign the ticket to another developer, modify the ticket's priority or status, and post internal comments visible to the assigned developer.

In addition, the page contains logs showing all activities performed on the ticket on who created it, who assigned it, and what changes were made over time. This historical log aids in diagnosing any delays or bottlenecks in the workflow. Admins can also view attachments submitted with the ticket and download them if necessary.

To improve manageability, the admin detail page includes dropdown filters and visual indicators (such as color-coded tags for priority and status), making it easier to process and act upon tickets efficiently. These capabilities ensure admins have complete control and visibility across the entire ticket lifecycle.

4.5 Database Design

The database schema consists of the following primary tables:

- users: Stores admin and developer credentials along with their roles
- tickets: Stores ticket data including creator, status, priority, and description

(*) Records comments posted on tickets

- workload: Summarized ticket count for each developer

All relationships are maintained via foreign keys and auto-incremented IDs.

4.6 Security Considerations

- Passwords are securely hashed
- Role-based access control is implemented
- File uploads are restricted to PDF, PNG, JPG formats and below 5MB
- Sessions are handled with Flask's secure session mechanism

4.7 Tools Used

The following tools were utilized:

Tool/Technology	Purpose
Python (Flask)	Backend logic and routing
MySQL	Data storage
Bootstrap	Front-end responsiveness
HTML/CSS	Layout and UI
PythonAnywhere	Hosting and deployment
Terminal/Command Line	Run Flask and database connections
Flask-Mail	Email notification to Admin
phpMyAdmin (via XAMPP)	Locally manage database
Localhost (via XAMPP & Terminal)	Local Development and testing

4.8 Summary

This chapter outlined the system design and implementation of the Benadigital Ticketing System. Through role-based navigation, public and internal ticket management, workload summaries, and real-time commenting, the system aims to streamline issue tracking and task assignment for service-based businesses.

Chapter 5: System Testing and Evaluation

5.1 Introduction

This chapter outlines the testing strategies applied to the Benadigital Ticketing System and presents a thorough evaluation of its functionality, usability, and system performance. Testing is crucial in ensuring the reliability and stability of the system, as it validates whether the final product meets the specified functional requirements and is free from major bugs or usability issues. The chapter also includes feedback from real users, particularly developers and administrative personnel who interact with the ticketing system during day-to-day operations.

Testing was carried out to validate both the functional and non-functional requirements of the system. The goal is to ensure that all modules including User, Admin, and Developer perform reliably and meet user expectations. Both unit-level testing and usability testing were conducted during development and post-deployment.

5.2 Functional Testing

The testing phase used a combination of Black Box Testing and User Acceptance Testing (UAT) to validate different modules of the system.

- Black Box Testing was performed by simulating various user actions without viewing the underlying source code. Each function such as login, ticket submission, filtering, assignment, and comment posting was tested using valid and invalid inputs to observe system behaviour.
- User Acceptance Testing was conducted by involving the actual users — primarily admin, developer, and general users to test the system in a real operational setting. Feedback was collected to identify any usability issues, feature improvements, and confirm satisfaction with system responses.

5.2.1 Functional Testing

Functional testing was carried out through unit testing of individual system modules to verify correct behaviour under various conditions.

5.2.2 Unit Testing

Unit testing was performed manually and through black box testing techniques on each of the core modules.

5.3 Test Cases and Results

Below is a summary of selected critical test cases and their outcomes:

Test Case ID	Feature Tested	Expected Result	Actual Result	Status
TC01	Role Selection Navigation	Redirects to appropriate login or form	Success	Passed
TC02	Admin Login (valid/invalid)	Grants/rejects access accordingly	Success	Passed
TC03	Developer Login	Displays developer dashboard	Success	Passed
TC04	Ticket Submission (Public)	Stores ticket and notifies admin	Success	Passed
TC05	Ticket Assignment	Developer assigned correctly	Success	Passed
TC06	Developer Claim Ticket	Claim updates assignedTo field	Success	Passed
TC07	Activity Log Display	Recent activities shown accurately	Success	Passed
TC08	Workload Summary Count	Shows count per developer	Success	Passed
TC09	Ticket Filtering	Returns filtered results	Success	Passed
TC10	Session Logout	Logs user out and clears session	Success	Passed

Table 5.1 Test Case Table

5.2.1.1 Test Case on User Module

The User module was tested for the ability to submit a ticket, receive a submission confirmation, and trigger email notifications to the admin.

Test Case	Expected Result	Outcome	Status
Submit ticket with valid data	Ticket saved to database and email sent	Success	Passed
Submit ticket with missing fields	Error message shown	Success	Passed
Submit ticket with file upload	File stored correctly	Success	Passed

5.2.1.2 Test Case on Admin Module

Admin users were tested for their ability to log in, view and assign tickets, update status, post comments, and view and manage workload summaries.

Test Case	Expected Result	Outcome	Status
Login with correct credentials	Redirect to dashboard	Success	Passed
View all tickets	All existing tickets shown	Success	Passed
Assign ticket to developer	Ticket updated with developer ID	Success	Passed
Filter tickets	Filter returns correct results	Success	Passed
View workload summary	Correct ticket count per developer	Success	Passed

Table 5.2 Test Case Table (admin)

5.2.1.3 Test Case on Developer Module

Developers were tested on their access to assigned tickets, ability to claim unassigned ones, update ticket status, and post comments.

Test Case	Expected Result	Outcome	Status
Login and view dashboard	Display all tickets	Success	Passed
Claim ticket	Ticket marked as assigned	Success	Passed
Update ticket status	Status updated in DB	Success	Passed
Post comment	Comment stored & displayed	Success	Passed

Table 5.3 Test Case Table (developer)

All critical test cases have passed successfully without major bugs. Minor UI layout issues were identified during early testing but were corrected before final deployment.

5.3.1.1 Usability Testing for Admin

Admin users found the dashboard to be informative and easy to navigate. Features like ticket filters, activity log, and workload summary made ticket delegation more manageable. The ability to sort and search tickets enhanced productivity.

Key feedback:

- The filters worked smoothly and allowed faster ticket sorting.
- Admins appreciated the real-time update of ticket assignments.

5.3.1.2 Usability Testing for Developer

Developers found the UI intuitive, especially the “Claim Ticket” and Notes “Assigned To Me” sections. Ticket details were clearly structured, and comments enabled team collaboration.

Key feedback:

- Developers liked seeing their task counts under workload summary.
- The “Claim Ticket” feature gave them autonomy over their workflow.
- All assigned and claimed tickets were easy to track.

5.4 Test Environment

All testing activities for this project were conducted locally on a development machine using Flask’s built-in development server. The system was not deployed to a live server at the time of testing; instead, localhost was utilized to simulate real-time system behaviour and validate all functionalities. This environment allowed for isolated, secure, and efficient testing of the system's backend and frontend components.

The database used during testing was MySQL, running locally and integrated with the Flask application. The user interface was designed using HTML, CSS, and Bootstrap, ensuring responsiveness across screen sizes during manual inspection.

Tools and techniques used for testing include:

- Browser console logs – to identify and resolve front-end issues such as JavaScript errors or CSS layout glitches.
- Terminal output – to monitor Flask server logs and detect any routing, session, or application errors in real time.
- SQL queries – to directly inspect and validate the accuracy of data operations such as ticket creation, updates, and user authentication.
- Flask route checks – to confirm expected behaviour for login sessions, role-based redirection, and data access security.

This setup provided a reliable environment for both functional and non-functional testing prior to deployment.

5.5 User Feedback and Evaluation

Upon completing the system testing phase, a User Acceptance Testing (UAT) survey was conducted involving a total of 8 respondents from different roles including Admins, Developers, and General Users as the one who participated in hands-on testing of the Benadigital Ticketing System for Issue Management.

The objective of this evaluation was to gather comprehensive feedback regarding the functionality, usability, and overall experience of using the system.

Section 1: Role

Section 1: Role
8 responses

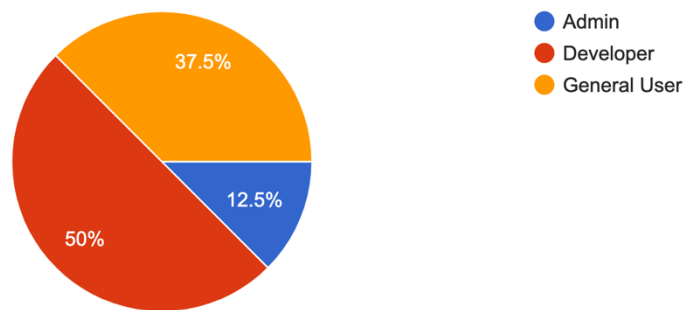


Figure 5.1 Role

Based on the survey data, 50% of respondents were Developers, 37.5% were General Users, and 12.5% were Admins. This diversity ensured a balanced evaluation from all perspectives.

Section 2: System Functionality

How would you rate the ease of logging into the system?
8 responses

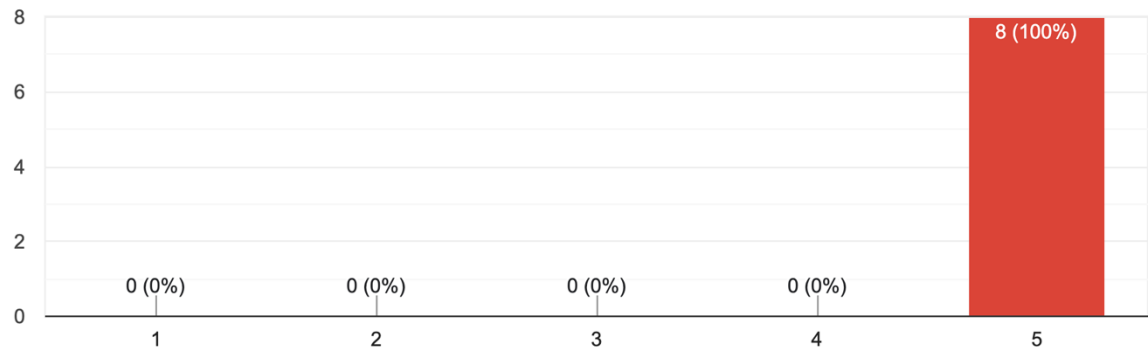


Figure 5.2 System Login Functionality Rate

Was the Role Selection Page clear and easy to navigate?
8 responses

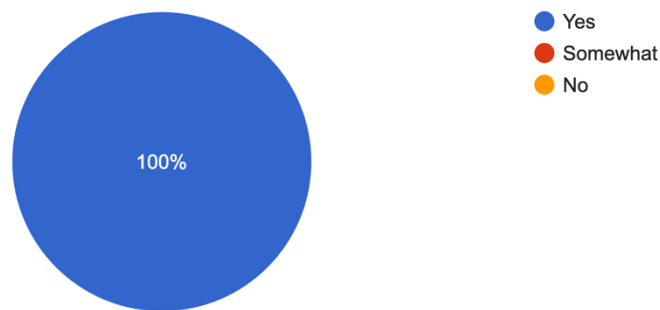


Figure 5.3 Role Selection Page Navigation Functionality

How would you rate the Ticket Submission process?

8 responses

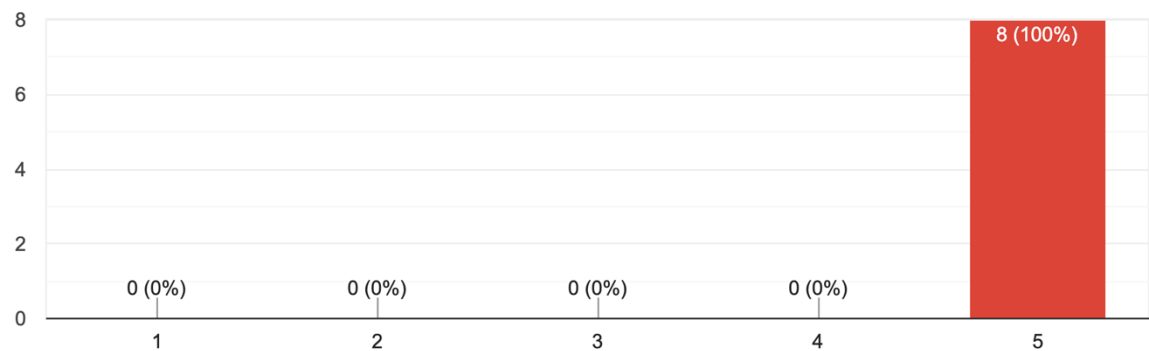


Figure 5.4 Ticket Submission Process Functionality

Were you able to find and view tickets without difficulty?

8 responses

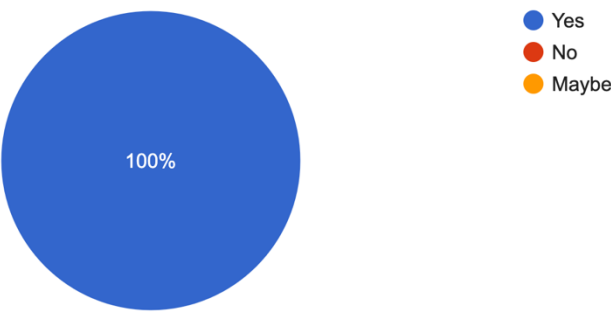
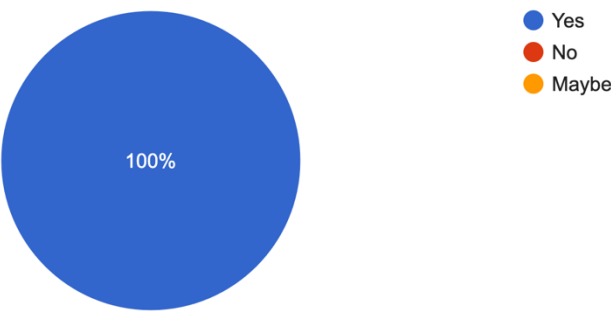


Figure 5.5 View Ticket Page Functionality

Did the system behave as you expected?

8 responses



All participants rated the login process as highly accessible, indicating that authentication and role-based redirection were smooth and intuitive.

Additionally, All of the users confirmed that the Role Selection Page was clear and easy to navigate, which validated the effectiveness of the entry flow design.

Ticket submission functionality received perfect scores, with all 8 users rating the process a 5 out of 5. Furthermore, all respondents indicated they were able to find and view tickets without difficulty, and the system behaved as expected throughout their usage.

Section 3: Usability and Design

How would you rate the overall user interface (layout, readability, buttons)

8 responses

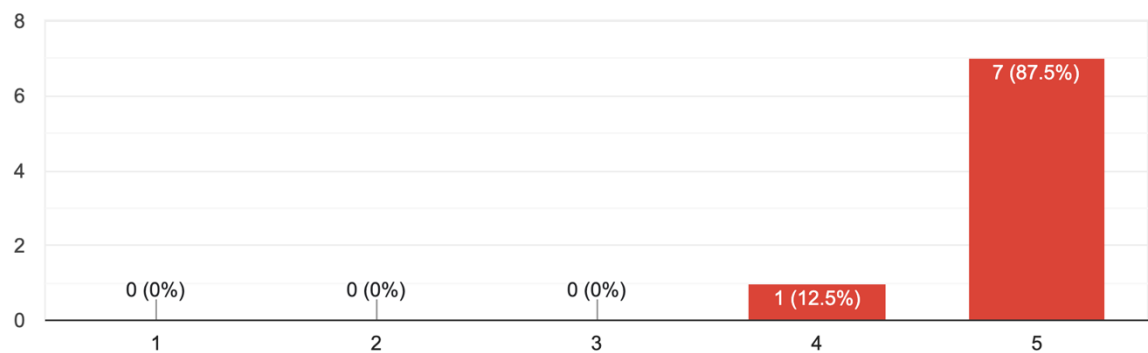


Figure 5.7 User Interface Design Rate

Do you find the Developer Workload Summary and Activity Log helpful?

8 responses

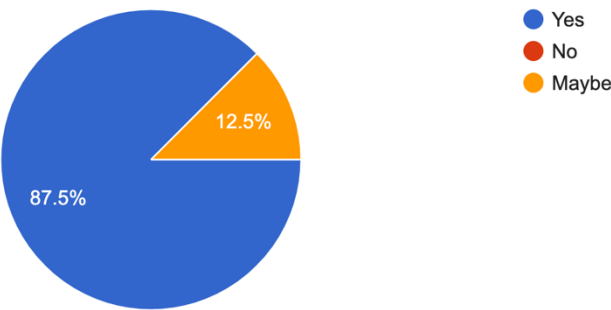


Figure 5.8 Developer Workload Summary and Activity Log Usability

Were the ticket filtering and sorting features useful?

8 responses

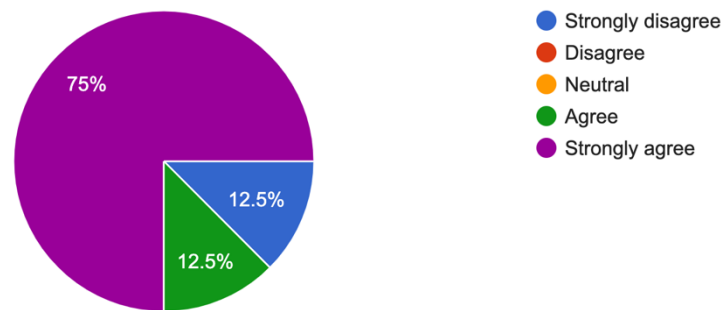


Figure 5.9 Ticket Filtering and Sorting Usability

The overall user interface including layout, button readability, and responsiveness receives 87.5% of users rate concluding the design as excellent, while 12.5% gave it a good score (4 out of 5). This reflects strong satisfaction with the visual and interaction design of the system.

The Developer Workload Summary and Activity Log were also highlighted positively, with 87.5% agreeing they were helpful in understanding task distribution and recent activity. Meanwhile, 75% of respondents strongly agreed that the ticket filtering and sorting features were useful, and another 25% responded with neutral to agree demonstrating high appreciation for this productivity-enhancing functionality.

Section 4: Open Feedback

Any suggestions for improvement?

5 responses

No
Button color is a bit meh
Make it into mobile applications
Make it multi language

Figure 5.10 Suggestion For Improvement

Among the open-ended feedback, some users suggested enhancements such as:

- Improving button colors for better visual hierarchy
- Expanding the system into a mobile application for accessibility
- Adding multilingual support to accommodate a more diverse user base

These suggestions are valuable and will be considered in future iterations of the system.

Section 5: Readiness and Recommendation

Based on your experience, do you think the system is ready for real-world use?

8 responses

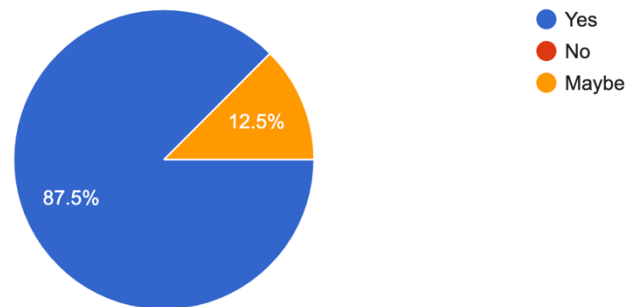


Figure 5.11 System Readiness for Real-World Deployment

Would you recommend this system to your colleagues?

8 responses

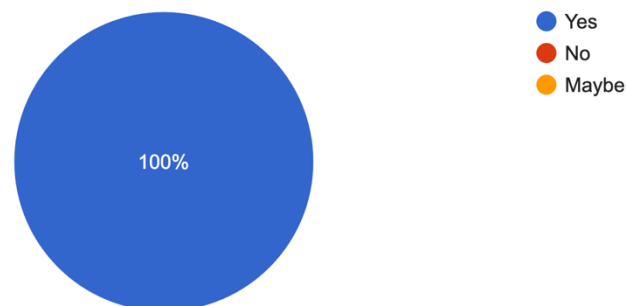


Figure 5.12 System Recommendation To Public User

A strong 87.5% of participants felt that the system is ready for real-world deployment, while all of them stated they would recommend it to their colleagues. This indicates a high level of confidence and satisfaction from testers across different user roles.

Overall, the system was deemed easy to navigate, responsive, and aligned with the original project objectives. The performance was stable on desktop views.

5.6 Conclusion

Testing was conducted thoroughly across all roles and system modules. Both functional (unit, integration, functional (usability) testing) and the system was tested on desktop. All

specified requirements for Benadigital's internal ticket handling process. Feedback collected during usability testing confirms that the system is user-friendly and ready for operational deployment.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

The development of the Benadigital Ticketing System has successfully fulfilled its primary objective of streamlining internal issue tracking and resolution for Benadigital's technical support ecosystem. Designed to cater to three core user roles which is general users, administrators, and developers , the system has introduced an organized workflow for ticket submission, delegation, and resolution.

Through comprehensive planning, design, and implementation phases, the system now supports:

- A role-based access mechanism to differentiate user privileges
- Public ticket submission without login
- Secure login system for admin and developer roles
- Ticket assignment and claim functionalities
- Real-time activity logs and workload monitoring for developers
- Comment and status updates for effective communication
- Advanced filtering and sorting of tickets for efficient ticket management

The system design was supported by Flask as the web framework, MySQL as the backend database, and PythonAnywhere as the deployment environment (though final testing was done on localhost). Bootstrap and custom CSS were used to create a responsive and user-friendly interface.

All major functional and non-functional requirements were tested and validated successfully. Usability testing showed positive reception from users, particularly developers and administrative personnel who used the platform during real test scenarios.

The project also demonstrates how a lightweight framework like Flask can be utilized effectively for enterprise-grade ticket tracking systems when supported by well-planned database architecture and role-based logic.

6.2 Project Strengths

Several key strengths were observed during the course of this project:

- Intuitive UI with simple role selection
- Real-time activity log for transparency
- Developer workload summary to assist in fair task distribution
- Ticket claiming feature to reduce idle tickets
- Use of modular Flask routes, enabling future scalability

6.3 Limitations

Despite its strengths, the system still has certain limitations:

- Email notifications were only simulated due to limited access to a live SMTP service during localhost testing
- Lack of user registration or user profile update modules
- Admin and developer share the same login page; future improvements can further personalize their experience
- No in-system alert or notification system (e.g., pop-up alerts or toast messages)

6.4 Future Enhancements

The following enhancements that can be expected for future development:

1. Email Service Integration to deploy a live email service to fully automate ticket notifications to admins and developers.
2. Notification Center as in adding a notification system within the dashboard for new ticket assignments, comment updates, or claim confirmations.
3. Ticket History to enable returning users to track ticket history and responses.
4. Ticket Analytics Dashboard that build charts and metrics summarizing ticket load, average resolution time, ticket categories, and developer performance.

References

Chhibber, M., Mittal, R., & Singh, A. (2023). AI-powered chatbot for improved customer feedback management using NLP. 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1–5.

<https://doi.org/10.1109/ICCCNT56998.2023.10308364>

M. G. C. P, Srivastava, A., Chakraborty, S., Ghosh, A., & Raj, H. (2021). Development of information technology telecom chatbot: An artificial intelligence and machine learning approach. 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 216–221. <https://doi.org/10.1109/ICIEM51511.2021.9445354>

Castro, E., Calvo, H., Kolesnikova, O., & Castro, C. (2023). Analysis of emotions in speech acts for chatbots: An overview and a model proposal. 2023 IEEE Symposium Series on Computational Intelligence (SSCI), 1467–1471.

<https://doi.org/10.1109/SSCI52147.2023.10371792>

Lalwani, T., Bhalotia, S., Pal, A., Rathod, V., & Bisen, S. (2018). Implementation of a chatbot system using AI and NLP. International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 6(3). <https://doi.org/10.2139/ssrn.3531782>

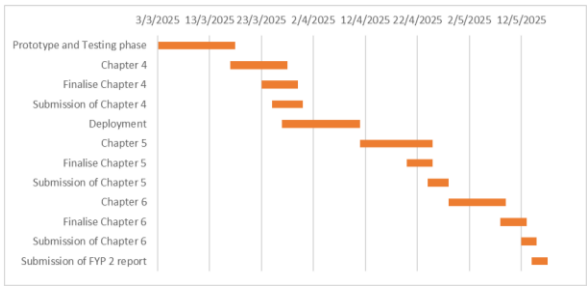
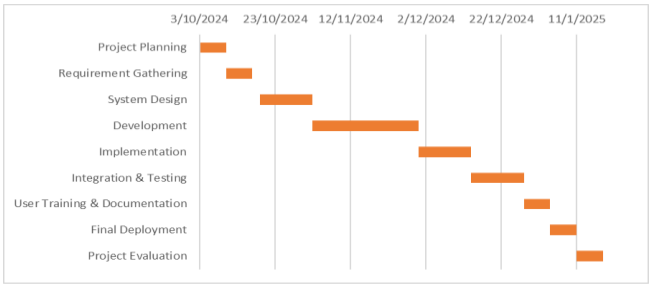
Sadhu, A. K. R., Parfenov, M., Saripov, D., Muravev, M., & Sadhu, A. K. R. (2024). Enhancing customer service automation and user satisfaction: An exploration of AI-powered chatbot implementation within customer relationship management systems. Journal of Computational Intelligence and Robotics, 4(1), 103–123. Retrieved from

<https://nucleuscorp.org/jcir/article/view/249>

Anaya, L., Braizat, A., & Al-Ani, R. (2024). Implementing AI-based chatbot: Benefits and challenges. Procedia Computer Science, 239, 1173–1179.

<https://doi.org/10.1016/j.procs.2024.06.284>

Appendix A



Appendix B

Final Year Project - Benadigital Ticketing System For Issue Management

🎯 Benadigital Ticketing System – UAT Feedback Form

Thank you for participating in the User Acceptance Testing (UAT) of our Benadigital Ticketing System. Your feedback is valuable to help us improve the system before final deployment.

* Indicates required question

Section 1: Role *

- ☐ Admin
- ☐ Developer
- ☐ General User

Final Year Project - Benadigital Ticketing System For Issue Management

* Indicates required question

Section 2: System Functionality

How would you rate the ease of logging into the system? *

	1	2	3	4	5	
Very Difficult	☐	☐	☐	☐	☐	Very Easy

Was the Role Selection Page clear and easy to navigate? *

- ☐ Yes
- ☐ Somewhat
- ☐ No

How would you rate the Ticket Submission process? *

	1	2	3	4	5	
Very confusing	☐	☐	☐	☐	☐	Easyyyy

Were you able to find and view tickets without difficulty? *

- ☐ Yes
- ☐ No
- ☐ Maybe

Did the system behave as you expected? *

- ☐ Yes
- ☐ No
- ☐ Maybe

Back

Next

Clear form

Section 3: Usability & Design

How would you rate the overall user interface (layout, readability, buttons) *

	1	2	3	4	5	
Poor	☐	☐	☐	☐	☐	Excellent

Do you find the Developer Workload Summary and Activity Log helpful? *

- ☐ Yes
- ☐ No
- ☐ Maybe

Were the ticket filtering and sorting features useful? *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

Section 4 - Open Feedback

Any suggestions for improvement?

Your answer

Section 5: Final

Based on your experience, do you think the system is ready for real-world use? *

- ☐ Yes
- ☐ No
- ☐ Maybe

Would you recommend this system to your colleagues? *

- ☐ Yes
- ☐ No
- ☐ Maybe