Group Project Log

Group Name:	Group 15

Group Members:	Darshit Mukeshbhai Dhameliya, Nisarg Sharadkumar Vaghela, Bhautik Rameshbhai Koshiya, Dhruvik Maheshbhai Kakadiya, Kuldeep Rajeshbhai Gajera, Gowri Prashanth Kanagaraj

Deliverable:	"IssueStack" project report and front-end code (via GitLab)

Group Member Name	**Work Done (%)**
Darshit Mukeshbhai Dhameliya	16.7
Nisarg Sharadkumar Vaghela	16.7
Bhautik Rameshbhai Koshiya	16.7
Dhruvik Maheshbhai Kakadiya	16.7
Kuldeep Rajeshbhai Gajera	16.7
Gowri Prashanth Kanagaraj	16.7
Total:	**100%**

TECHNICAL REPORT

Project Group 15

ISSUESTACK

Modernize how to handle internal support requests

Members and Contributors

Darshit Mukeshbhai Dhameliya
B00953044
darshit.dhameliya@dal.ca

Nisarg Sharadkumar Vaghela
B00945738
nvaghela@dal.ca

Bhautik Rameshbhai Koshiya
B00955643
bh889463@dal.ca

Dhruvik Maheshbhai Kakadiya
B00962810
d.kakadiya@dal.ca

Gowri Prashanth Kanagaraj
B00942544
gowri.kanagaraj@dal.ca

Kuldeep Rajeshbhai Gajera
B00962793
kl210309@dal.ca

Faculty of Computer Science
Dalhousie University

09, April, 2024

ABSTRACT

IssueStack is a web application designed to modernize how we handle internal support requests in our organization. It provides a central place for employees to send in requests for help, whether it's with IT problems, HR matters, administrative questions, or payroll issues. IssueStack makes it easier for employees to communicate their needs and for us to solve problems quickly, which ultimately boosts morale and productivity.

KEYWORDS

ExpressJS, MERN, MongoDB, NodeJS, ReactJS, Service Analytics, Support Ticket Management, Service Ticket, Service Ticket Board, Support System

CONTENTS

1. INTRODUCTION

IssueStack is a newly developed application designed to modernize how we handle internal support requests in our organization. It provides a central place for employees to send in requests for help, whether it's with IT problems, HR matters, administrative questions, or payroll issues. IssueStack makes it easier for employees to communicate their needs and for us to solve problems quickly, which ultimately boosts morale and productivity.

The objective of IssueStack (Internal Support Ticket Management System) is to address the inefficiencies and communication gaps prevalent in the current process of handling internal support requests within the company. By providing a centralized and streamlined platform, IssueStack aims to enhance the efficiency and effectiveness of support ticket management across various departments such as IT, HR, Admin, and Payroll. The overarching purpose is to ensure timely resolution of internal issues, thereby fostering improved employee satisfaction and productivity.

1.1 Live Project URL

In this Section, you are expected to provide the URL from which your application can be accessed throughout the term:

Application URL: **https://csci5709-web-project.netlify.app/**

GitLab Repository URL: **https://git.cs.dal.ca/kanagaraj/csci-5708-grp-15**

2. BACKGROUND

2.1 Competitive Landscape

IssueStack distinguishes itself from competitors such as Zendesk [1] and Service-now [2] by offering a streamlined interface tailored specifically for internal support ticket management. Unlike other platforms, IssueStack enables users to categorize tickets by department, facilitating efficient routing and resolution.

Existing competitors for our application are Zendesk and Service-now.

2.2 Problem and Approach

The objective of IssueStack (Internal Support Ticket Management System) is to address the inefficiencies and communication gaps prevalent in the current process of handling internal support requests within the company. By providing a centralized and streamlined platform, IssueStack aims to enhance the efficiency and effectiveness of support ticket

management across various departments such as IT, HR, Admin, and Payroll. The overarching purpose is to ensure timely resolution of internal issues, thereby fostering improved employee satisfaction and productivity.

Specifically, the project aims to achieve the following objectives:

1. **Efficient Ticket Handling**: IssueStack endeavors to streamline the process of raising, assigning, and resolving support tickets, thereby minimizing response times and ensuring quick and effective problem resolution.

2. **Enhanced Communication**: Through IssueStack, clear and transparent communication channels will be established between employees and support departments. This will keep all stakeholders informed about the status and progress of their support requests, fostering better collaboration and understanding.

3. **Tracking and Analytics**: The system will provide robust tools for tracking and analyzing support ticket data. By leveraging analytics, trends, bottlenecks, and areas for improvement within the support process can be identified and addressed proactively.

4. **User-Friendly Interface**: IssueStack will be designed with an intuitive and user-friendly interface, ensuring that employees can easily submit and track their support tickets without the need for extensive training. This will promote adoption and usage across the organization, maximizing the system's effectiveness.

In summary, IssueStack aims to bridge the gap in internal support ticket management by offering a comprehensive solution that caters to the diverse needs of employees across different departments.

3. APPLICATION DETAILS

IssueStack is an Internal Support Ticket Management System designed to streamline and enhance the process of handling support requests within the company. It aims to provide an efficient and organized platform for employees to submit support tickets related to various departments such as IT support, HR support, Admin support, etc. The system prioritizes timely and effective resolution of internal issues to improve overall employee satisfaction and productivity.

Below are the must have features of our application:

1. **Authentication:** Implement a secure authentication system for users to access the support ticket management system, including both a user-friendly interface and API for backend integration.
2. **Profile Management/Settings:** Allow users to manage their profiles, change passwords, and update personal information, providing a seamless and personalized experience.
3. **Manage the Board:** Administrative functionality to add or delete teams within the system, ensuring flexibility in adapting to organizational changes.

4. **Team Management/Team Assignment:** Enable team leaders or administrators to efficiently manage team members, assigning or removing them as needed for effective support distribution.
5. **Raise Ticket:** Provide an intuitive user interface for employees to submit support tickets, capturing essential details to initiate the ticket resolution process.
6. **Ticket Assignment:** Backend functionality to automatically assign tickets to the appropriate department based on the issue, ensuring swift and accurate routing.
7. **Ticket Management:** Comprehensive ticket management features allowing support staff to prioritize, forward, and update the status of tickets, ensuring efficient resolution.
8. **Comments on Ticket:** Enable users and support staff to add comments to tickets, facilitating communication and maintaining a detailed record of the ticket resolution process.
9. **Notification:** Set up a notification system to keep users informed about ticket updates via email and in-app notifications, enhancing communication and transparency.

Below are the nice to have features of our application:

1. **Escalation:** Implement an escalation process where critical issues are automatically assigned to team leads, ensuring prompt attention to high-priority matters.
2. **Knowledge Base:** Create a user-friendly knowledge base with department-specific FAQs to empower users with self-help resources, reducing the volume of support tickets.
3. **Reporting and Analytics:** Implement a visual analytics dashboard displaying charts of ticket distribution by department, offering insights for informed decision-making and process optimization.

3.1 Target User Insights

IssueStack serves as a comprehensive support platform designed to assist all employees within a company across various domains. Its user base encompasses individuals seeking assistance with technical issues through IT Support, resolving personnel-related matters via HR Support, addressing general administrative queries with Admin Support, and resolving payroll-related concerns through dedicated Payroll Support. Whether an employee encounters a software glitch, seeks clarification on company policies, requires assistance with administrative tasks, or needs help with payroll-related inquiries, IssueStack aims to streamline the support process and provide timely resolutions. By catering to these diverse needs, IssueStack ensures that employees can efficiently access the necessary resources and expertise within the organization, fostering smoother operations and enhanced productivity across departments.

An SAP Software Developer encounters complex issues during work on SAP modules and relies on IssueStack to raise tickets, utilizing the Knowledge Base feature when necessary. A Support Department Team Lead oversees support ticket distribution and

resolution, relying on IssueStack's Team Management, Ticket Assignment, Reporting and Analytics, and Escalation features to ensure efficient support operations. A tech-savvy software developer values secure systems and efficient collaboration tools, relying on IssueStack's Ticket Management, Board Management, Notification, and Authentication features to streamline issue resolution. An IT Support Team Lead prioritizes effective communication and team management, utilizing features like Comments on Tickets, Notification, and Escalation within IssueStack to ensure smooth project delivery and collaboration among teams

3.1.1 User Personas and Intended Scenarios:

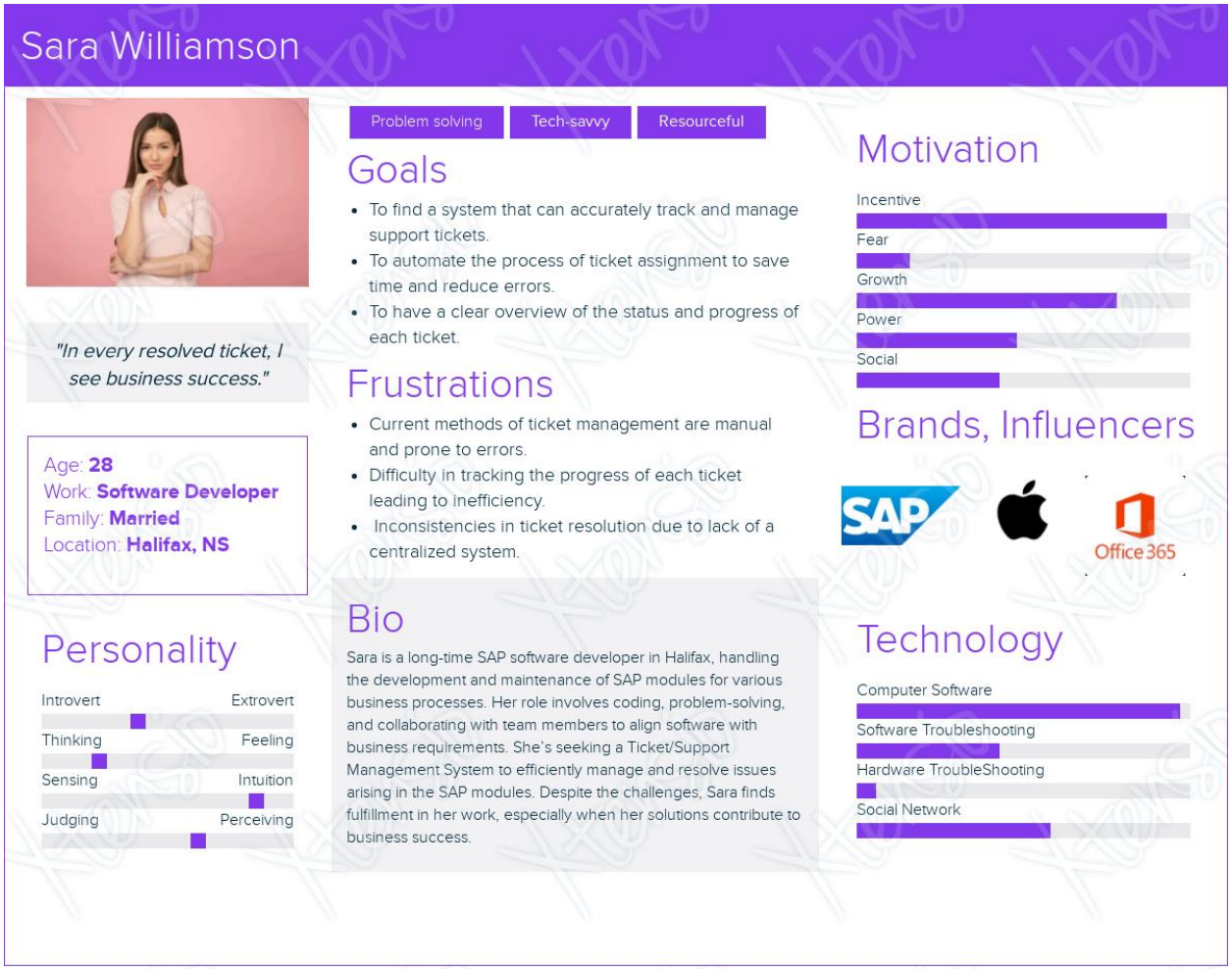


Figure 1: User persona of Team Lead prepared with Figma [3].

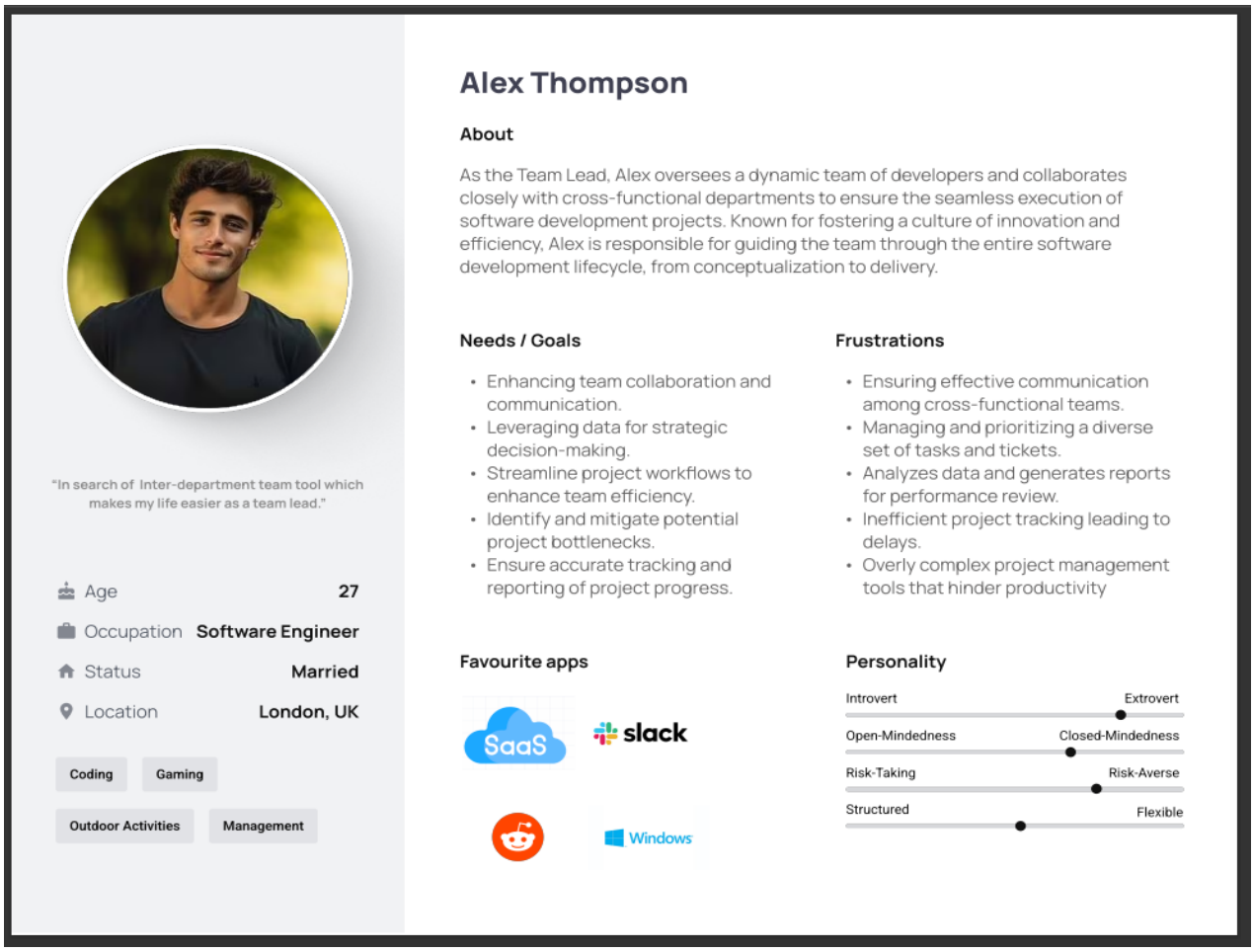


Figure 2: User persona of Support team member prepared with Figma [3].

Scenario 1:

Sara, a SAP Software Developer, is working on a complex SAP module and she comes across an issue that she can't resolve. She went through the knowledge base articles but she was not able to find any article related to the issue, so she decided to set up her profile and raise a ticket in the Ticket/Support Management System, providing all the necessary details about the issue. The system categorizes the ticket to SAP Module category

Persona: SAP Software Developer
Feature: Knowledge Base, Raise a Ticket, Profile Management

Scenario 2:

Alex, as a Team Lead, faces the challenge of efficiently managing team collaboration and communication across various departments. One day, he notices a surge in support tickets related to a critical project. To address this, he needs to quickly assess the status of the tickets, assign them to the relevant teams, and escalate high-priority issues to ensure timely resolution. Additionally, he aims to gather data on ticket distribution for strategic

decision-making. An ideal scenario would involve a tool that enables Alex to streamline these processes seamlessly, enhancing team efficiency and facilitating informed decision-making in a fast-paced software development environment.

Persona: Support Department Team Lead
Feature: Team Management, Ticket Assignment, Reporting and Analytics and Escalation

Scenario 3:

Sara, an introverted yet tech-savvy software developer, encounters a scenario where she needs to collaborate with cross-functional teams on resolving complex SAP module issues. Her frustration lies in the inefficiency of current manual ticket management methods which do not let her stay in the loop by not sending notifications, leading to errors and inconsistencies in ticket resolution. In this situation, a streamlined internal support ticket management system that fosters efficient collaboration and provides a centralized platform for issue resolution becomes essential for Sara. Such a system would alleviate her frustrations, allowing her to focus on coding and problem-solving, ultimately contributing to business success and her personal growth in the software development field. Sara always prefers to use secure systems which are only allowed to authenticated users.

Persona: Tech-savvy Software Developer
Feature: Ticket Management, Manage the Board, Notification, Authentication

Scenario 4:

As an IT Support Team Lead, Alex faces a scenario where a critical software development project is reaching its deadline. He realizes that effective communication among cross-functional teams is crucial to avoid bottlenecks and ensure timely delivery. Alex needs a tool that allows him to manage the team boards seamlessly, add or remove teams as needed, and ensure clear communication channels.

Persona: IT Support Team Lead
Feature: Comments on Ticket, Notification, Escalation

3.2 User-Centered Design Approach

3.2.1 Information Architecture

The website is divided into five distinct sections, each corresponding to a feature of the application (see section 3). Each box represents a page which enables the user to navigate between the features in a maximum of four clicks. Following this four-click rule makes the UI (User Interface) efficient and reduces the cognitive load on the users (see Figure 3 sitemap).

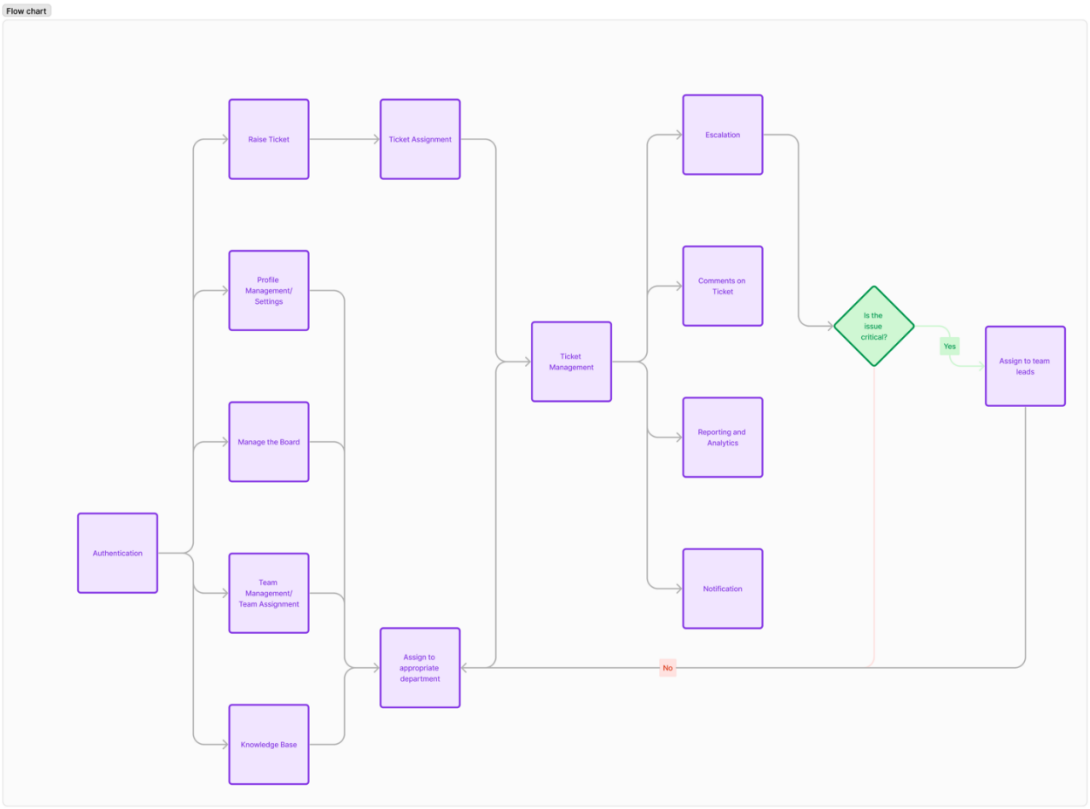


Figure 3: Sitemap of Application prepared with Figma [3]

3.2.2 Design and Layout

3.2.2.1 Wireframes

Feature: Employer/Employee Account Creation and Authentication

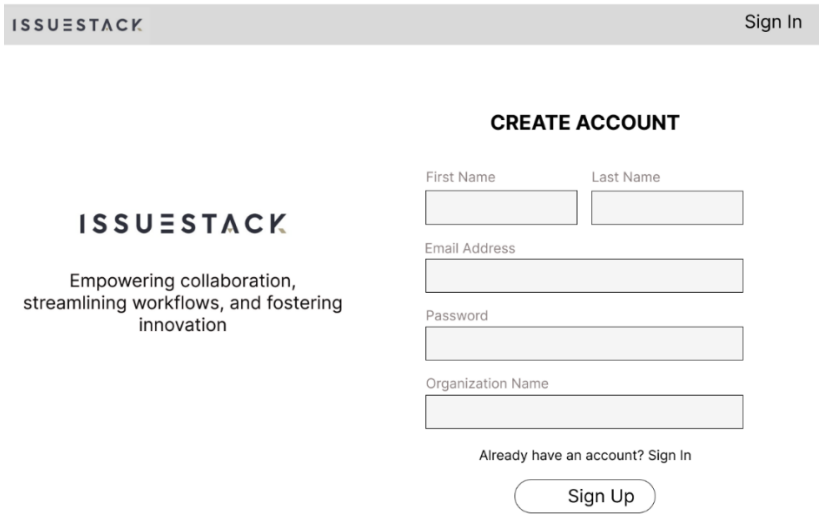


Figure 4: Prototype of registration by Bhautik Rameshbhai Koshiya [4]

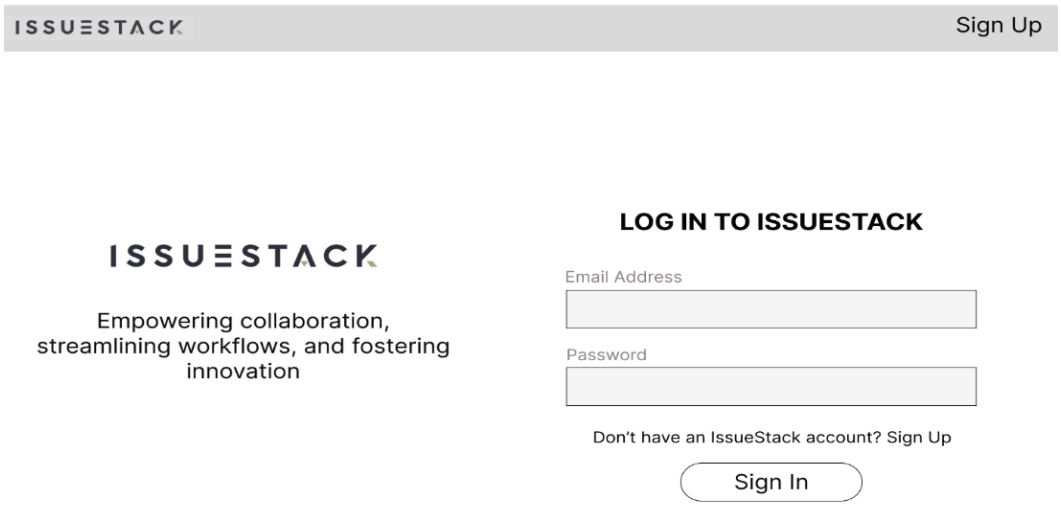


Figure 5: Prototype of sign in by Bhautik Rameshbhai Koshiya [4]

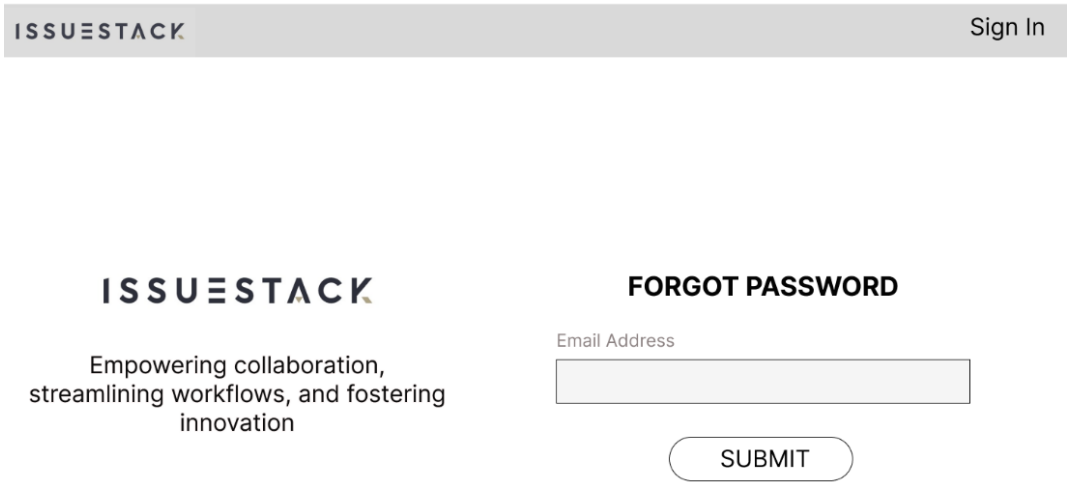


Figure 6: Prototype of forget password by Bhautik Rameshbhai Koshiya [4]

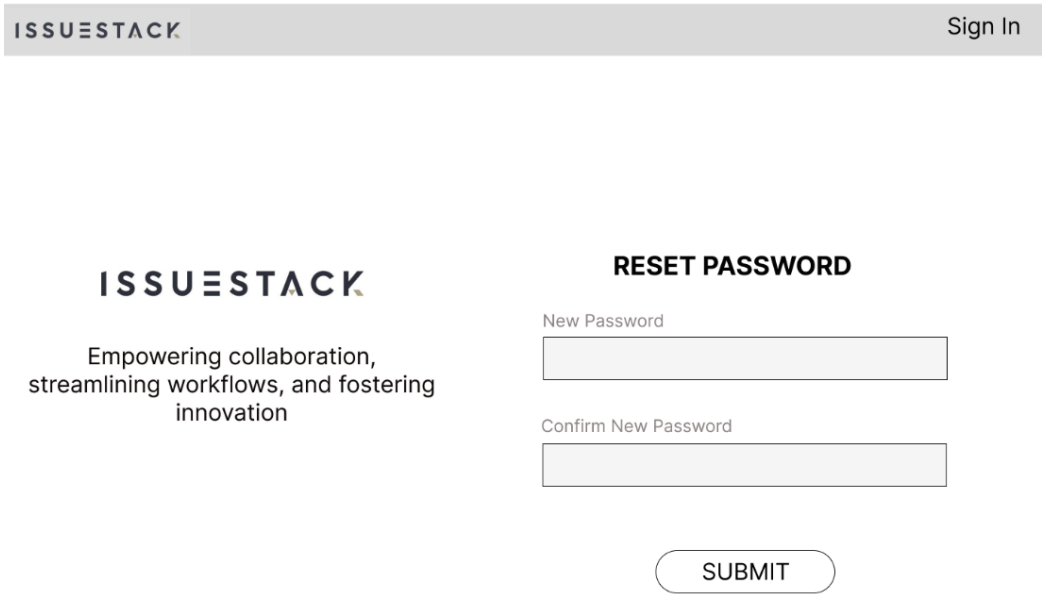


Figure 7: Prototype of reset password by Bhautik Rameshbhai Koshiya [4]

Feature: Profile Management/Settings

Figure 8: Prototype of Profile Settings by Dhruvik Maheshbhai Kakadiya [4]

Feature: In-app Notification

Figure 9: Prototype of in-app notification by Bhautik Rameshbhai Koshiya [4]

Feature: Raise Ticket

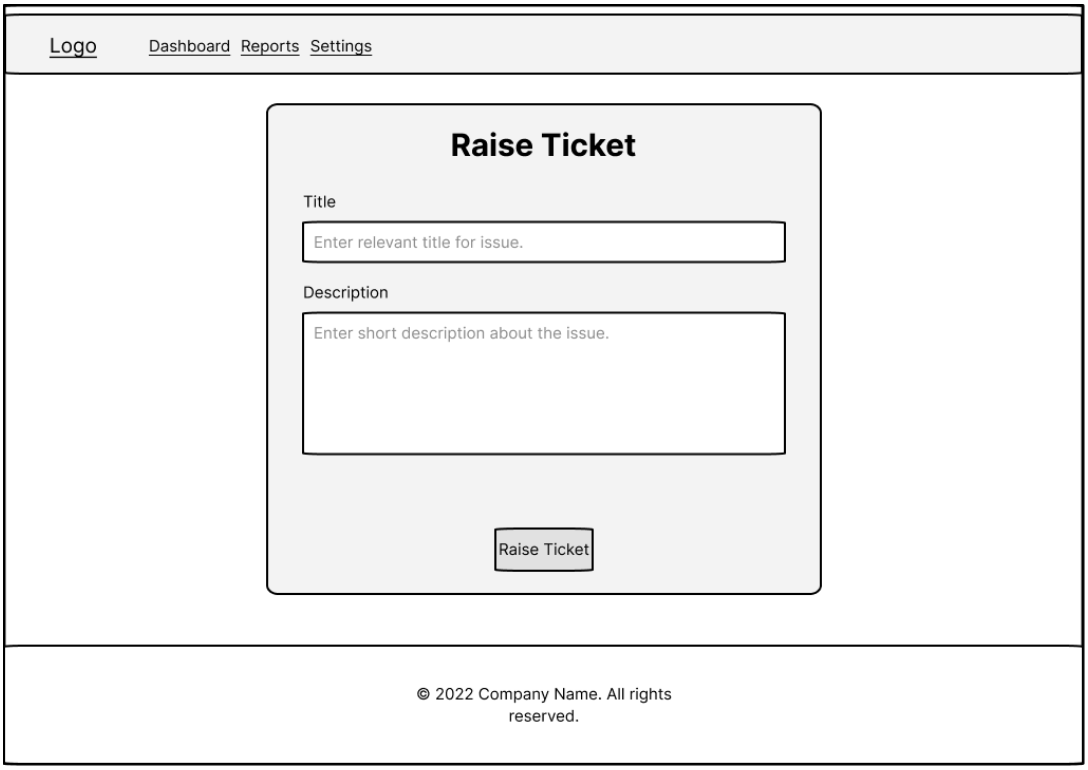


Figure 10: Prototype of raise a ticket by Nisarg Sharadkumar Vaghela [4]

Feature: Escalate Ticket

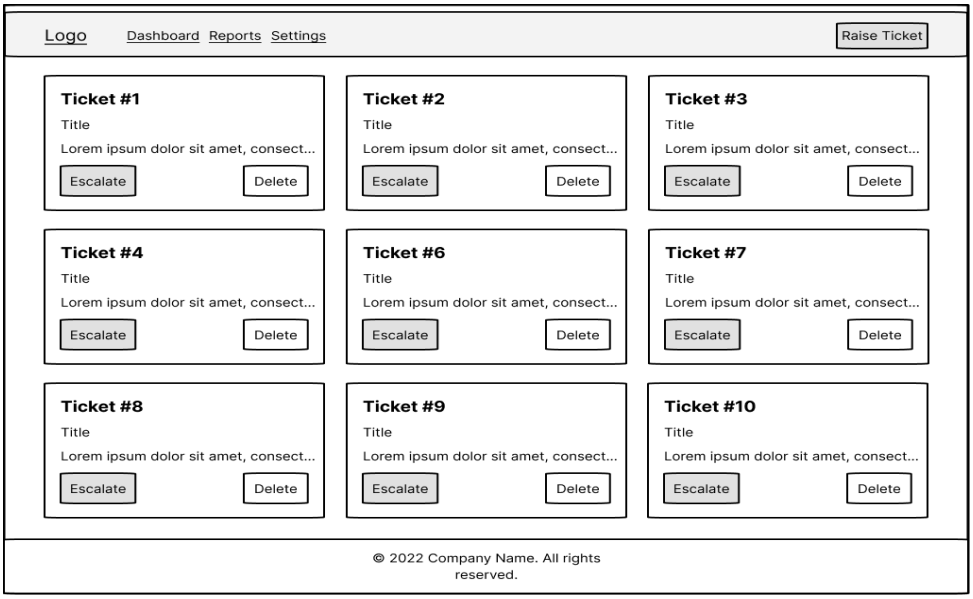


Figure 11: Prototype of escalate ticket by Nisarg Sharadkumar Vaghela [4]

Feature: Ticket Management

Figure 12: Prototype of ticket details page by Darshit Mukeshbhai Dhameliya [4]

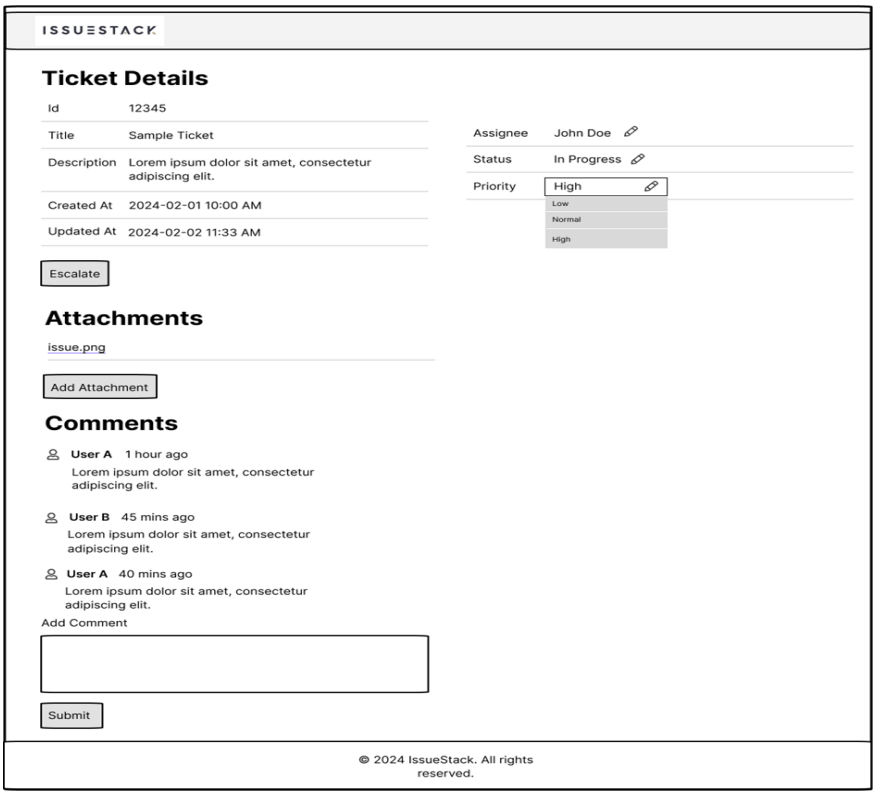


Figure 13: Prototype of update/change Priority by Darshit Mukeshbhai Dhameliya [3]

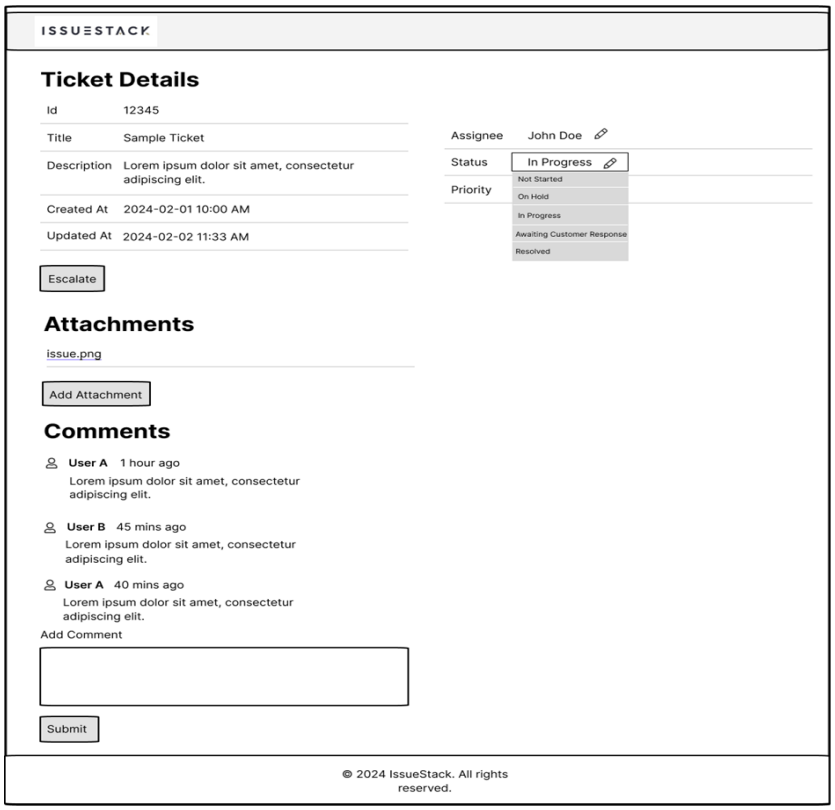


Figure 14: Prototype of update/change status by Darshit Mukeshbhai Dhameliya [3]

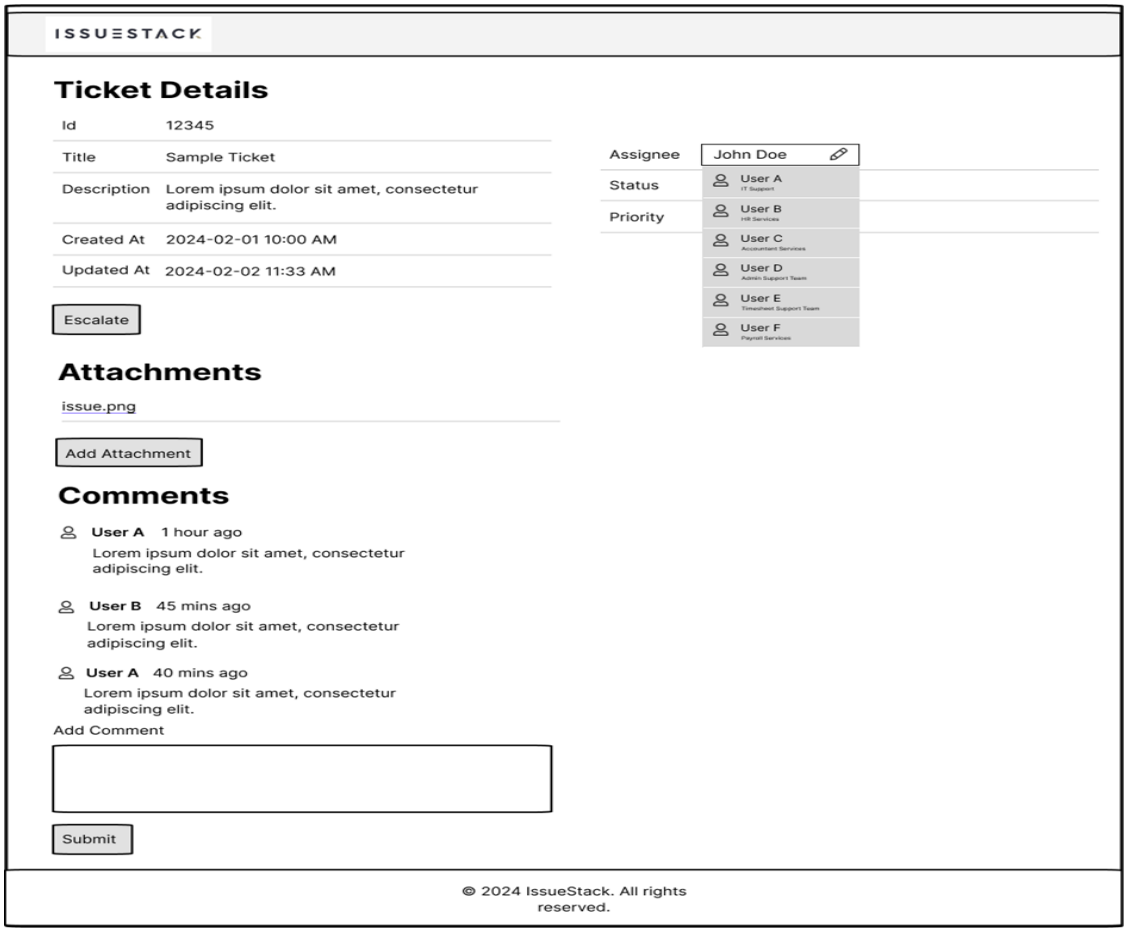


Figure 15: Prototype of forward to another team by Darshit Mukeshbhai Dhameliya [3]

Feature: Manage the team

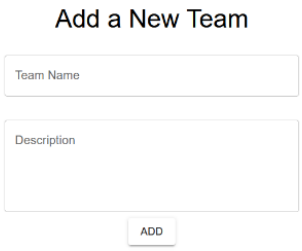


Figure 16: Prototype of adding a new team Kuldeep Rajeshbhai Gajera [3]

Feature: Ticket Assignment

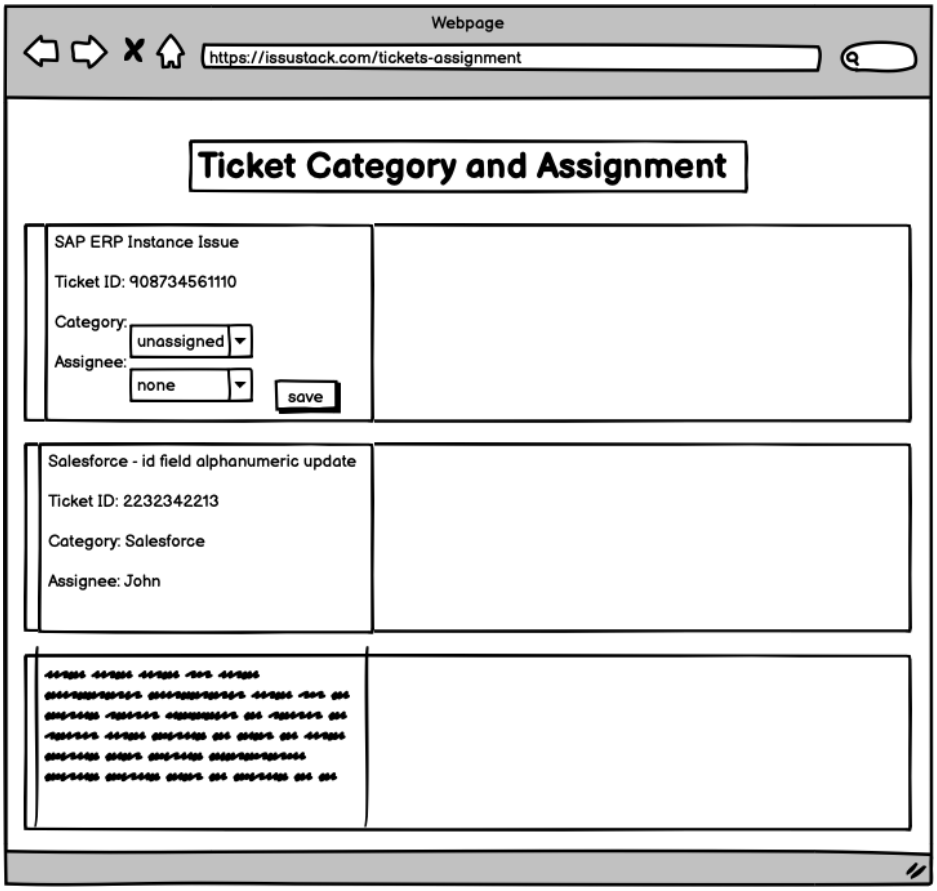


Figure 17: Prototype of Ticket Assignment by Gowri Prashanth Kanagaraj [6]

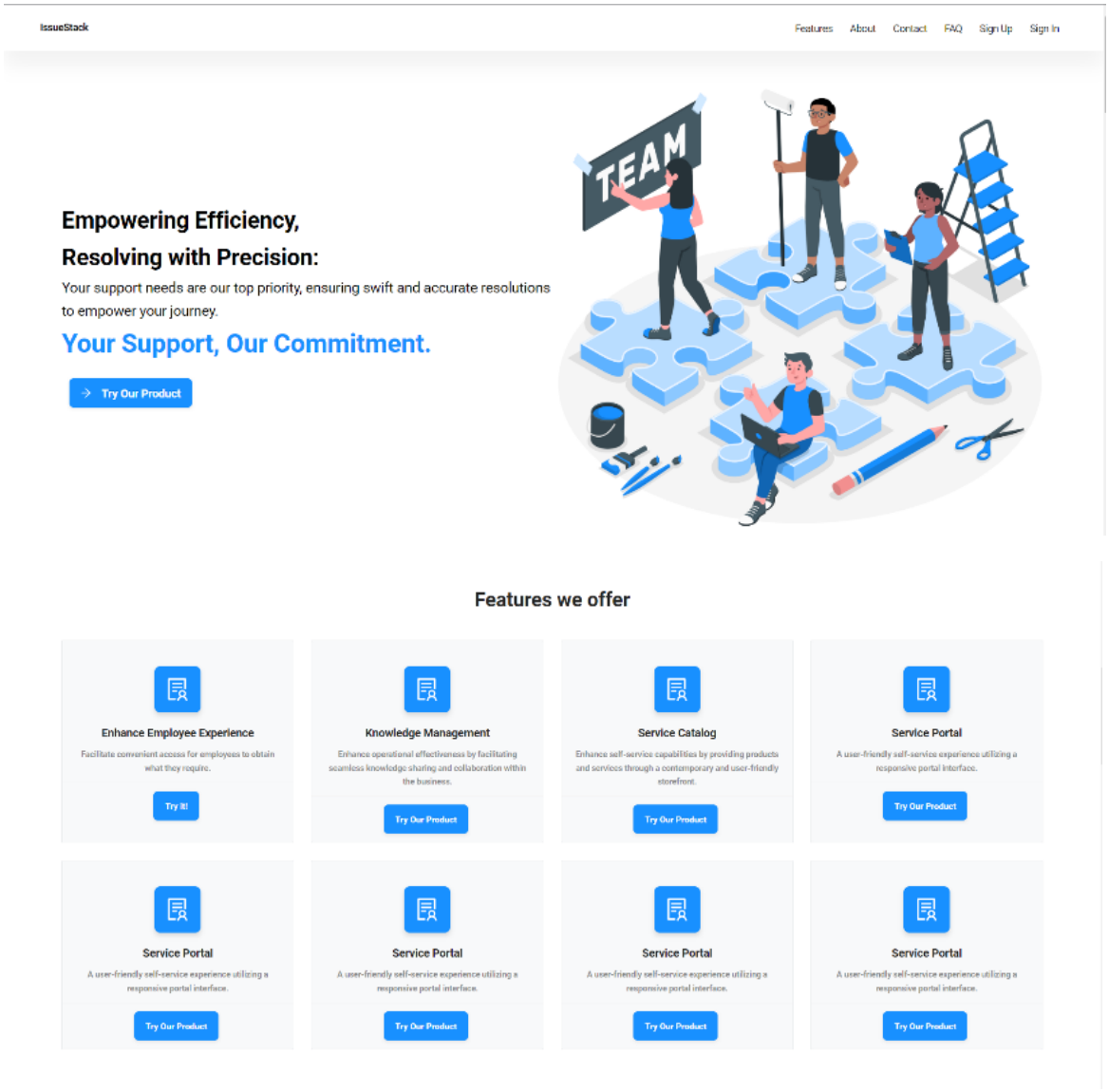
3.3 Design

We focused on creating a user-friendly interface, keeping it simple and easy to navigate for users with different levels of technical knowledge. Our design adjusts smoothly to different devices and screen sizes, ensuring everyone can access our website comfortably. By using modular architecture, we made sure our website can grow and change easily as needed. We included plenty of buttons to reduce user errors and make it effortless for users to get the most out of our website.

Below are the design decisions for different pages:

1) Our homepage is designed to provide users with all the essential information they need without having to navigate to other pages. We've made it a priority to ensure easy navigation, so users can quickly find what they're looking for. To achieve this, we've created clear and intuitive menus, along with prominent calls-to-action, to guide users seamlessly through the website. The design of our homepage focuses on delivering a visually appealing and user-friendly experience. We've ensured that it's easy to navigate and responsive across different devices. Our goal is to effectively communicate the purpose and offerings of our website to users. At the end of the page, we've included a

"Contact Us" section. This makes it easy for users to reach out to us if they need any assistance or have any questions.



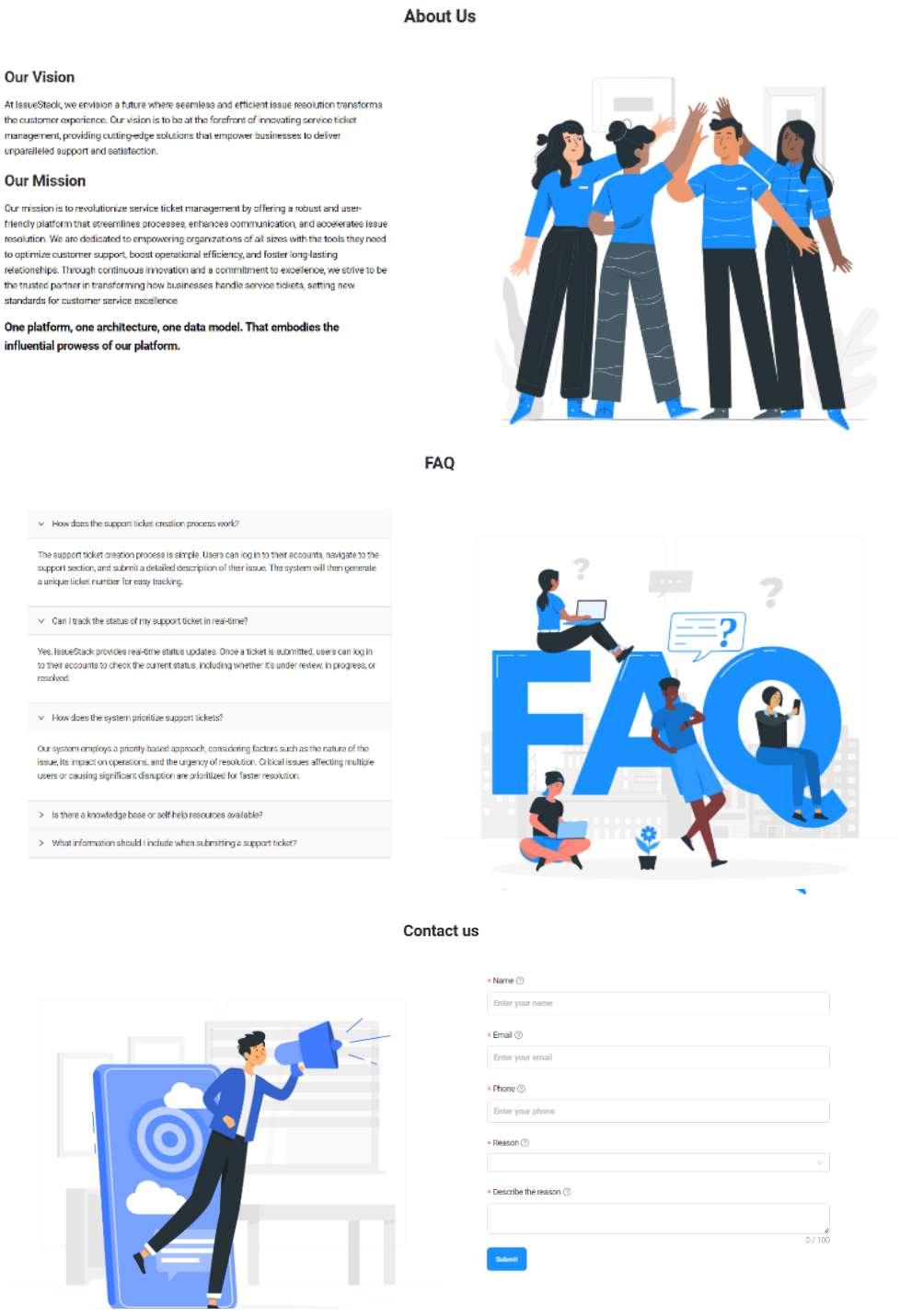


Figure 18: Landing Page [11]

2) The signup and sign-in pages were designed with a focus on simplicity, security, and user experience. Firstly, we ensured that the design elements and layout were intuitive and easy to understand, minimizing the cognitive load for users during the signup and sign-in processes. We used Clear and concise form fields, error messaging, and guidance to facilitate a smooth signup and sign-in experience for users.

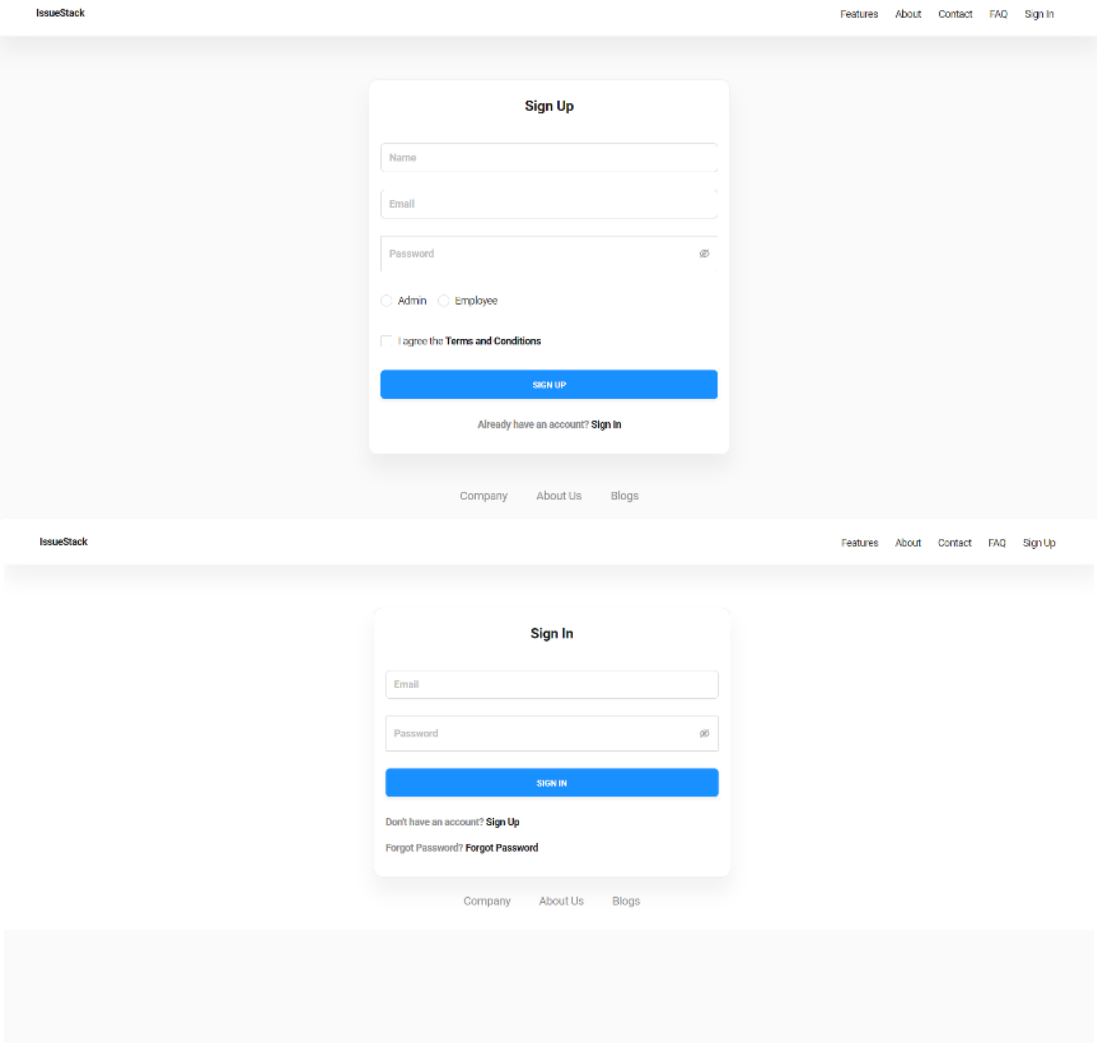


Figure 19: Signup and Signin Page [11]

3) The Update Profile page was carefully crafted with a strong emphasis on user convenience, data security, and customization options. Our design prioritizes a clean and intuitive interface, allowing users to navigate the update profile process effortlessly. We've incorporated clear labels, logically grouped fields, and intuitive controls to streamline user interaction and minimize any confusion. Additionally, we provide user feedback through alert toasts, informing users when their profile information has been successfully updated or if an error has occurred. This ensures a smooth and user-friendly experience while updating profiles.

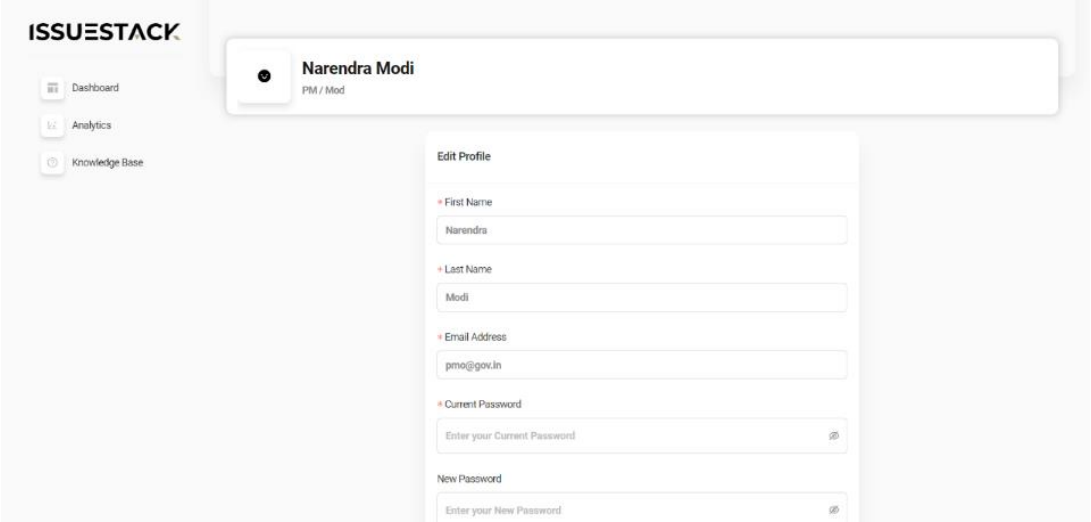


Figure 20: Update Profile Page [11]

4) The Knowledge Base page has been meticulously designed to offer users easy access to comprehensive information and resources, all while ensuring a seamless browsing experience. Our design focuses on efficiently organizing information to facilitate easy navigation and quick access to relevant resources. To achieve this, we've categorized information logically into topics and subtopics, allowing users to browse and locate guides efficiently. Additionally, we've implemented collapsible cards to display lists of questions and their respective answers, providing users with a structured and intuitive way to access the information they need.

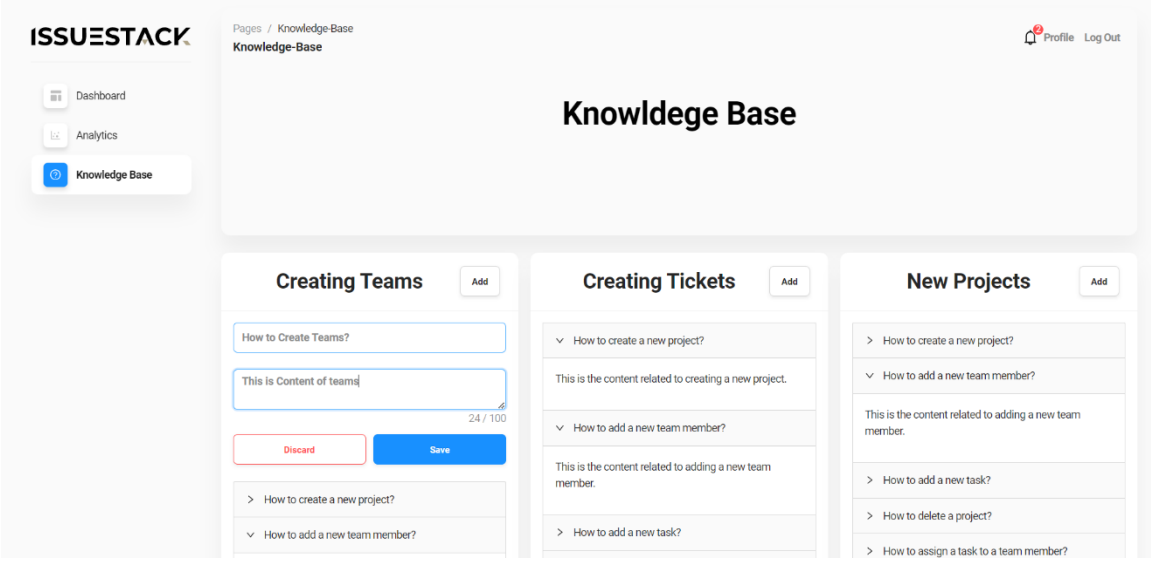


Figure 21: Knowledge base Page [11]

5) The Analytics page has been meticulously designed to offer users comprehensive insights and data visualization tools, enabling informed decision-making and performance analysis. Our design focuses on effective data visualization techniques to present complex analytics data in a visually appealing and easy-to-understand format. To achieve this, we've incorporated various charts, graphs, and interactive visualizations, such as line charts and bar graphs. These tools provide users with meaningful insights and trends at a glance, allowing them to quickly grasp important information and make informed decisions based on the analytics data presented.

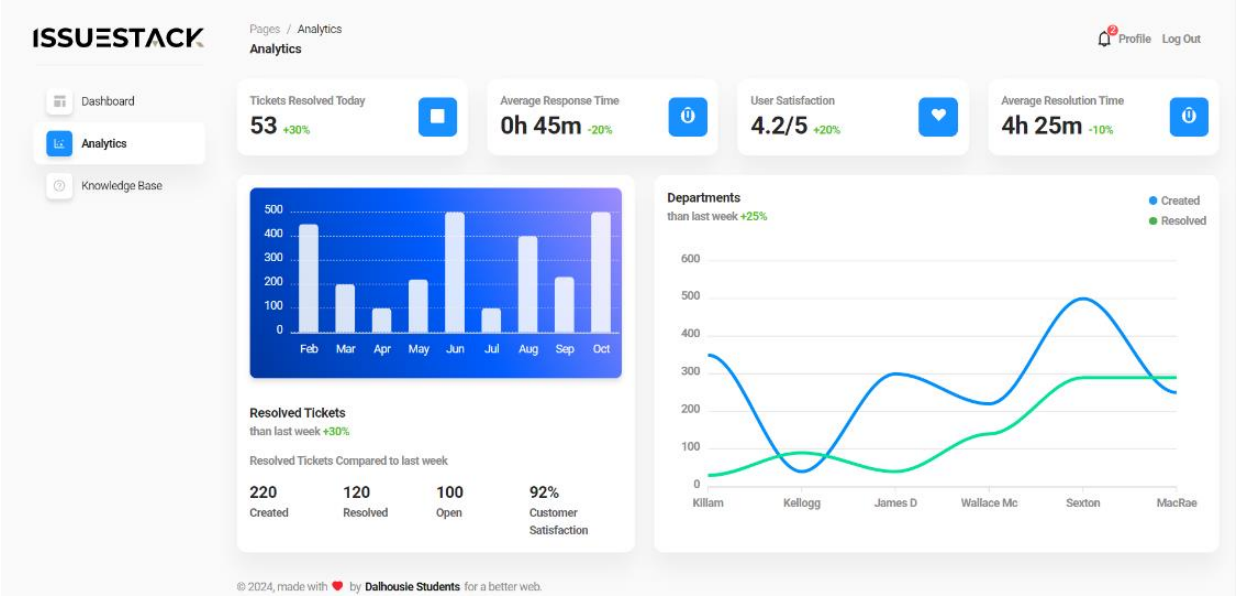


Figure 22: Analytics Page [11]

6) The Issue Board page was meticulously crafted with the primary objective of offering users a centralized platform to efficiently manage and track issues, tasks, and projects. Our design focuses on providing robust task management capabilities, enabling users to seamlessly create, assign, prioritize, and track tasks. To achieve this, we've implemented features such as task cards, labels, due dates, and status indicators to facilitate effective task management and organization. We've also incorporated principles of the Kanban methodology to visualize workflow and streamline task progression. The design adopts a Kanban board layout with customizable columns representing different stages of work, such as To Do, In Progress, and Done. This layout provides users with a clear overview of task status and progress, making it easy to track and manage tasks efficiently.

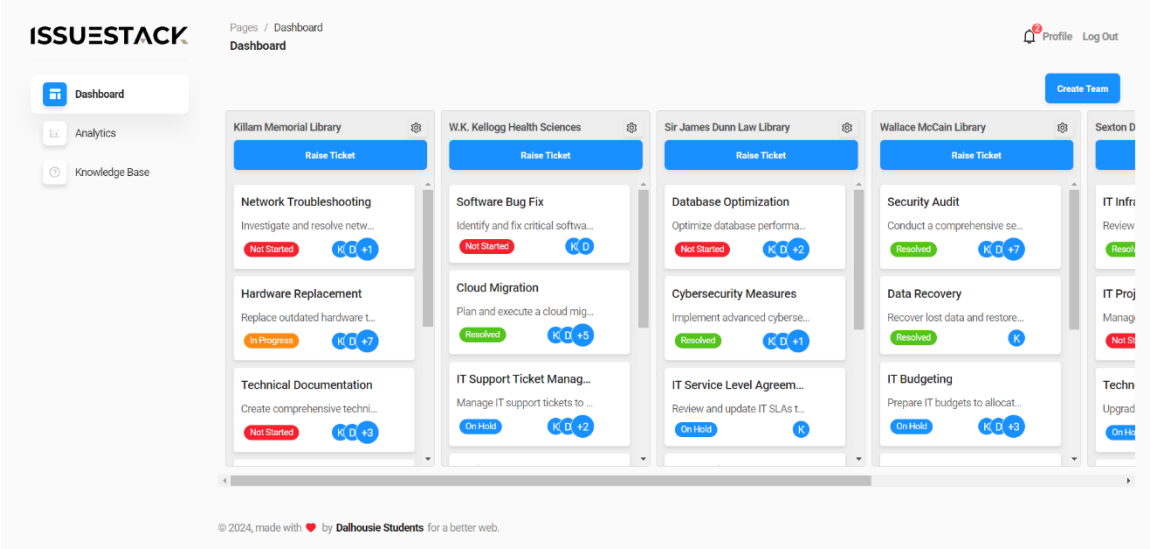


Figure 23: Issue Board [11]

4. APPLICATION WORKFLOW

4.1 Interaction Design

4.1.1 Feature: Authentication

Tasks:

- Sign up
- Sign in
- Forget Password

Use Case – Sign Up

1. The user navigates to the system's authentication page.
2. The user selects the "Create Employer Account" option.
3. The user fills out the required information, including name, email, password, and company details.
4. Information will be validated against validation rules, after successful validation
5. The user submits the registration form.
6. The system verifies the information, creates the employer account, and sends a confirmation email.
    6.1. The user receives the confirmation email and clicks on the verification link.
7. If the email verification link is not received or expires, the user requests a new verification email.
8. The user receives a new email, clicks the verification link, and successfully verifies the account.

9. After Successful verification, the user will get message stating Account created successfully.

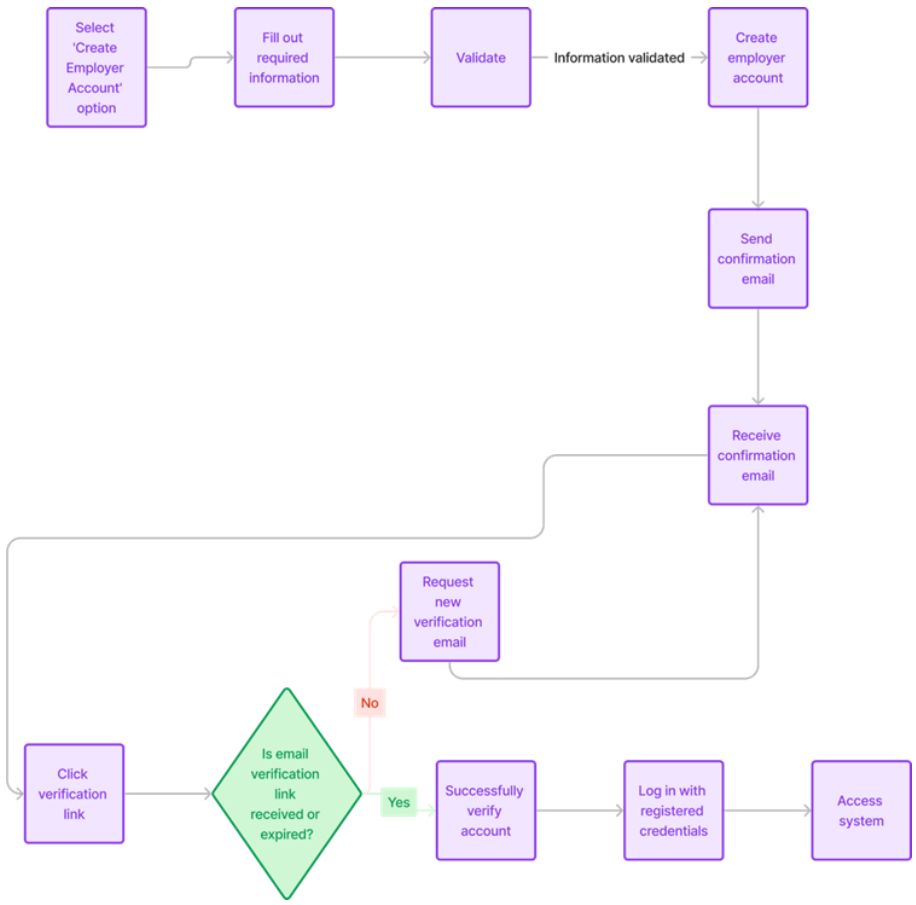


Figure 24: Task flow diagram of sign up [3].

Use Case – Sign in

1. The user navigates to the platform's login page.
2. The user enters their registered email address and password into the respective fields.
3. The user submits the login form.
4. The system verifies the entered credentials against the stored information.
    4.1. If the credentials are correct, the system grants access to the employer dashboard.
    4.2. If the credentials are incorrect, the user receives an error message prompting them to verify the information entered and try again.
5. Upon successful authentication, the user gains access to the employer dashboard where they can manage teams, access reporting tools, and utilize analytical features for performance evaluation.
6. The user remains logged in until they choose to log out or their session expires due to inactivity or security protocols.

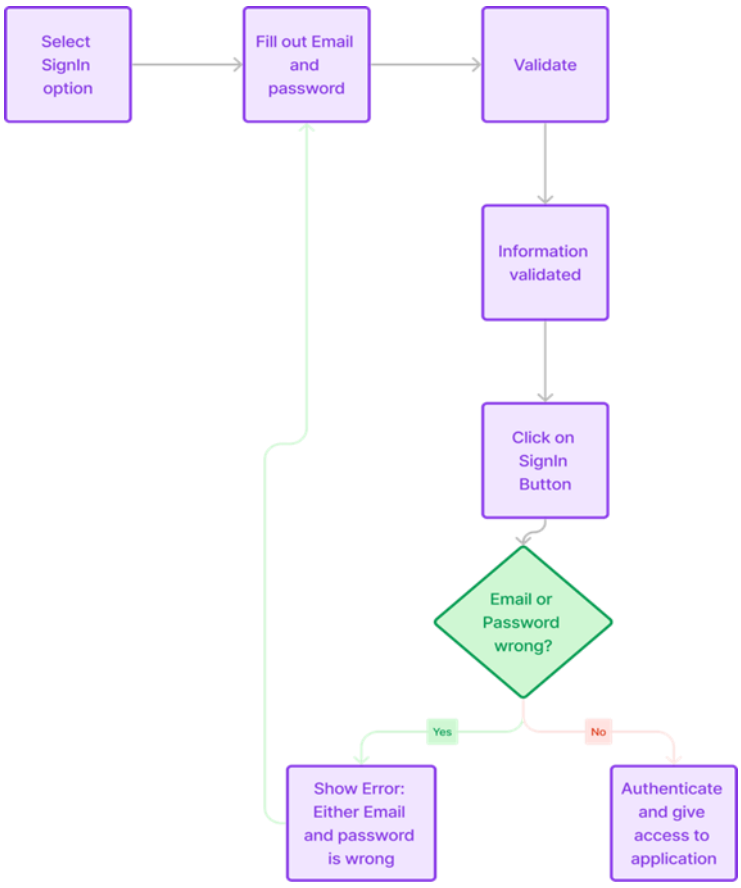


Figure 25: Task flow diagram of sign in [3].

Use Case – Forget Password

1. The user navigates to the "Forgot Password" page provided by the system.
2. The user enters their registered email address into the designated field.
3. The user submits the form for password recovery.
4. The system proceeds to verify the existence of the provided email address within its database.
    4.1. If the email address exists within the system, The user receives an email containing instructions for resetting the password.
    4.2. If the email address does not match any records in the system, The user is promptly notified with an error message, advising them to verify the entered email address and retry the password recovery process.
5. Upon receiving the password reset email, The user clicks on the provided reset password link, directing them to the password reset page.
    5.1. The user then sets a new password adhering to the system's password policy and validation rules.
    5.2. If the newly entered password meets the validation criteria, The user is notified via email that the password reset was successful.

5.3. In case the validation process fails, The user receives an error message instructing them to enter a password that complies with the system's requirements.

6. With the password successfully reset, The user gains the ability to log in using the new credentials.

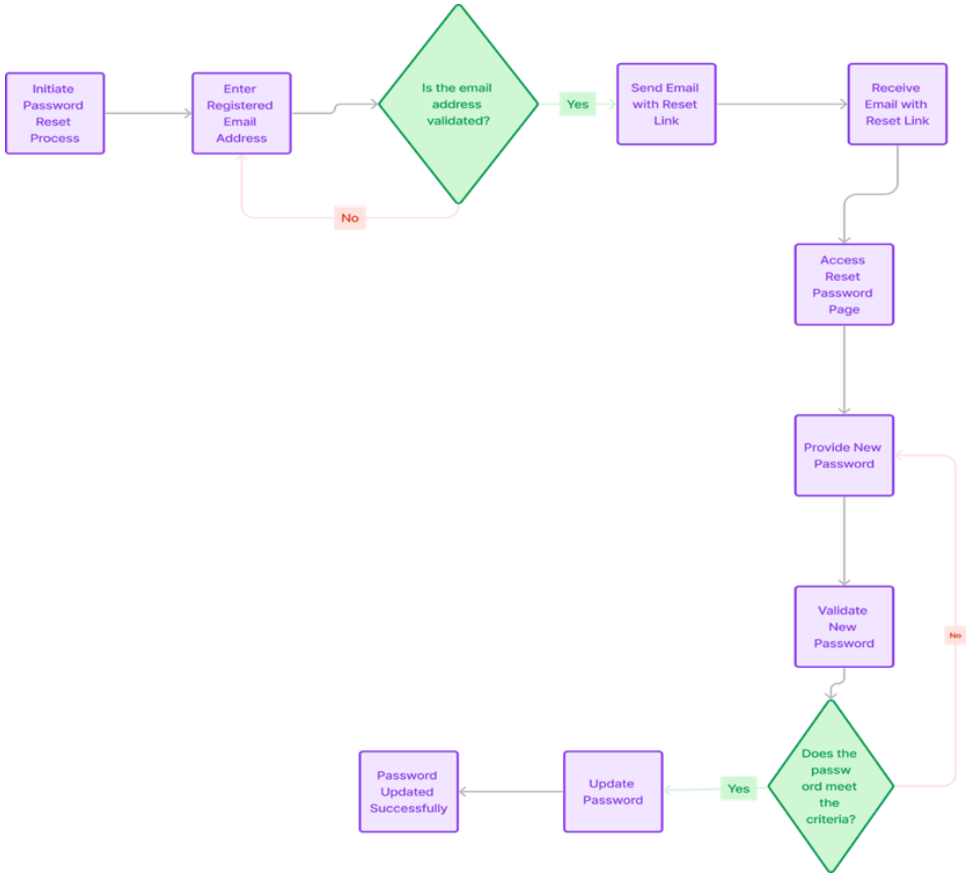


Figure 26: Task flow diagram of forget password [3].

4.1.2 Feature: Notification (email notification, in app notification)

Tasks:

- Email Notification
- In-App Notification

Use Case – Email Notification

1. The user needs to register and log in to the Issue Stack website to access its features.
2. Users create new tickets to report issues or tasks.

3. When a ticket is assigned to a team member, Issue Stack triggers an email notification to inform the assigned member about the new task.
4. The assigned team member receives an email notification containing details of the newly assigned ticket.
5. When the status of a ticket changes (e.g., from open to in progress, from in progress to resolved), Issue Stack sends an email notification to the relevant users involved with the ticket.
6. Users associated with the ticket receive an email notification reflecting the status change, ensuring they are updated on the ticket's progress.
7. If there are any updates or comments added to a ticket, users associated with the ticket receive email notifications to keep them informed.
8. The email notifications are designed to be informative, including details such as ticket title, ticket ID, description, status changes, relevant comments, or updates.

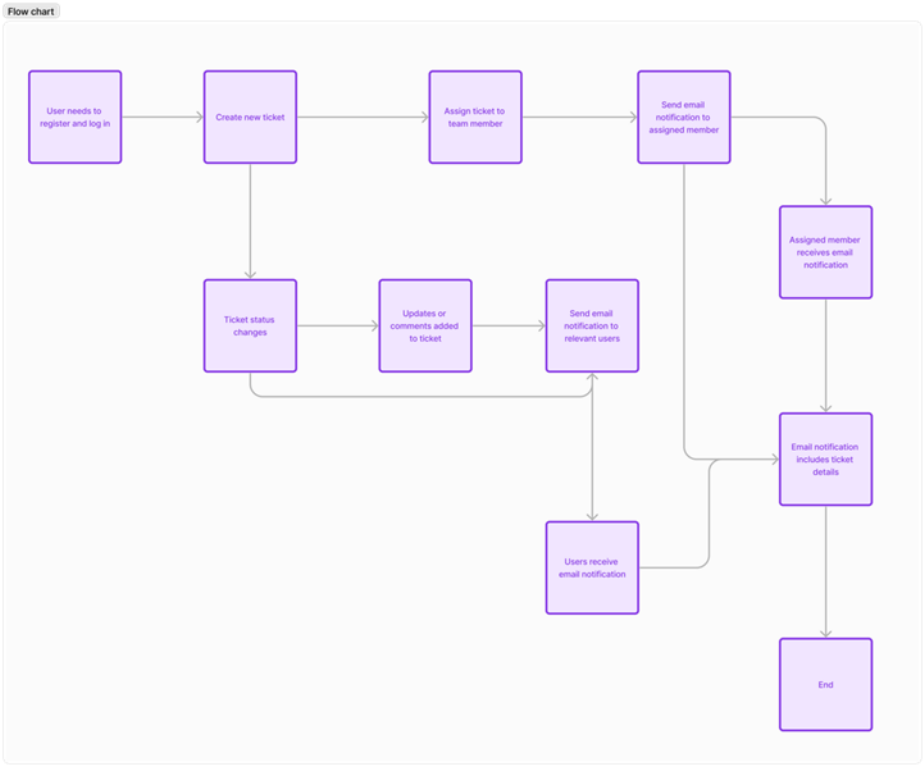


Figure 27: Task flow diagram of email notification [3].

Use Case – In-App Notification

1. A user needs to register and log in to the Issue Stack website to access its features.
2. Users create new tickets to report issues or tasks.
3. When a ticket is assigned to a team member, Issue Stack triggers an in-app notification to inform the assigned member about the new task.
4. The assigned team member receives an in-app notification containing details of the newly assigned ticket.

5. When the status of a ticket changes (e.g., from open to in progress, from in progress to resolved), Issue Stack sends an in-app notification to the relevant users involved with the ticket.
6. Users associated with the ticket receive in-app notification reflecting the status change, ensuring they are updated on the ticket's progress.
7. If there are any updates or comments added to a ticket, users associated with the ticket receive in-app notifications to keep them informed.
8. The in-app notifications are designed to be informative, including details such as ticket title, ticket ID, description, status changes, relevant comments, or updates.

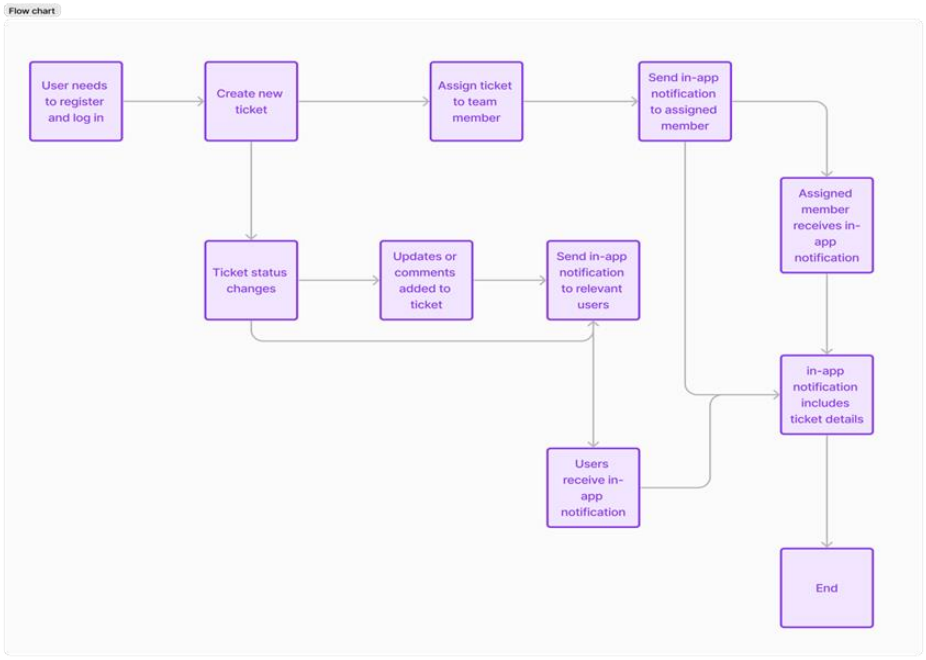


Figure 28: Task flow diagram of in-app notification [3].

4.1.3 Feature: Ticket Assignment – automatic ticket assignment to the right support agent

Tasks:

- Ticket Categorization
- Agent Matching

Use Case – Ticket Categorization

1. The user raises a ticket providing all the necessary details about the issue, the module it's related to, the steps the user took before encountering the issue, and any error messages received.

2. After filling in all the details, the user submits the ticket. The ticket is now in the system and ready to be processed.
3. The System starts to analyze the details provided in the ticket.
    3.1. If the system fails to find a suitable category for the ticket, it places the ticket in a default category or leaves it uncategorized.
4. This allows for manual categorization by a team member who can review the ticket details and assign an appropriate category.
5. The system identifies that the ticket falls under a particular category, based on the details provided by the user.
6. This category is now attached to the ticket and can be used in further process.

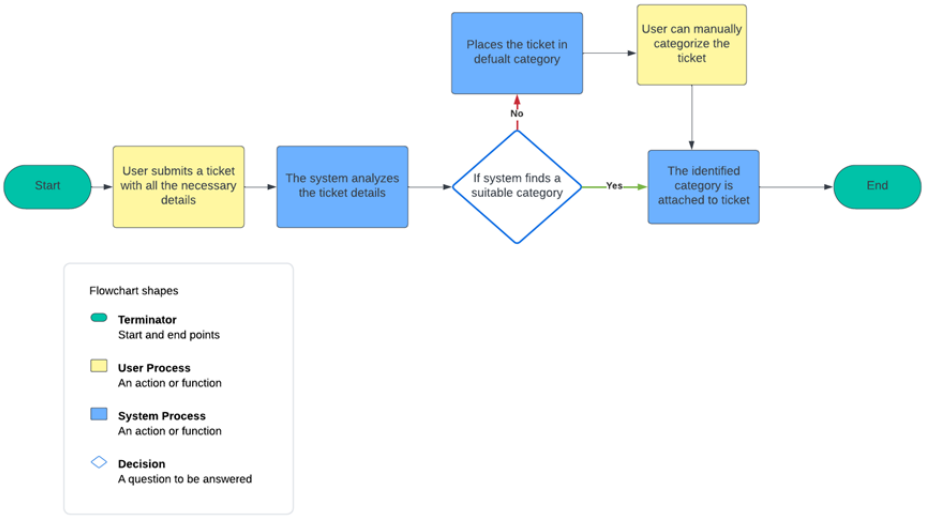


Figure 291: Task flow diagram of ticket categorization [5]

Use Case – Agent Matching

1. Once the ticket is categorized, the system starts to analyze the category and other details of the ticket.
2. The system checks the availability and expertise of all support agents.
    2.1. If no agent is found, the system places the ticket in a queue until a suitable agent becomes available.
    2.2. This means that the ticket will wait till the right expert is available
3. The system identifies a Support Agent as the most suitable agent to handle the ticket based on their expertise in handling similar issues and current availability.
4. The system assigns the ticket to the agent.
5. This assignment is now attached to the ticket and can be used in further process

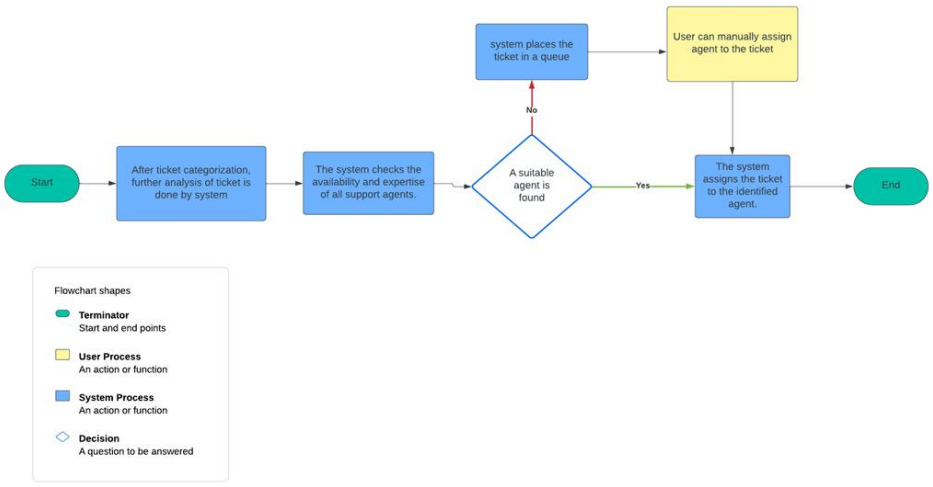


Figure 302: Task flow diagram of agent matching [5]

4.1.4 Feature: Raise Ticket

Tasks:

- Raise Ticket form

Use Case – Raise Ticket form

1. The user navigates to the project dashboard on the Project Management Ticket Service Board.
2. The user finds the team responsible for addressing the bug.
3. The user selects the option to "Raise Ticket" under that team.
4. The user fills in essential details about the bug:
5. The user provides a descriptive title for the bug.
6. The user is altered the title should not be empty.
    6.1. The user enters the proper title.
    6.2. The user provides a detailed explanation of the issue encountered.
7. The user is altered the description should not be empty.
    7.1. The user enters the proper description.
    7.2. The user reviews the information entered for accuracy.
8. The user clicks on the "Submit" button to raise the ticket.
9. The system acknowledges the successful creation of the ticket with a confirmation message.
10. The ticket is assigned a unique identification number for tracking purposes.

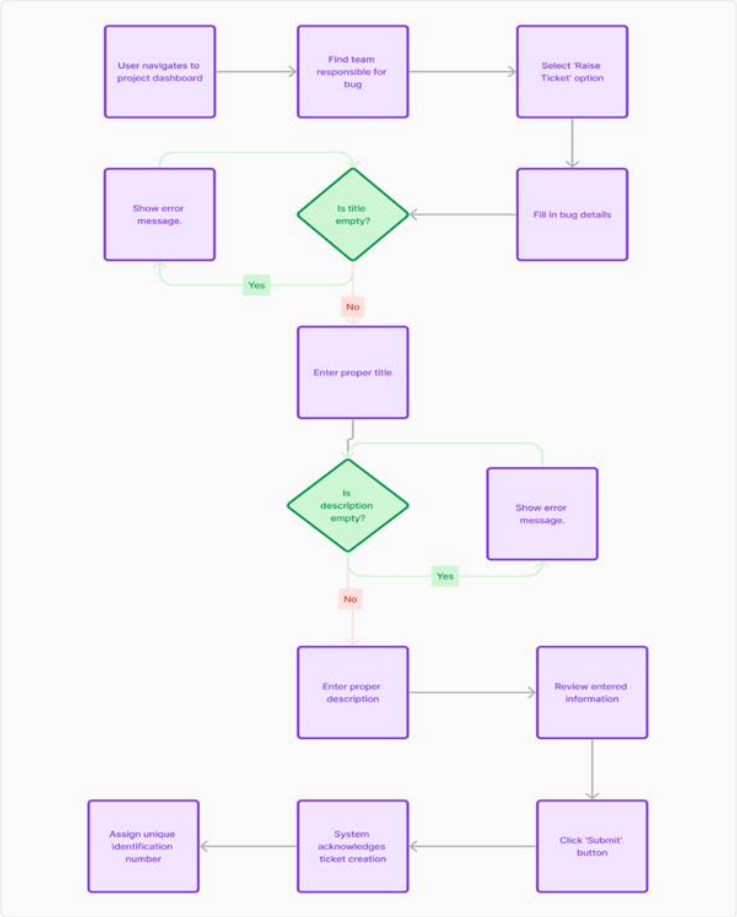


Figure 313: Task flow diagram of raise a ticket by [3]

4.1.5 Feature: Escalate Ticket

Tasks:

- Escalate ticket

Use Case – Escalate Ticket

1. The user navigates to the project dashboard on the Project Management Ticket Service Board.
2. The user locates the ticket related to the critical bug that requires escalation.
    2.1.If the ticket is not found, then the user recreates it.
3. The user evaluates the severity and impact of the bug to determine if escalation is warranted. Criteria for escalation may include the potential for project delays, significant functionality issues, or high client impact.
4. If the user determines that escalation is necessary, they select the "Escalate" option associated with the identified bug ticket.

5. The system acknowledges the successful escalation of the ticket with a confirmation message. The ticket status may be updated to reflect the escalated state.
6. The system automatically notifies the relevant teams or individuals responsible for handling escalated tickets. This may include senior developers, project managers, or specialized support teams.

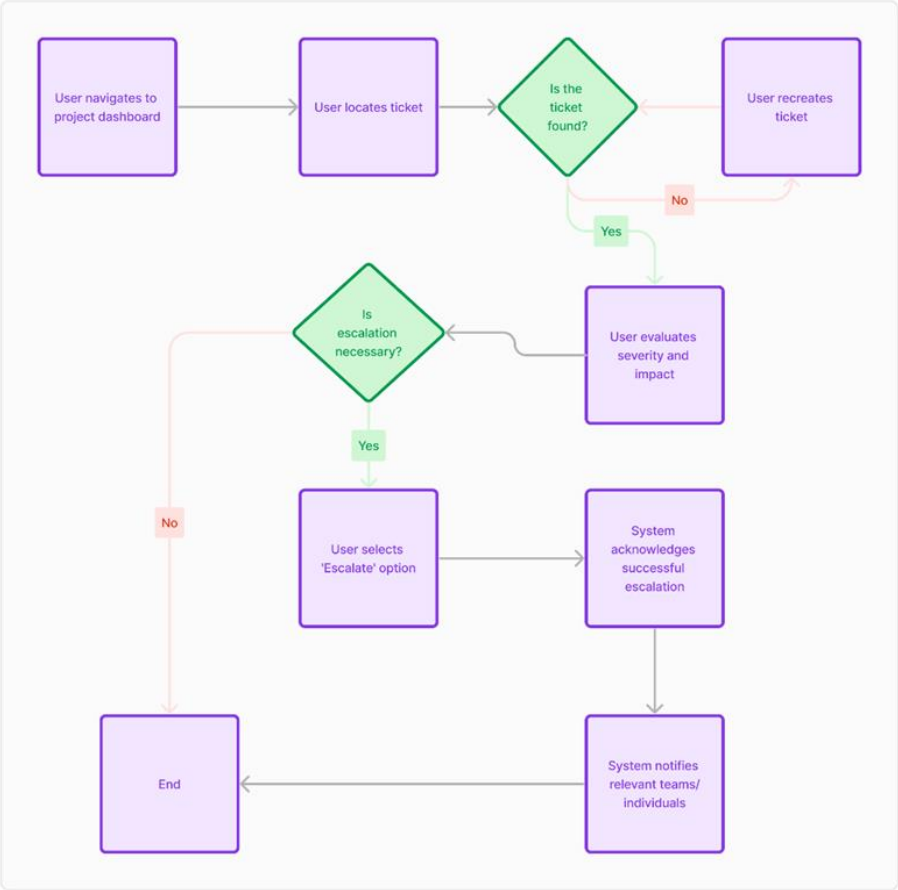


Figure 324: Task flow diagram of escalate ticket [3]

4.1.6 Feature: Ticket Management

Tasks

- Set Priority and Update Ticket Status
- Forward or Involve Other Support Team

Use Case – Priority and Update Ticket Status**:**

1. The user logs into the Service Ticket Management System.

2. The user navigates to the ticket dashboard to view the list of assigned tickets.
3. The user selects the ticket with the urgent issue.
4. The user updates the priority of the ticket to 'High' based on the severity of the reported problem.
    4.1. In the event of a system error preventing the priority update, the user receives an error message.
    4.2. The user troubleshoots and verifies the internet connection and system status.
    4.3. If the issue persists, the user contacts the IT support team via email or phone to resolve the technical problem.
    4.4. Once the technical issue is resolved, the user repeats the process of updating the priority of the ticket.
5. The user changes the status of the ticket to 'In Progress' to indicate active resolution.
    5.1. In the event of a system error preventing the status update, the user receives an error message.
    5.2. The user troubleshoots and verifies the internet connection and system status.
    5.3. If the issue persists, the user contacts the IT support team via email or phone to resolve the technical problem.
6. Once the technical issue is resolved, the user repeats the process of updating the status of the ticket.
7. The user saves the changes.

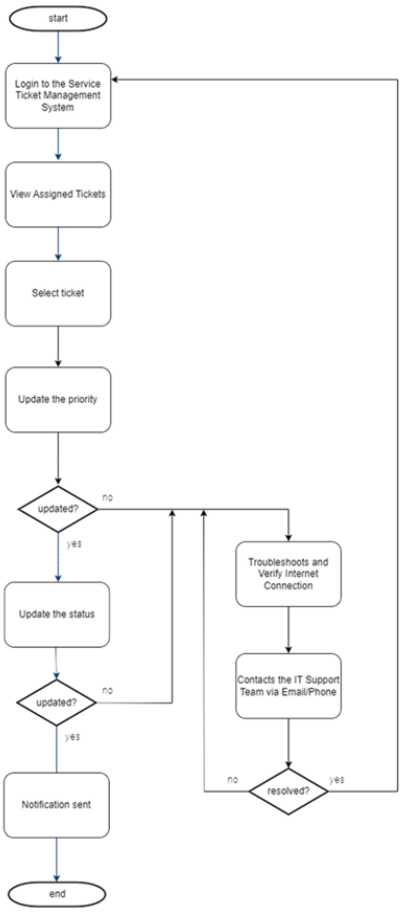


Figure 33: Task flow of Set Priority and Update Ticket Status [3]

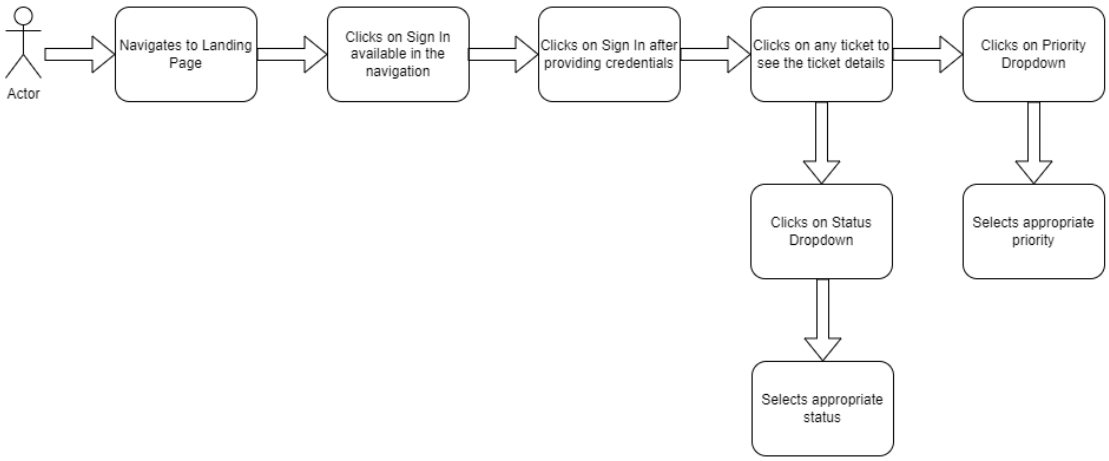


Figure 34: Click stream of set priority and update ticket status task [3].

Figure 34 shows click streams of set priority and update ticket status task of Ticket Management feature. The click stream starts with user navigating to landing page. After signing in, user can go to ticket detail page by clicking on any ticket. User can update the status and priority by selecting the appropriate option available in dropdown respectively.

Figure 33 shows the task flow diagram with normal and alternate flow of set priority and update ticket status task of Ticket Management feature.

Use Case - Forward or Involve Other Support Team:

1. The user logs into the Service Ticket Management System.
2. The user identifies the ticket with a complex technical issue.
3. The user selects the option to involve another support team.
4. The user chooses the specific team and assigns the ticket to them.
    4.1. If there is a system error preventing the assignment, The user receives an error message.
    4.2. The user checks the system status and ensures a stable internet connection.
    4.3. If the issue persists, the user contacts the IT support team via email or phone to address the technical problem.
    4.4. Once the technical issue is resolved, the user reassigns the ticket to the relevant support team.
5. The system generates notifications to inform the assigned team about their involvement.
6. The newly assigned team reviews the ticket details and begins collaborating with the user on the resolution.
7. The user monitors the progress and communicates with the team involved through the system's collaboration features.

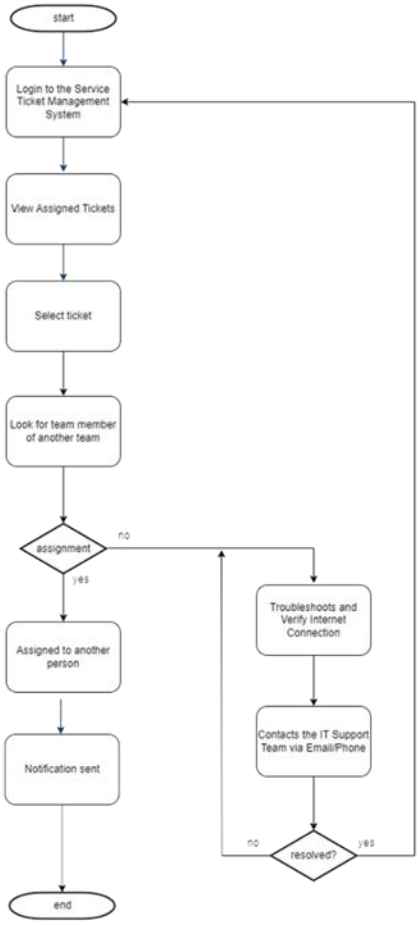


Figure 355: Task flow of Forward or Involve Other Support Team [3]

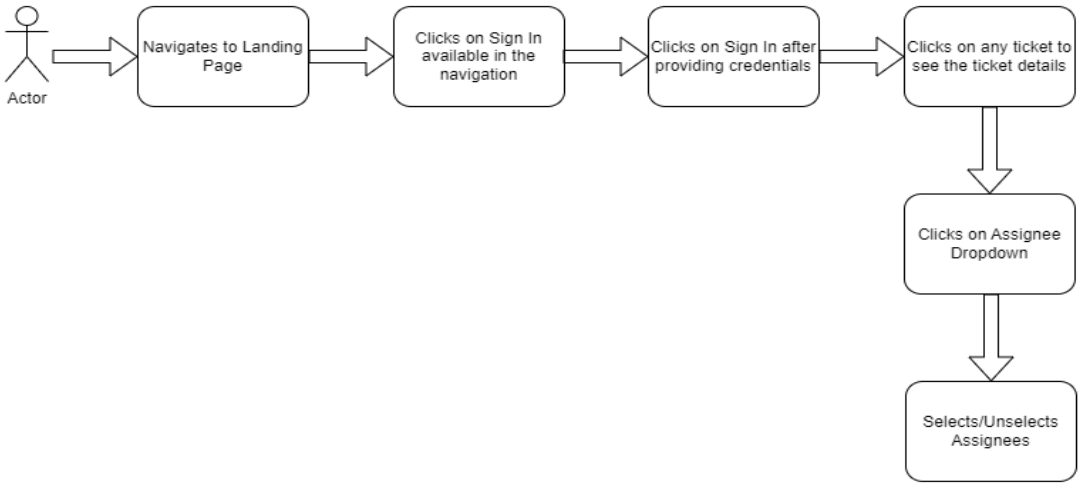


Figure 36: Click stream of updating assignee or forward to another team task [3].

Figure 36 shows click streams of updating assignee or forward to another team task of Ticket Management feature. The click stream starts with user navigating to landing page. After signing in, user can go to ticket detail page by clicking on any ticket. User can update assignees by selecting the appropriate option available in dropdown.

Figure 35 shows the task flow diagram with normal and alternate flow of updating assignee or forward to another team task of Ticket Management feature.

4.1.7 Feature: Manage the Board

Tasks:

- Add a new team
- Delete an existing team
- Edit an existing team

Use Case – Add a new team

1. User navigates to the "Manage Board" section.
2. User selects the option to "Add a New Team."
3. User fills in the required information such as team name, and description, and assigns team members.
4. User confirms and saves the new team details.
5. The system validates the information and adds the new team to the board.
    5.1. If there is any missing or incorrect information, the user receives an error message prompting her to correct it.

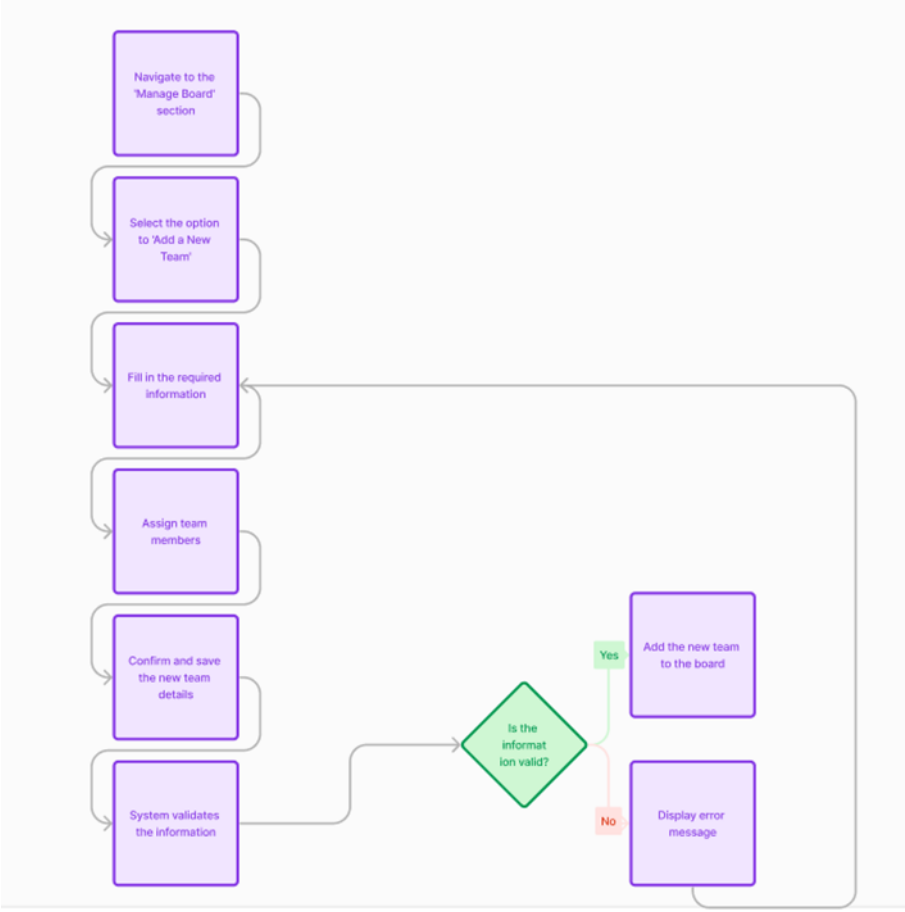


Figure 376: Task flow diagram of add a new team [3]

Use Case – Delete an existing team

1. User navigates to the "Manage Board" section.
2. User selects the team to be deleted from the list.
3. User confirms the deletion action.
    3.1. If there are any issues with deletion, such as ongoing tickets assigned to the team, the system notifies the user and guides resolving the issues.
4. The system removes the team from the board.

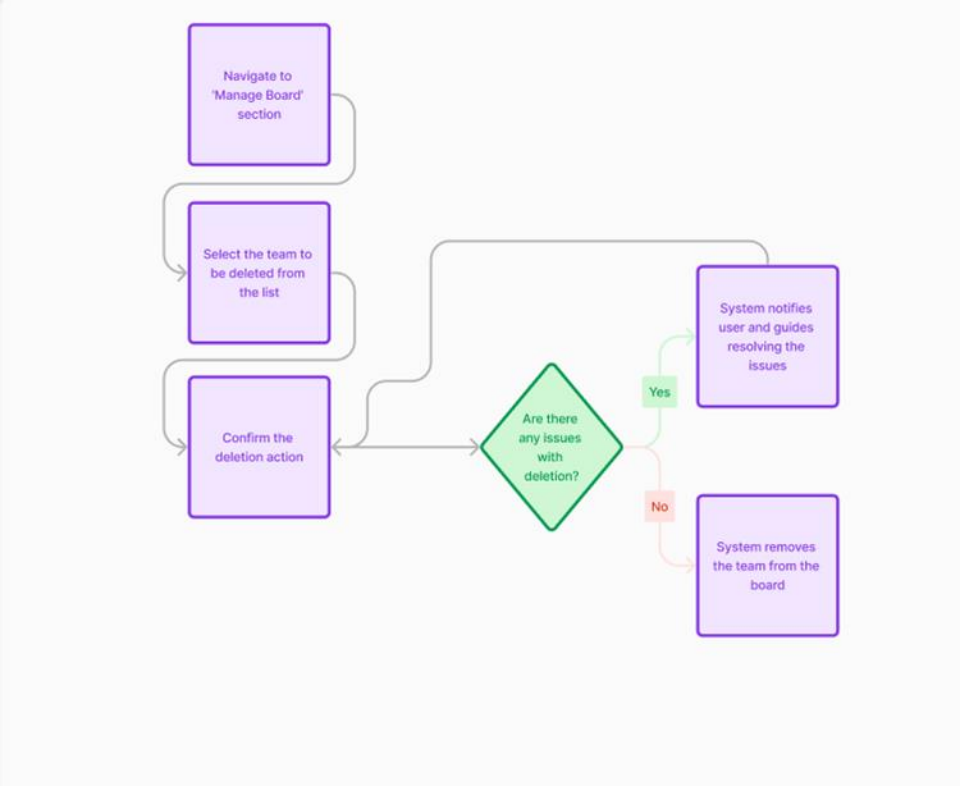


Figure 387: Task flow diagram of delete an existing team [3]

Use Case – Edit an existing team

1. User navigates to the "Manage Board" section.
2. User selects the QA team from the list of existing teams.
3. User modifies the team details such as team lead, members, or description.
4. User saves the changes.
5. The system validates the modifications and updates the existing team details.
    5.1. If there are any issues with the modification, User receives an error message and is prompted to correct the information.

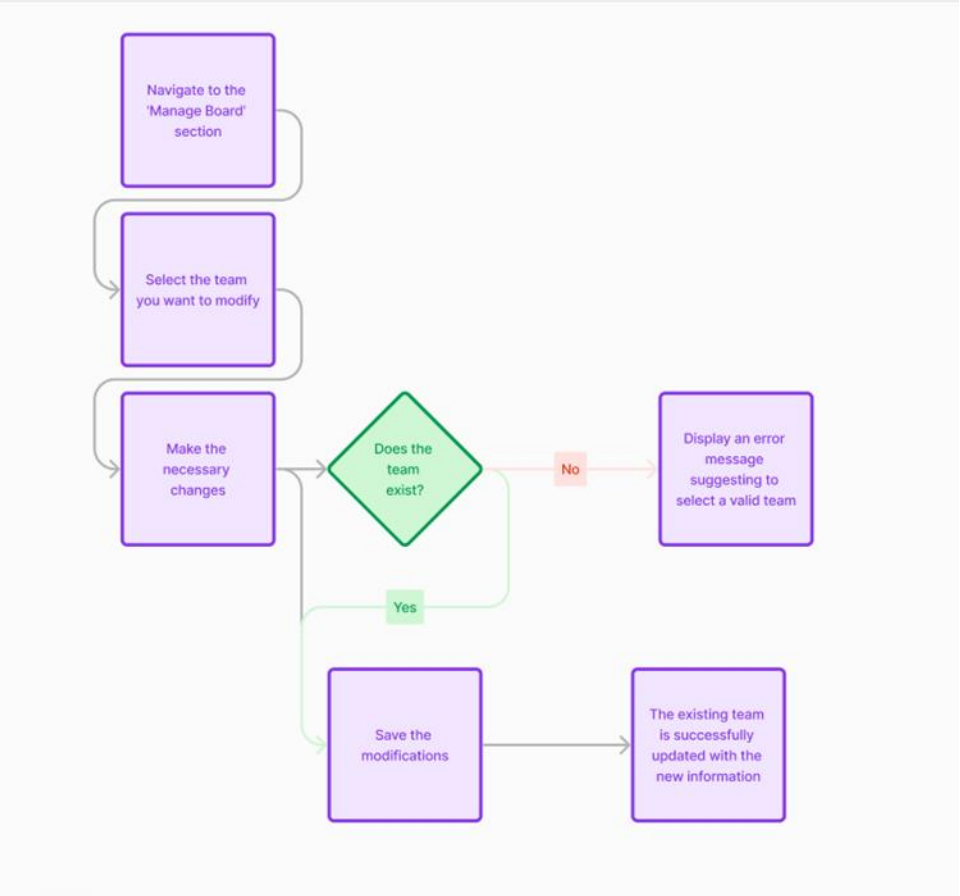


Figure 398: Task flow diagram of edit an existing team [3]

4.1.8 Feature: Profile Management/Settings

Tasks:

- Update Profile Name
- Change Password

Use Case – Update Profile Name

1. User logs into the service ticket management system using his credentials.
2. He navigates to the "Profile Settings" or "Account Settings" section.
3. User selects the "Edit Profile" option.
4. He locates the "Name" field and modifies it with his new legal name.
5. After making the changes, User reviews the updates and confirms the changes.
6. The system validates the changes and updates the user's profile name accordingly.

6.1.f User encounters any errors while updating the profile name, such as entering invalid characters or exceeding the character limit:

6.2.The system prompts User to correct the errors or provide a valid name.

6.3. User makes the necessary corrections and submits the updated profile name again.
6.4. Steps 5-7 are repeated until the user's profile name is successfully updated.

7. User receives a confirmation message indicating that his profile name has been successfully updated.

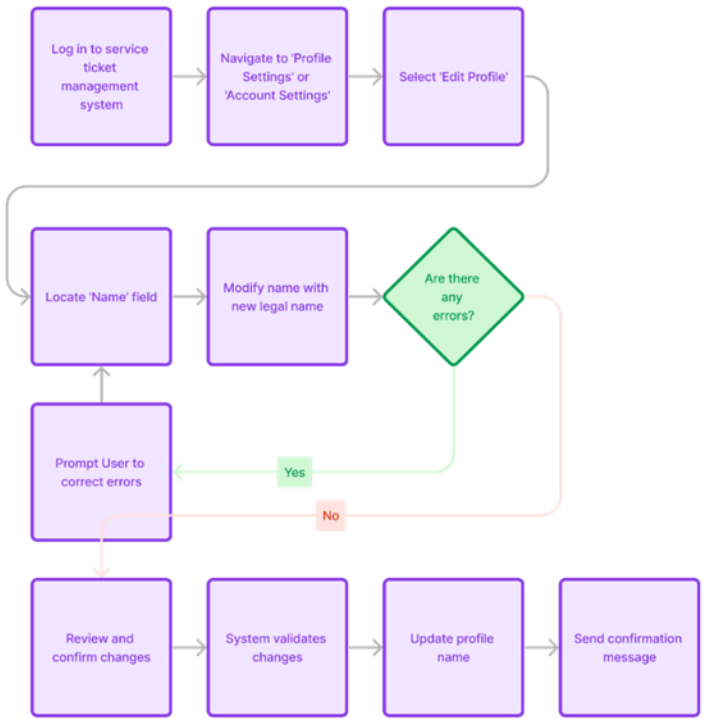


Figure 409: Flowchart applicable for Updating Profile name [3].

Use case - Change Password

1. The user logs into the service ticket management system using his current credentials.
2. He navigates to the "Profile Settings" or "Account Settings" section.
3. User selects the "Change Password" option.
4. He enters his current password and then provides a new password as per the system's password policy.
5. After entering the new password, User confirms the password change.
    5.1. If User encounters any issues while changing the password, such as entering an incorrect current password or not meeting the password policy requirements:
    5.2. The system prompts User to correct the errors or ensure the new password meets the criteria.

5.3. User makes the necessary corrections and submits the password change request again.
5.4. Steps 5-7 are repeated until User's password is successfully changed.

6. The system validates the new password and updates the user's account with the new password.
7. User receives a confirmation message indicating that his password has been successfully changed.

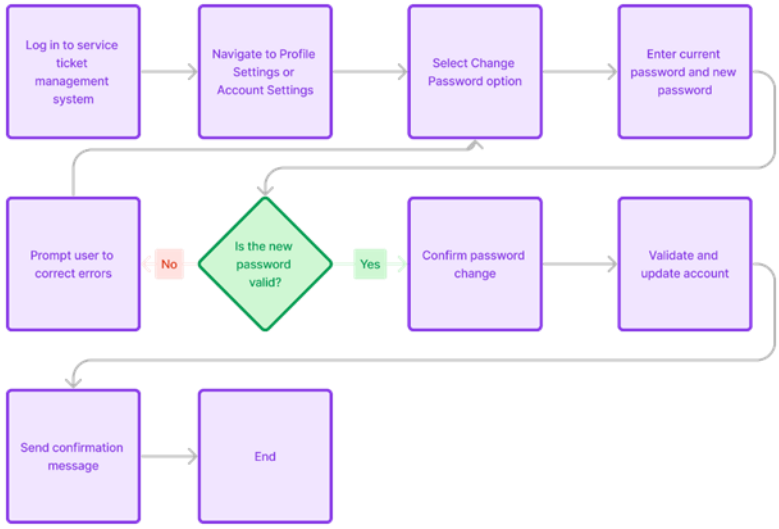


Figure 4110: Flowchart for updating password [3]

.

4.2 Process and Service Workflow

4.2.1 Application Workflow:

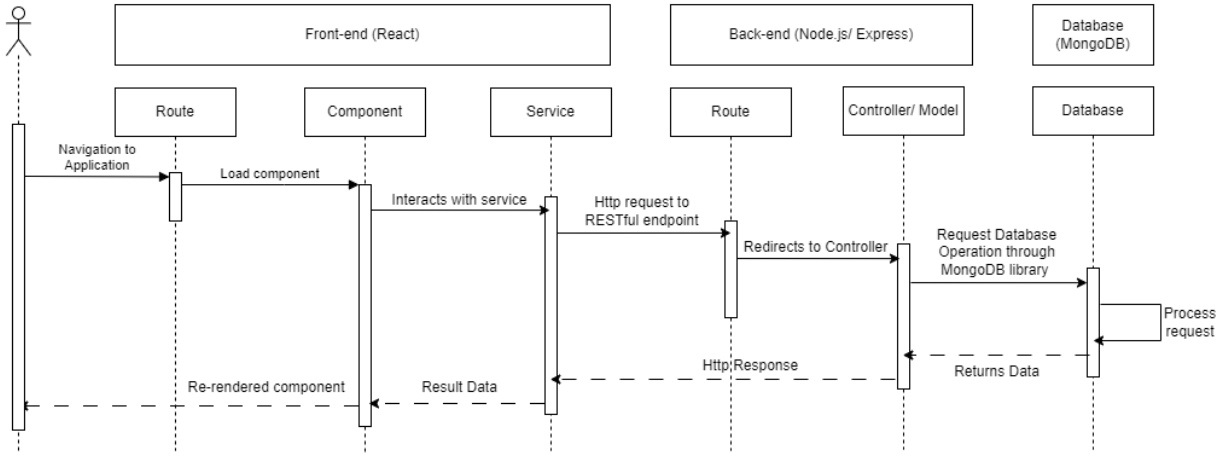


Figure 42: Application workflow diagram [3].

In Figure 42, the interaction flow during user actions is illustrated. React Router matches user-requested routes, loading the associated component. This component communicates with a service, triggering an HTTP request to the server. The backend controller, containing route-specific business logic, interacts with MongoDB to retrieve real-time data. The controller sends the data back to the React service, prompting a component re-render.

5. Asset Inventory

5.1 Client-side languages:

- React: Renders user interface components, handles client-side validation, and makes API calls to the server.
- CSS: Provides custom styling to React-rendered components and adjusts design responsiveness.
- ANT design: Simplifies React component design with pre-styled, responsive components.

5.2 Server-side languages:

- Node: Node.js powers our website's backend, managing server-side operations such as handling requests, processing data, and interacting with databases.

5.3 APIs and Dependencies:

1. **@ant-design/icons**: Provides a collection of icons for enhancing the user interface with visual elements.

2. **@dnd-kit/sortable**: Enables drag-and-drop functionality for reordering items within lists or containers.

3. **antd**: Offers a wide range of customizable UI components for building modern web applications with React [7].

4. **apexcharts**: A JavaScript charting library for creating interactive and visually appealing charts and graphs.

5. **moment**: A lightweight library for parsing, validating, and manipulating dates and times in JavaScript.

6. **react-apexcharts**: React wrapper for ApexCharts library, facilitating the integration of charts into React applications.

7. **react-dom**: Facilitates rendering React components into the browser's DOM for display.

8. **react-router-dom**: Provides declarative routing for navigating between different views or pages in React applications.

9. **react-scripts**: Command-line tools for managing React projects, including development server and bundling utilities.

10. **react-type-animation**: A React component for creating typewriter-style animations to enhance user experience.

11. **styled-components**: Enables styling of React components with scoped styles and dynamic capabilities for enhanced UI.

6. CONCLUSION

Issuestack can successfully enter into the competitive market as a service ticket management platform by offering a user-friendly platform which caters to Employers, Employee and Service Help Desk Representatives. Use of this quick and simple tool is facilitated by its unique features and minimalist appearance. By solving their issues with unique features, this platform seeks to deliver a positive user experience, having successfully captured the sentiments of its target consumers.

7. RECOMMENDATIONS

The program must be implemented with a microservices design that enables vertical and horizontal scaling with optimization if worldwide success is to be guaranteed. Developers should also concentrate on making the most of the application's special features and enrich knowledge base articles for users who want to access them before raising a ticket.

8. REFERENCES

[1] "Zendesk: The complete customer service solution," *Zendesk*, 22-Mar-2010. [Online]. Available: https://www.zendesk.com/. [Accessed: 1-Feb-2024].

[2] "ServiceNow - Put AI to Work", *Servicenow.com*. [Online]. Available: https://www.servicenow.com/. [Accessed: 1-Feb-2024].

[3] "Figma: The Collaborative Interface Design Tool," *Figma*. [Online]. Available: https://www.figma.com. [Accessed: 28-Feb-2024].

[4] A. Cakir, "Create powerful business content together with," *Xtensio*, 17-Jun-2023. [Online]. Available: https://xtensio.com/. [Accessed: 28-Feb-2024].

[5] "Intelligent diagramming," *Lucidchart*. [Online]. Available: https://www.lucidchart.com/. [Accessed: 28-Feb-2024].

[6] "Balsamiq: Fast, focused wireframing for teams and individuals," *Balsamiq.com*. [Online]. Available: https://balsamiq.com/. [Accessed: 1-Mar-2024].

[7] "Ant Design - The world's second most popular React UI framework," *Ant.design*. [Online]. Available: https://ant.design/. [Accessed: 1-Mar-2024].

[8] "Scale & ship faster with a composable web architecture," *Netlify.com*. [Online]. Available: https://www.netlify.com/. [Accessed: 1-Mar-2024].

[9] "Npm," *Npmjs.com*. [Online]. Available: https://www.npmjs.com/. [Accessed: 1-Mar-2024].

[10] L. M. A. Extensions and L. A. V. Dev, "Visual Studio Code - code editing. Redefined," *Visualstudio.com*. [Online]. Available: https://code.visualstudio.com/. [Accessed: 1-Mar-2024].

[11] "IssueStack," *Netlify.app*. [Online]. Available: https://csci5709-web-project.netlify.app/. [Accessed: 5-Apr-2024].

[12] "Cloud application hosting for developers," *Render.com*. [Online]. Available: https://render.com/. [Accessed: 5-Apr-2024].