Supportive Application

Group No: 02

CSH-4112: Group Project

Department of Physical Science

Faculty of Applied Science

University of Vavuniya

2023

Group Members:

Registration Number	Name
J.Neethini	2017/ASP/03
J.Tuvensha	2017/ASP/18
I.Tharsika	2017/ASP/19
S.Bramyah	2017/ASP/33
J.Jenusiya	2017/ASP/39
N.Tharaha	2017/ASP/43
A.Tharshini	2017/ASP/59

ACKNOWLEDGEMENT

Here we gladly present this project report on “Supportive Application” as our activity in the system analysis design course. At this time of submitting this report, we use this opportunity to mention those people who were with us throughout the work.

We owe a great thanks to many people who helped and supported us during the designing and completing of our project as well as the writing of this report also.

We extend our sincere and heartfelt thanks to our esteemed guide, Mr. S. Selvendra for providing us with the right guidance and advice at crucial junctures and for showing us the right way by supporting us to complete each milestone of our project from the beginning to the end.

We would like to thank our friends for the support and encouragement they have given us during the course of our work toward success.

Thank You.

Contents

1. Overview of the Project
    - 1.1 . Goal of the Project
    - 1.2 . The objective of the Project

2. Design
    - 2.1 . Diagrams
    - 2.2 . User Interface
3. Methodology
    - 3.1. Justification for the chosen Methodology

4. Software Tool Used

5. Requirements
    - 5.1. Functional Requirements
    - 5.2. Non-Functional Requirements

6. Testing

7. Conclusion

8. Reference

1. Overview of the Project

A supportive application is an electronic application designed to provide support to users in some capacity. The goal of the project is to create an application that is easy to use and provides the necessary support for users. Our project involves creating a web-based application.

The project began with a comprehensive analysis of the user's needs. The analysis also included an assessment of the user's existing support networks and any gaps in those networks. After the analysis was complete, our team developed a plan for building the application. This included selecting the appropriate technology, designing the features and user interface, and developing the application.

The project typically involved a team of developers (Front end developers, Back-end developers), designers, etc. The responsibility of the developers was, to create the application and ensure that it meets the user's needs. The responsibility of designers was, to design the ER diagram, sequence diagram, class diagram, and architectural design and make an overview of the application, and ensure that the application is easy to use.

The product manager was responsible for providing feedback on the project and ensuring that it meets expectations and also communicating with the customer.

The project was broken down into a series of tasks, with each task assigned to a member of our team. These tasks should be tracked and managed to ensure that the project is completed on time and within budget.

The project included testing to ensure that the application works as expected. Testing included both manual and automated testing. Automated testing should include both unit and integration tests. Finally, the application is released to the public.

1.1. The goal of the Project

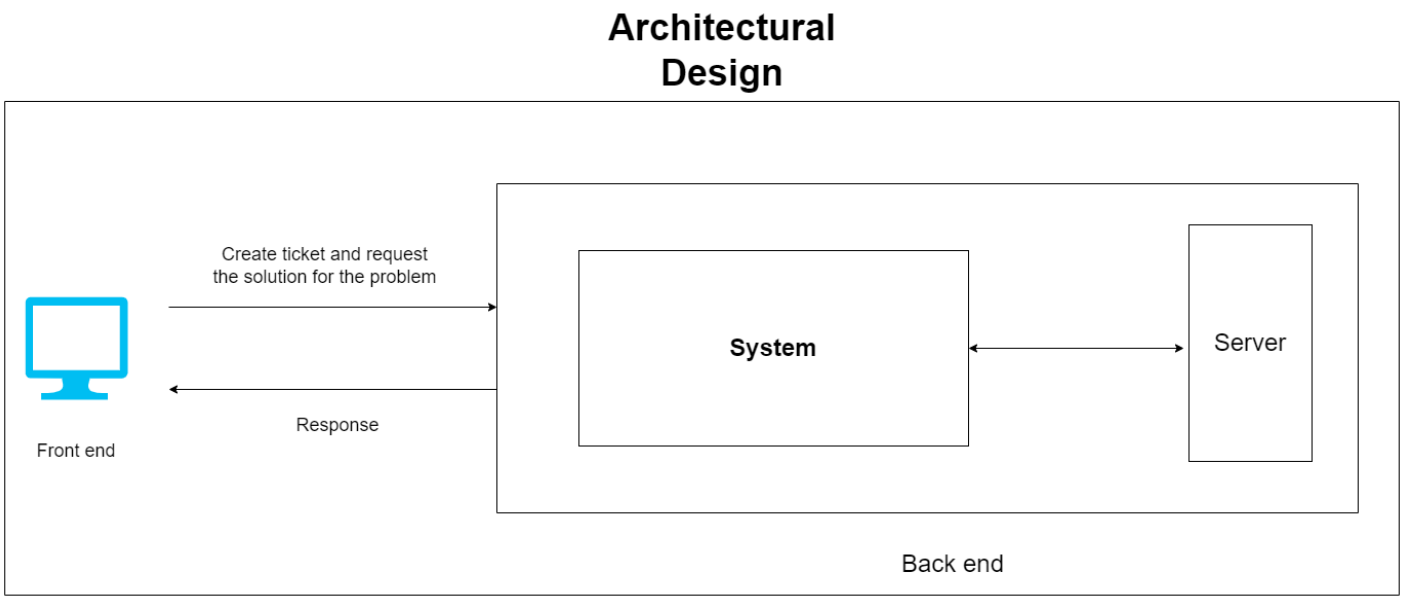
- To create an application that is easy to use and provides the necessary support for users.

1.2. The objective of the Project

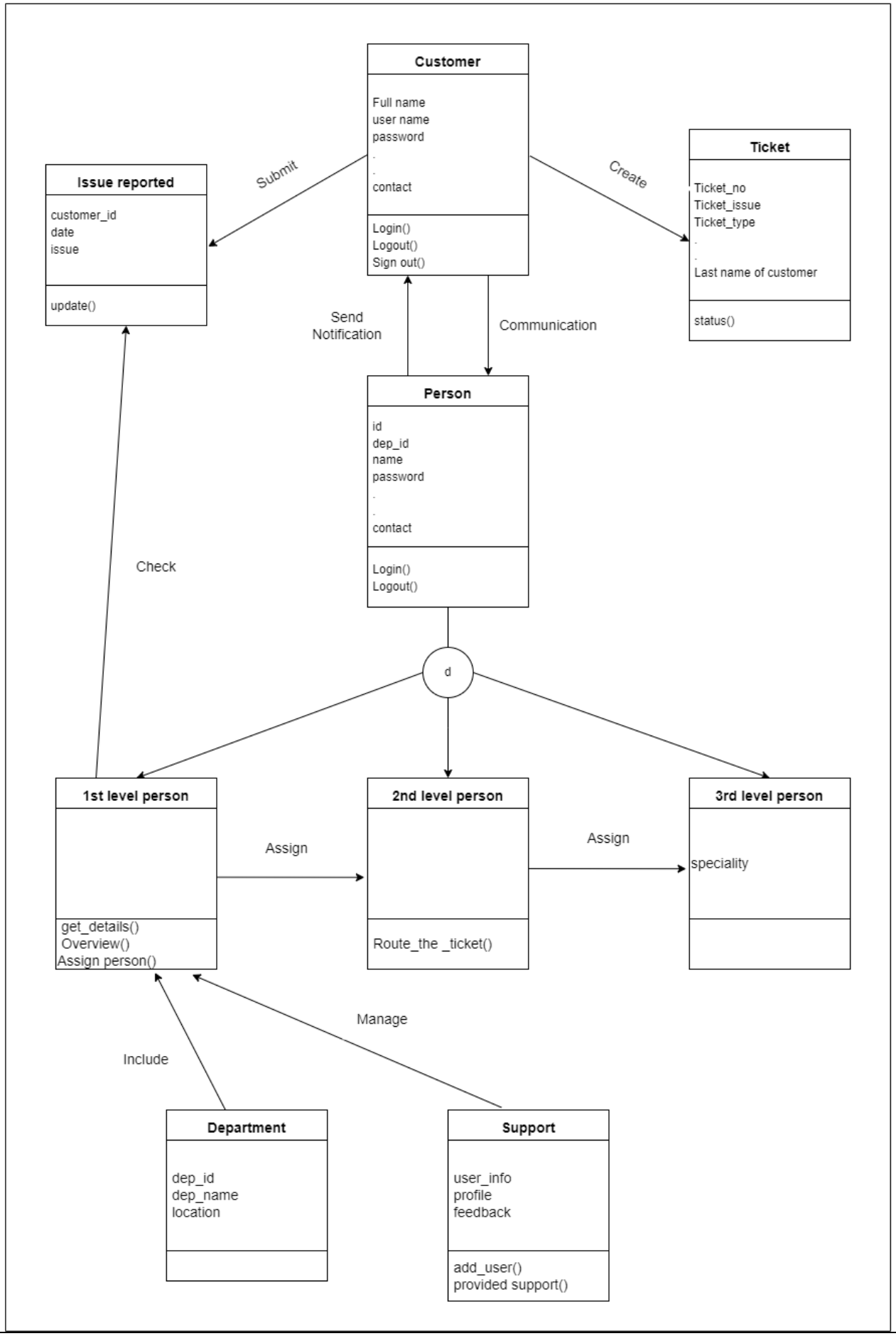
- Providing access to resources
- Helping users manage their tasks
- Creating an FAQ page
- Helping users find the information they need
- Connecting users to customer service agents
- Providing diagnostic feedback for troubleshooting technical issues
- Providing advice and guidance
- Helping users with decision-making
- Solving customer's problems or issues

2. Design

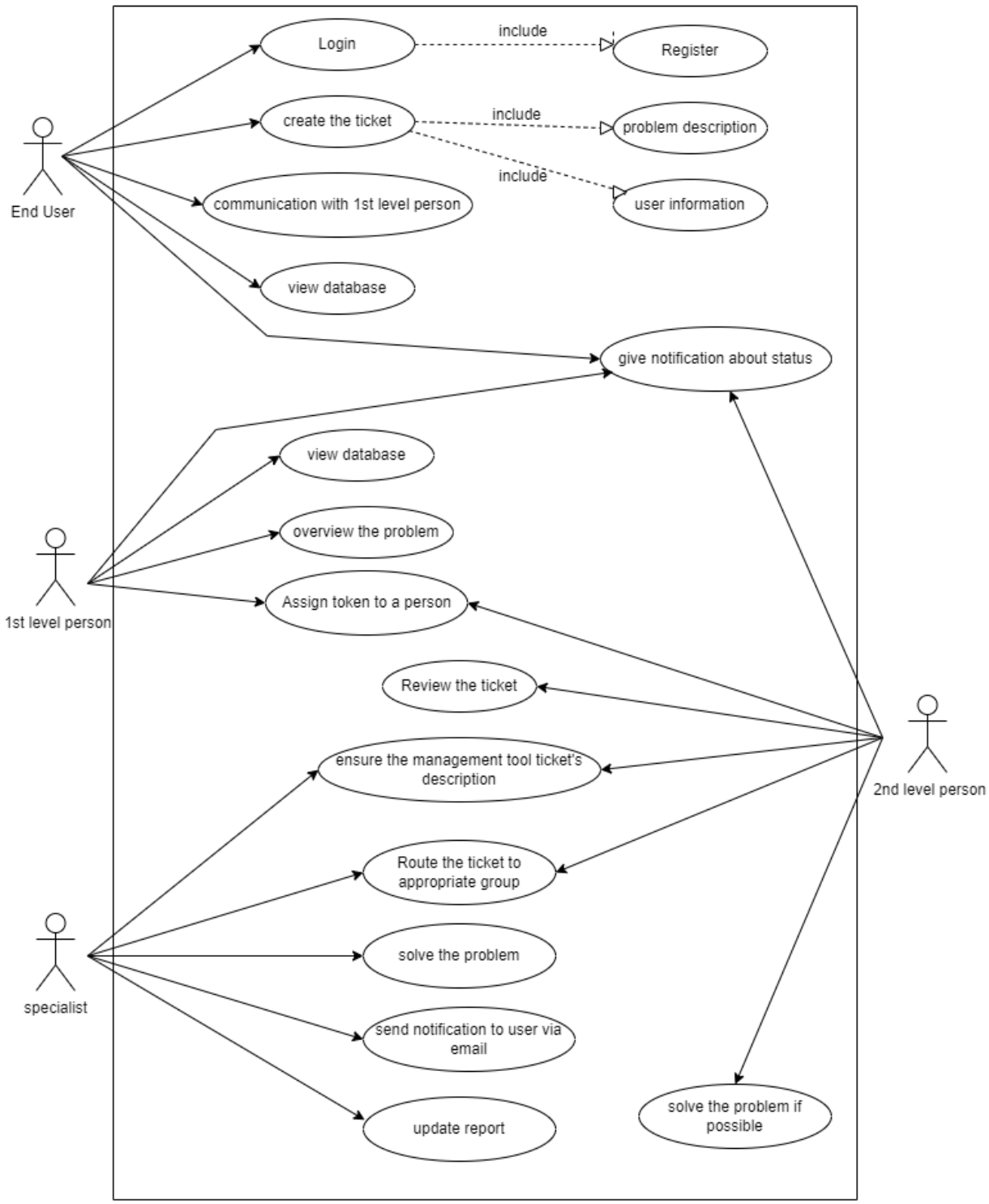
2.1. Diagrams



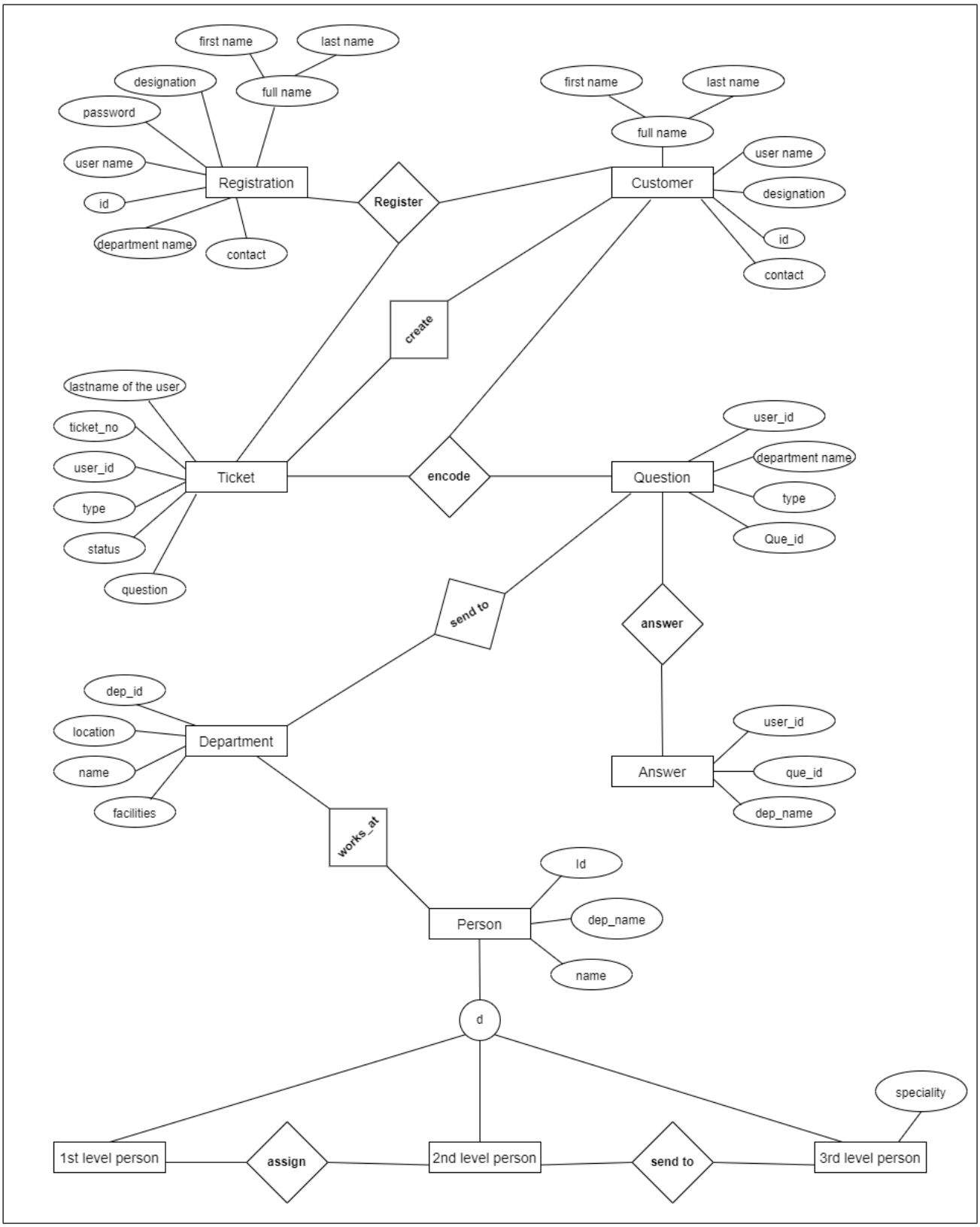
Class Diagram



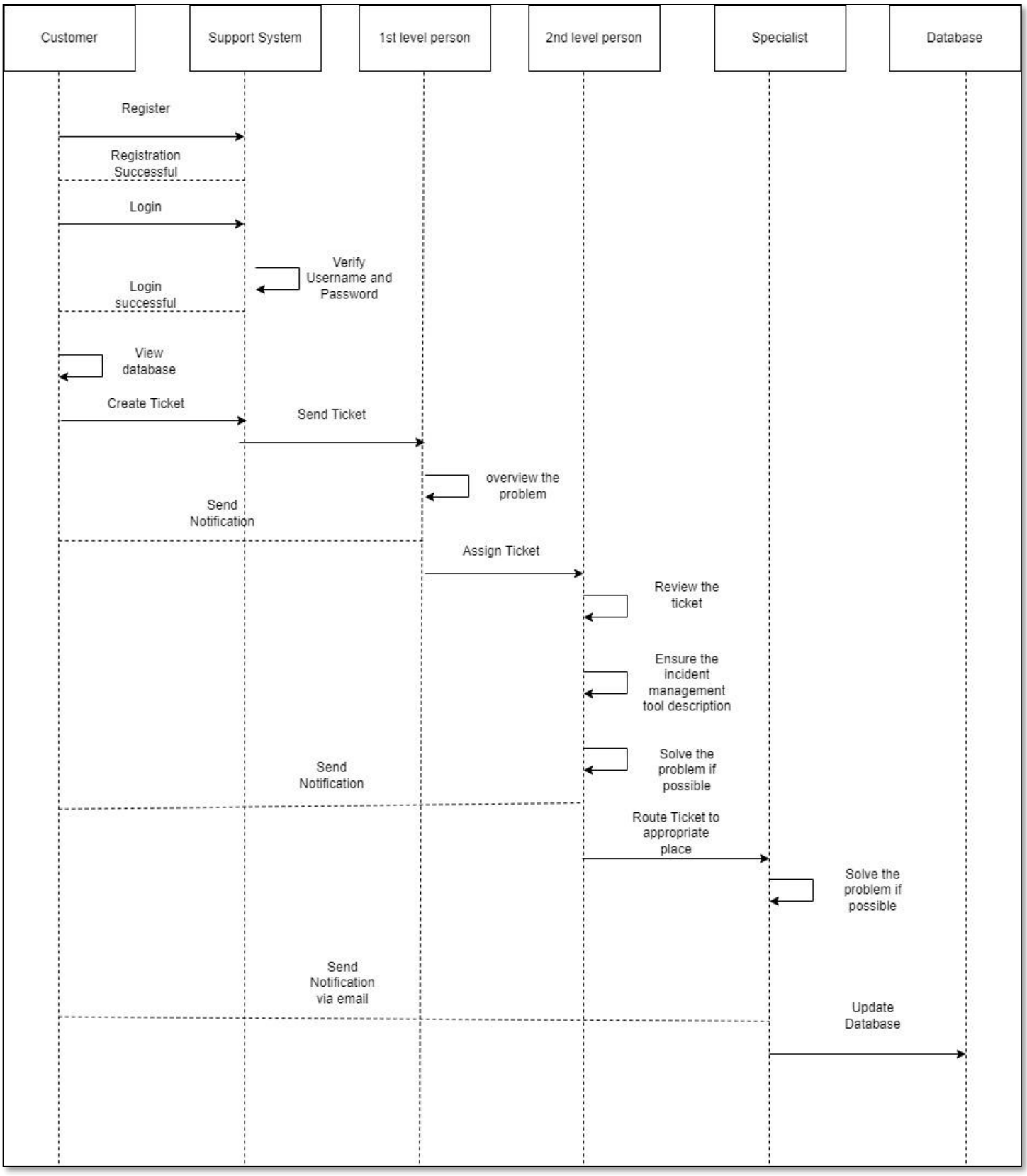
Usecase Diagram

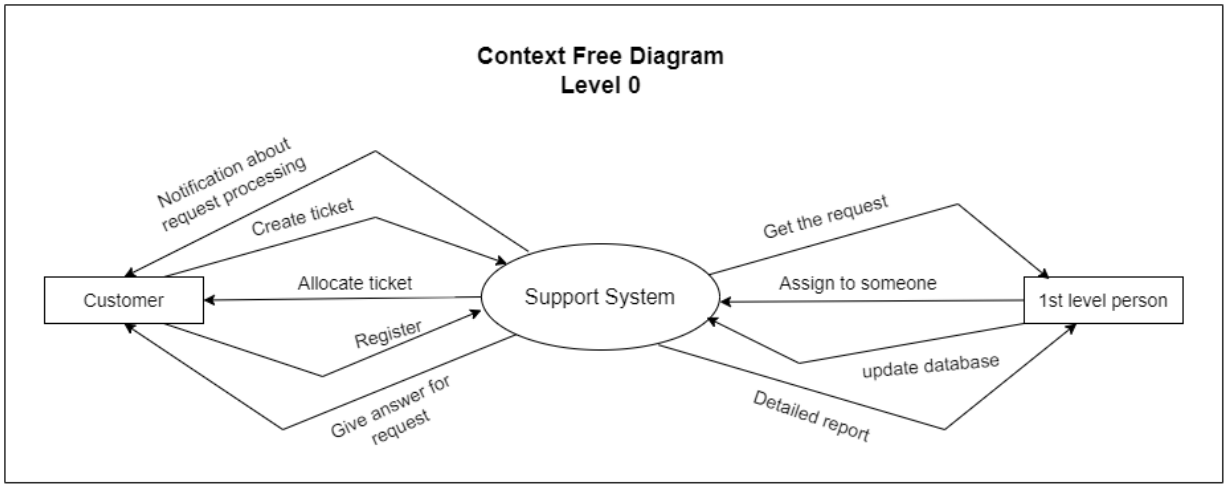


ER Diagram


first name
last name
full name
designation
password
user name
id
department name
contact
Registration
Register
first name
last name
full name
user name
designation
id
contact
Customer
create
lastname of the user
ticket_no
user_id
type
status
question
Ticket
encode
user_id
department name
type
Que_id
Question
send to
answer
dep_id
location
name
facilities
Department
works_at
user_id
que_id
dep_name
Answer
Id
dep_name
name
Person
d
speciality
1st level person
assign
2nd level person
send to
3rd level person

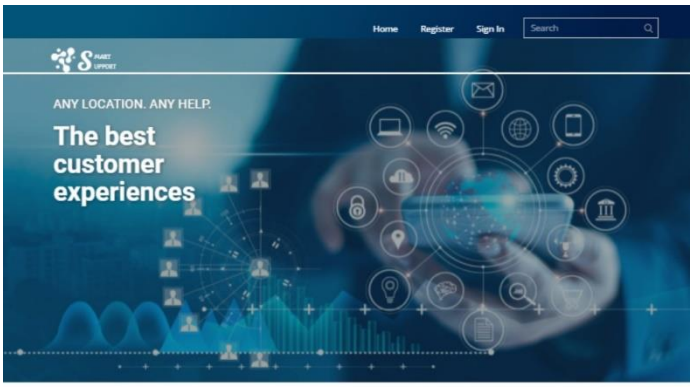
Sequence Diagram

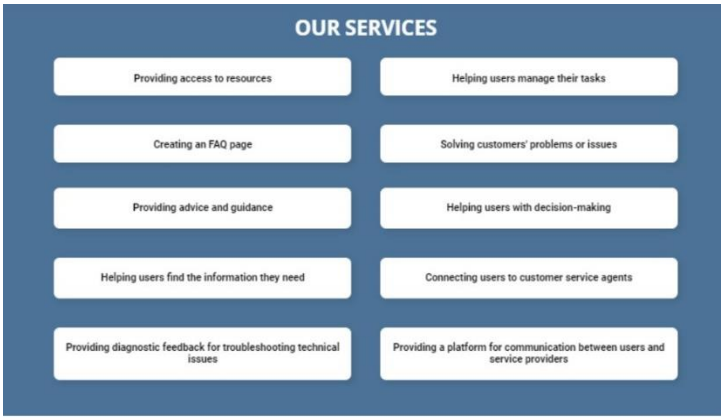


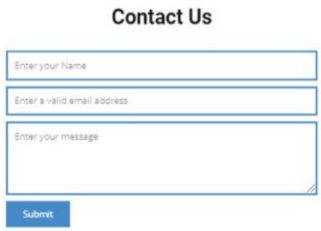

Context Free Diagram
Level 0
Notification about request processing
Create ticket
Allocate ticket
Register
Give answer for request
Customer
Support System
Get the request
Assign to someone
update database
Detailed report
1st level person

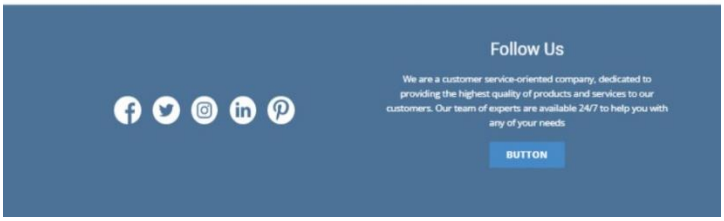
2.2. User Interface Design

Home Page

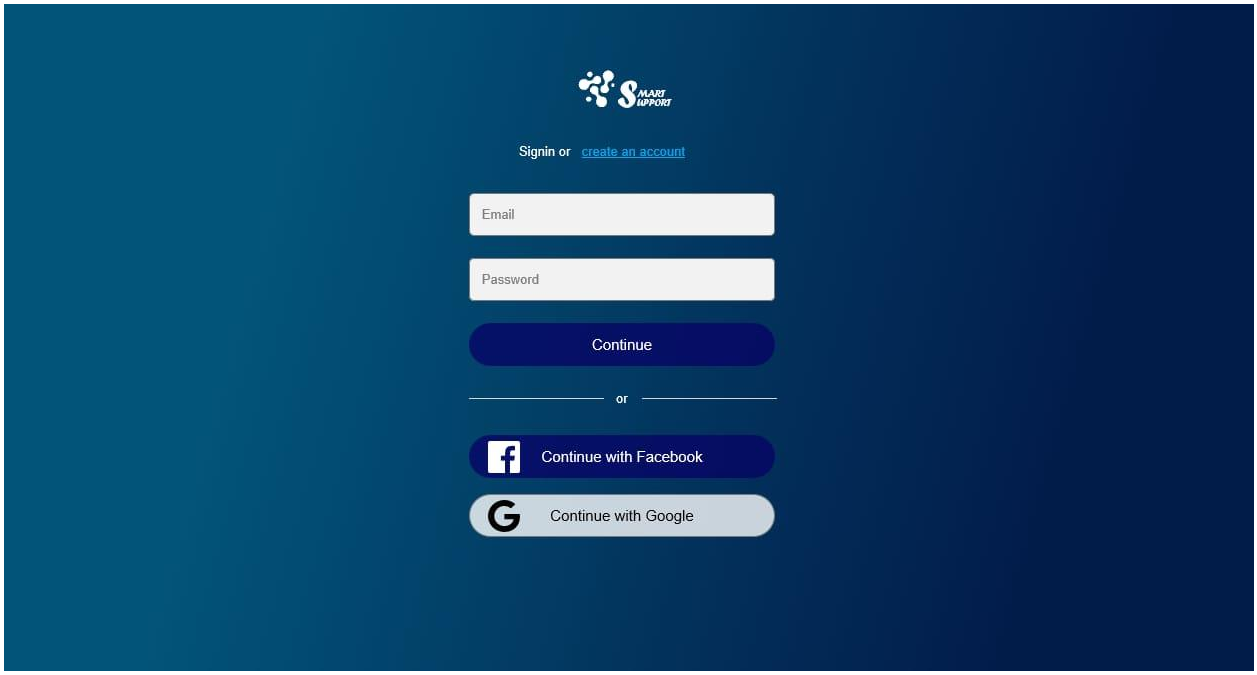




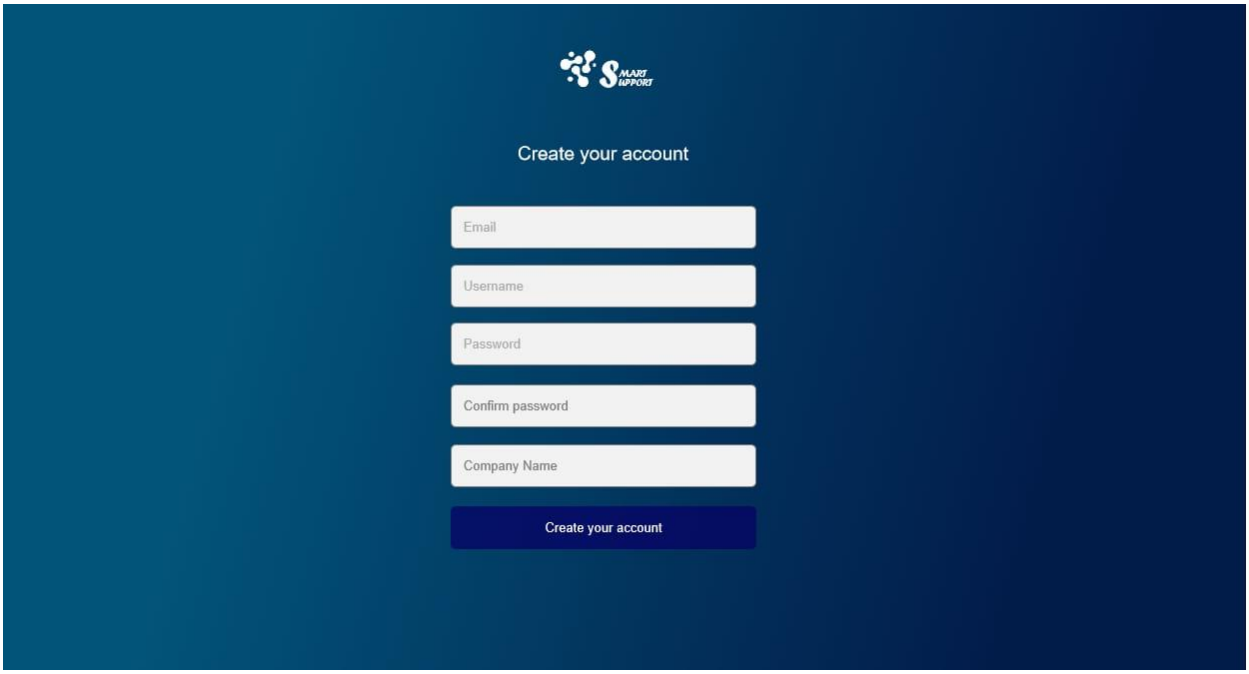




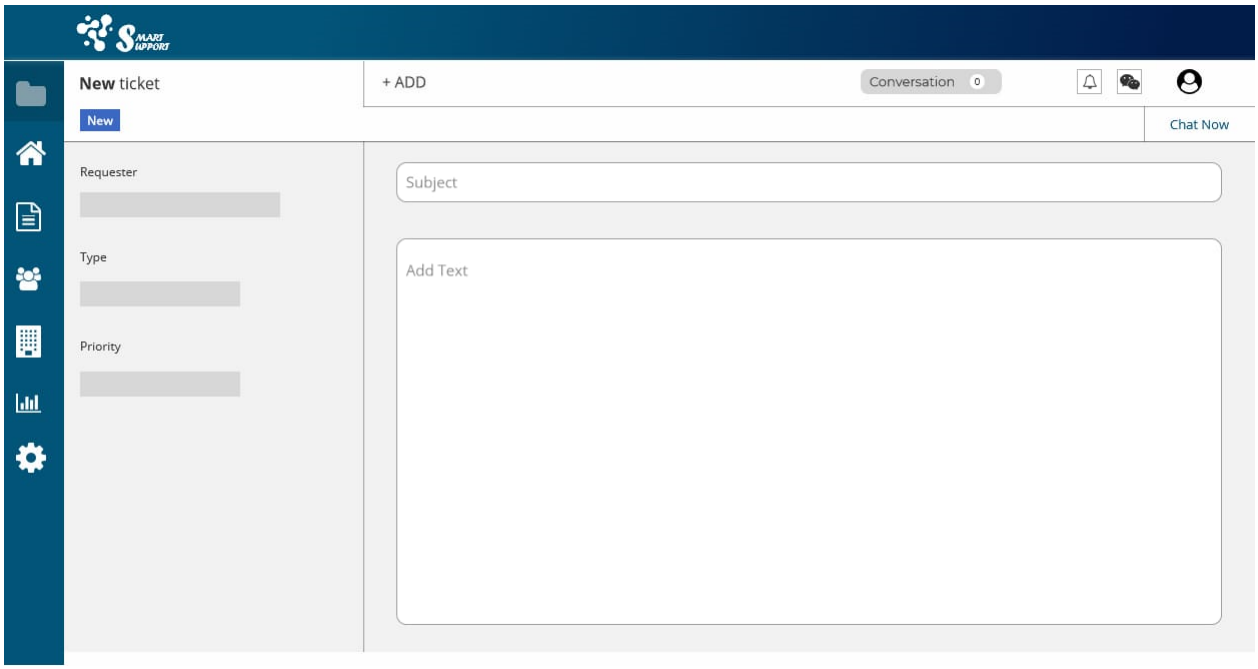
Sign Up/Login



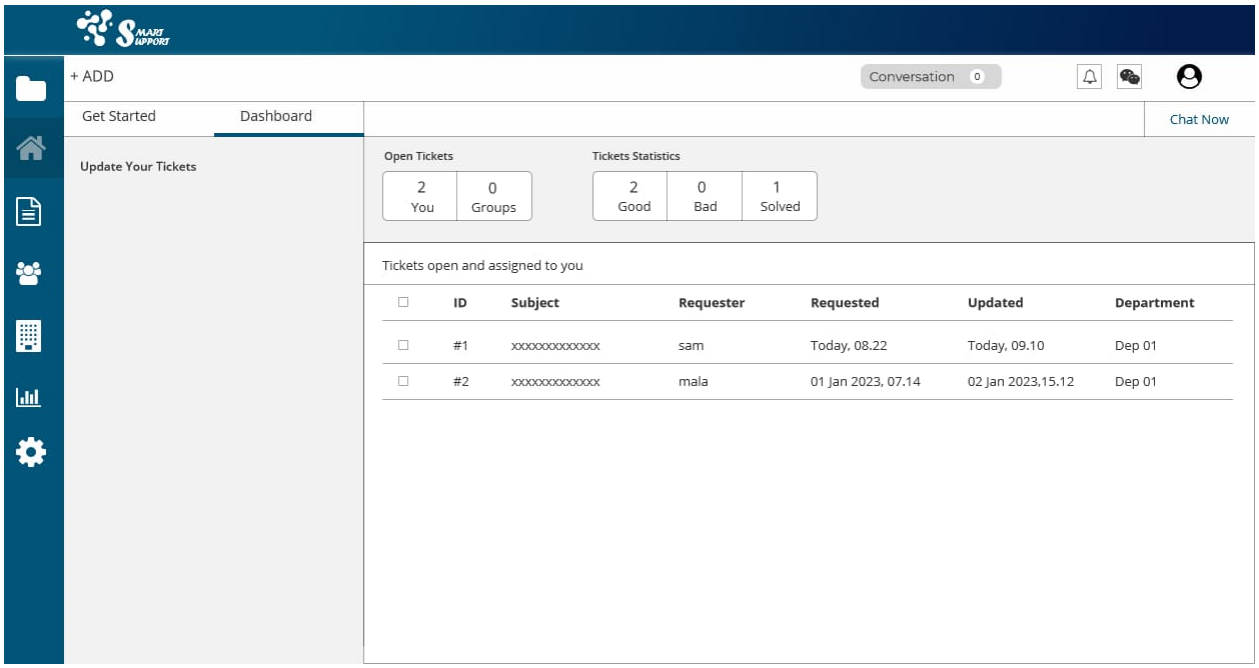
Create Account



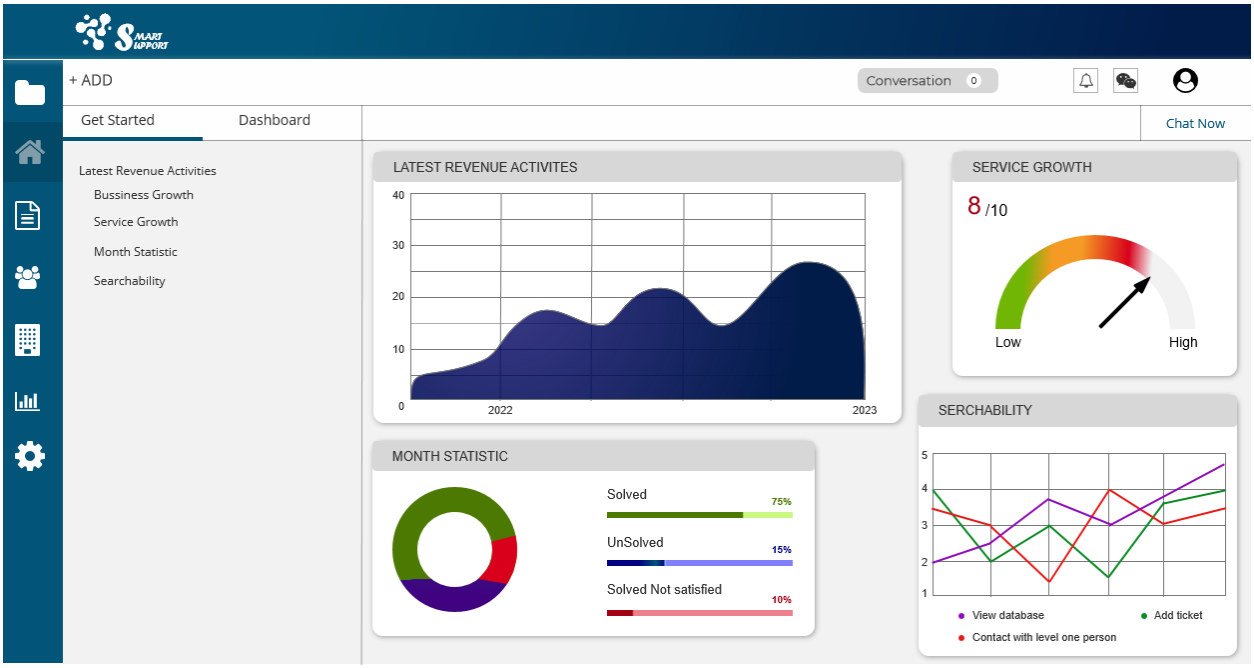
Create Ticket



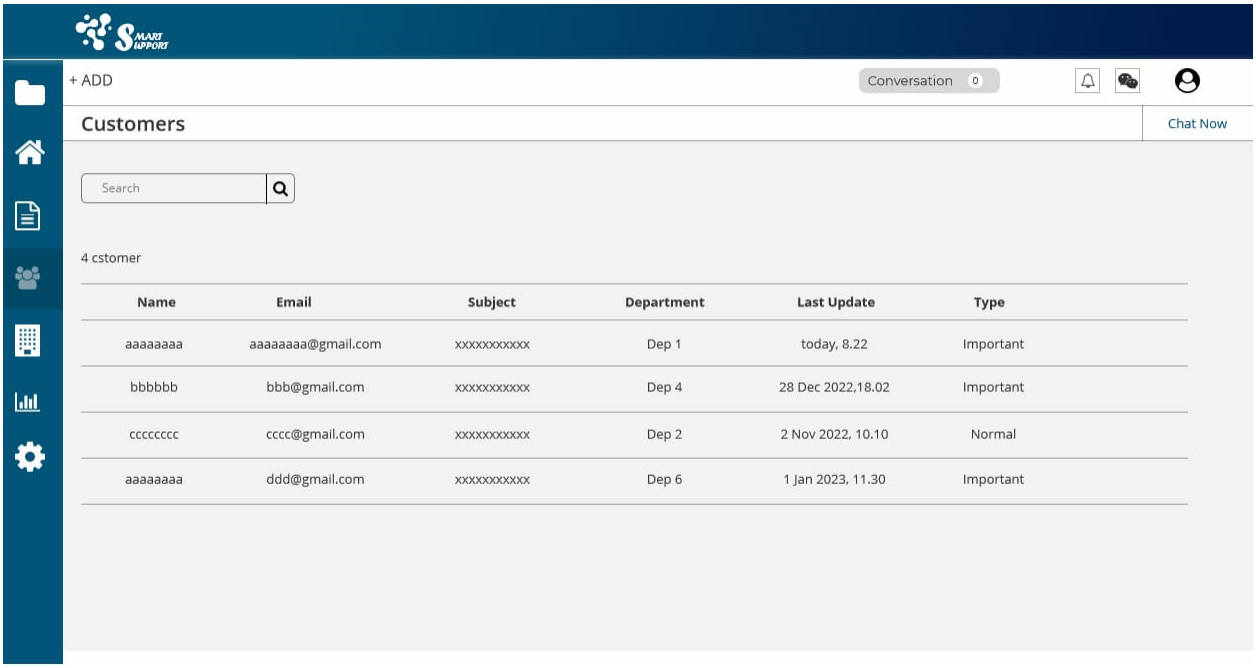
Database



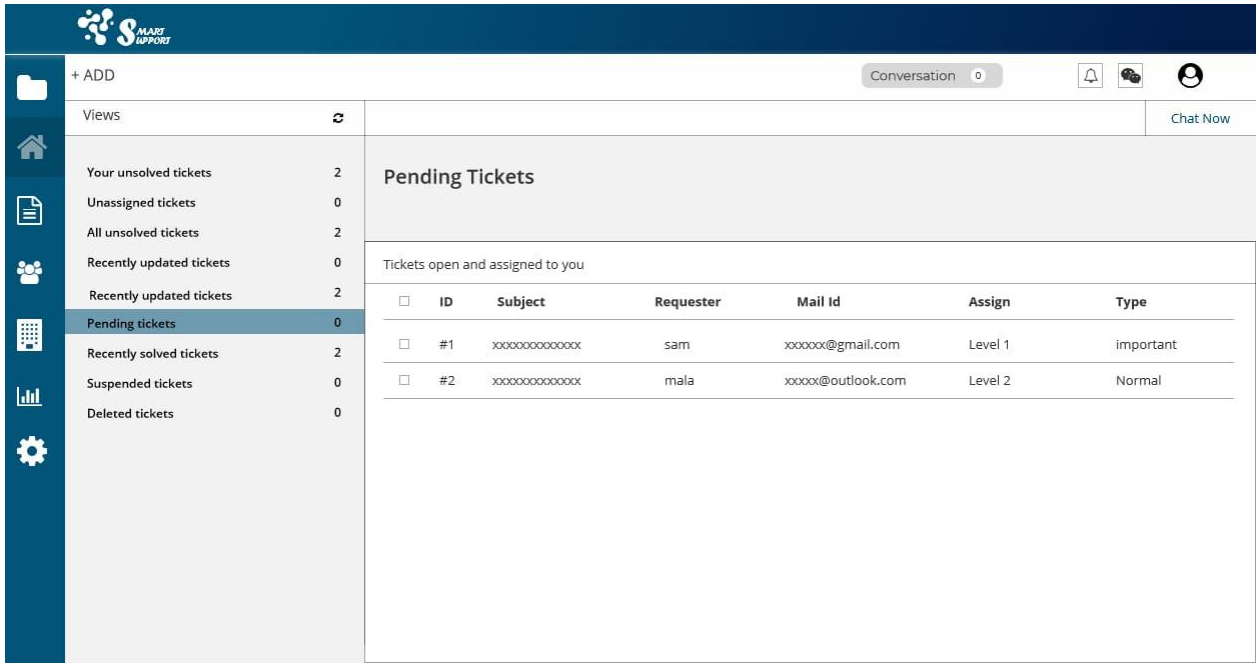
Annual Reports

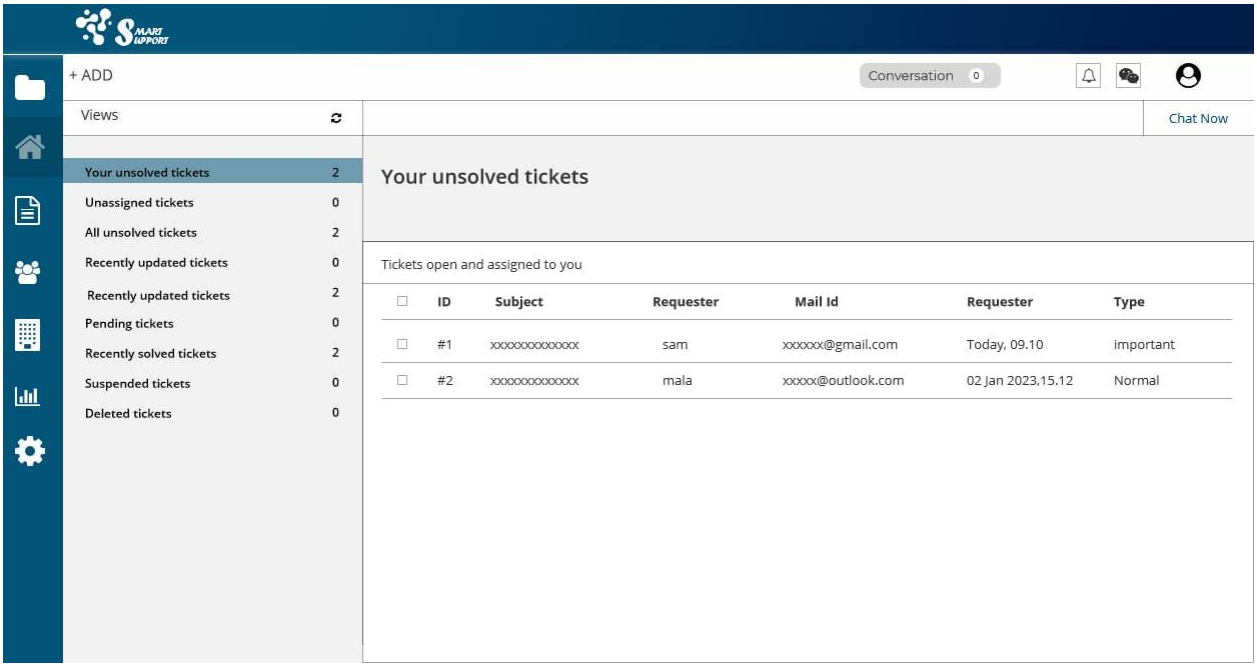


Customer Details

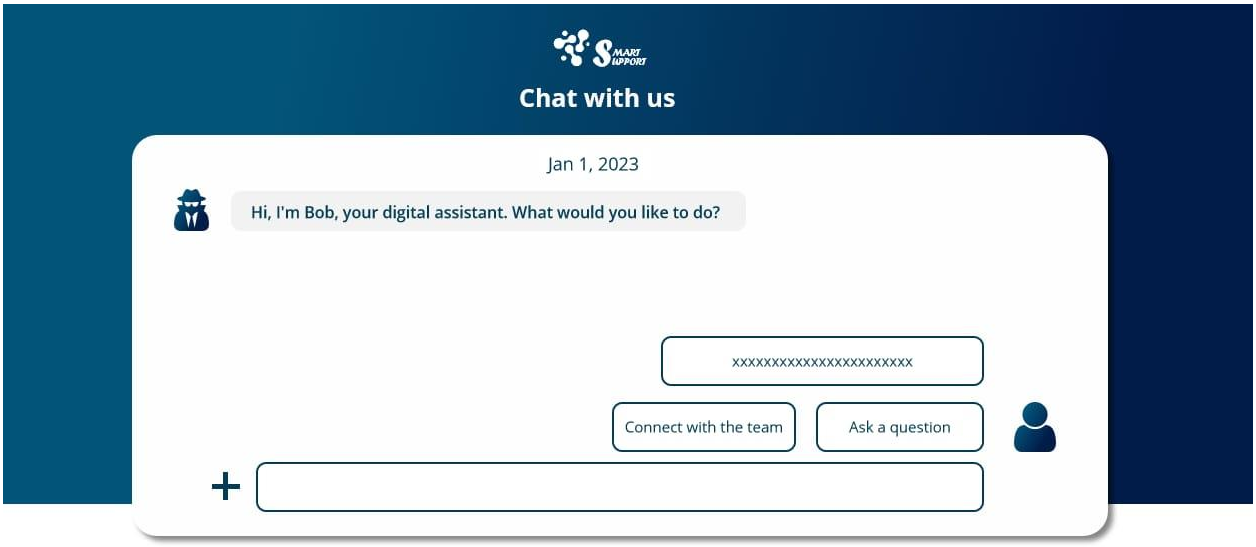


Ticket Details





Chatbox



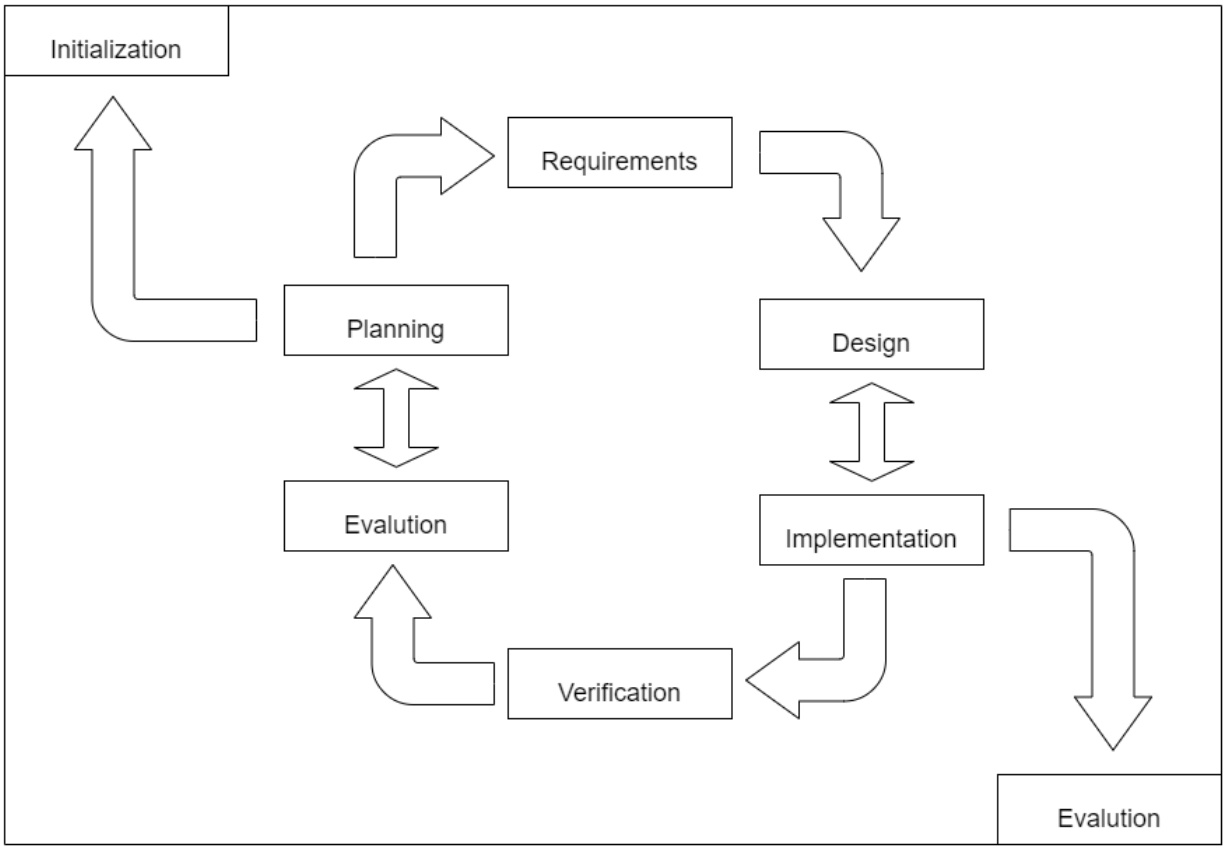
3. Methodology

We used the **Agile development model** to fulfill our project. Agile Process Model is an iterative and incremental approach to software development that is used in software engineering. It is an adaptive, lightweight, and flexible process model which focuses on customer satisfaction through early and continuous deliveries of valuable software. The Agile Process Model emphasizes on collaboration between self-organizing and cross-functional teams to deliver the highest value to the customers. It is based on the values and principles of the Agile Manifesto which includes customer satisfaction, individuals and interactions, working software, responding to change, etc. The Agile Process Model is designed to respond to changing customer needs and requirements. It encourages the teams to inspect and adapt the process based on the user's feedback. This helps in reducing the risk of delivering the wrong product and ensures that the customer gets the product that meets their needs.

Agile methods break tasks into smaller iterations or parts that do not directly involve long-term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration, and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

The iterative and incremental method consists of 8 phases namely planning, requirements, analysis and design, implementation, deployment, testing, evaluation, and initial planning.


Initialization
Requirements
Planning
Design
Evalution
Implementation
Verification
Evalution

AGILE
METHODOLOGY
The Agile Process
SCRUM MASTER
DAILY SCRUM
PRODUCT OWNER
TEAM
PRODUCT BACKLOG
SPRINT PLANNING MEETING
SPRINT BACKLOG
FINISHED WORK
SPRINT REVIEW
SPRINT RETROSPECTIVE
BYDREC

3.1. Justification for the chosen Methodology

- Agile process model helps to deliver the software faster and with more accuracy, leading to increased customer satisfaction.
- Agile process model brings together the customer, the development team, and the stakeholders in a collaborative environment to ensure everyone is on the same page.
- Agile process model allows for changes to be made quickly and easily during the development process, which makes it more flexible and adaptive.
- Agile process model provides visibility into the development process, allowing all stakeholders to monitor progress and provide feedback.
- Agile process model encourages continuous testing and feedback, which helps to ensure high-quality software.
- Agile process model helps to save costs by reducing waste and improving efficiency.
- Agile process model allows for faster development cycles, leading to faster delivery of the software.

4. Software Tool Used

1. Front End
    - Html
    - Java Script
    - CSS

2. Back End
    - PHP

3. Database
    - My SQL

5. Requirements

5.1. Functional Requirements

- Ability to create and manage user profiles.
- Ability to provide tailored content and resources to users.
- Ability to facilitate communication between user and 1st level person.
- Ability to integrate with third-party services.
- Ability to provide personalized notifications to users.
- The application must be able to authenticate and authorize users.
- The application must be able to access and store data in a database.
- The application must be able to generate reports.
- The application must be secure and must protect user data.
- The application must be able to be used on mobile devices.
- The application must provide reporting capabilities to allow users to view statistics, trends and insights.
- The application must provide a ticket management system which allows customer to create tickets
- The application must provide an easy-to-use interface which is accessible to users of all skill levels.
- Ability to create and manage user accounts
- Ability to provide personalized recommendations and content

5.2. Non-Functional Requirements

- The application must be available and accessible to users at all times.
- The application must be secure, protecting data from unauthorized access or modification.
- The application must provide a fast and reliable user experience.
- The application must be easy to use and understand.
- The application must be maintainable over time, with the capability to add new features and make changes with minimal disruption
- The application should be able to handle large volumes of data efficiently, with minimal latency and high throughput.
- The cost of the application should be within the budget of the organization.
- The application should be able to store large amounts of data.
- The application should have an effective backup and recovery system in place.
- The application should have a well-defined maintenance plan to ensure smooth functioning.

6. Testing

The project included testing to ensure that the application works as expected. Testing included both manual and automated testing. Automated testing should include both unit and integration tests.

As part of testing, we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance. Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies, a test plan is carried out on each module. The various tests performed are unit testing, integration testing, and user acceptance testing.

7. Conclusion

The development of this supportive application marks the successful completion of the project. With the help of this application, we have been able to provide an effective and efficient support system to our customers in order to improve their experience with our products and services. The application is easy to use and provides several useful features such as automated ticketing, customer support chat, FAQs, and analytics. The development of this application has helped us to address customer inquiries more quickly and effectively. It has also enabled us to keep track of customer queries and use the data to improve our products and services. We are confident that the application will continue to be a valuable asset for our organization in the future. We are sure that the application will help us to provide better customer service and improve customer satisfaction.

8. Reference

- https://www.camcode.com/blog/best-work-order-management-software-systems/