Project Report Format

1. INTRODUCTION

2. 1.1 Project Overview
   1.2 Purpose

3. LITERATURE SURVEY
4. 2.1 Existing problem
   2.2 References
   2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION
4. 3.1 Empathy Map Canvas
   3.2 Ideation & Brainstorming
   3.3 Proposed Solution
   3.4 Problem Solution fit

5. REQUIREMENT ANALYSIS
6. 4.1 Functional requirement
   4.2 Non-Functional requirements

5. PROJECT DESIGN

6. 5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
5.3 User Stories

7. PROJECT PLANNING & SCHEDULING

8. 6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule
6.3 Reports from JIRA

7:CODING & SOLUTIONING (Explain the features added in the project along with code)

7. 7.1 Feature 1
7.2 Feature 2
7.3 Database Schema (if Applicable)

8. TESTING

9. 8.1 Test Cases
8.2 User Acceptance Testing

9. RESULTS

10. 9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX

Source Code

GitHub & Project Demo Link

1INTRODUCTION

1 1.1 Project Overview

This Application has been developed to help the customer in processing their complaints. The customers can raise the ticket with a detailed description of the issue. An Agent will be assigned to the Customer to solve the problem. Whenever the agent is assigned to a customer they will be notified with an email alert. Customers can view the status of the ticket till the service is provided.

Admin : The main role and responsibility of the admin are to take care of the whole process. Starting from Admin login followed by the agent creation and assigning the customer's complaints. Finally, He will be able to track the work assigned to the agent and a notification will be sent to the customer.

User: They can register for an account. After the login, they can create the complaint with a description of the problem they are facing. Each user will be assigned with an agent. They can view the status of their complaint.

1.2 Purpose

Customer services is the support you offer your customers-that helps them have an easy,enjoyable experience with your brand

2. LITERATURE SURVEY

LITERATURE SURVEY

When done well, customer care registry boosts the overall customer Experience by providing answers to common questions through the website, Social media, or with customer support agents. Companies benefit from investing

Customer care

- Customer satisfaction can increase and customer loyalty can improve
- Customer service agents spend less time on routine tasks
- Answering commonly asked question,enabling agent to do more meaningful tasks.

SURVEY 1 :

Customer expectations are extremely high, putting increased pressure on

Companies to improve their customer relationship. According to forrester,

Only 18% of customers said they would continue doing business with protect that

Has disappointed them. Poor customer care is costly. But they are costly to run and

Can have a high rate of employee turnover,social media drawback.

SURVEY 2 :

To properly manage customer care, companies must understand how they are

Succeeding and what needs improvement. This requires performance can also

Provide insights into what is causing a breakdown in customer retention.

Which measures customer satisfaction.using like tools machine learning ML

Companies quickly identified the problem.

SURVEY 3 :

Customer-centric business are the need of the hour. And customer -centric business can

Only be created by adapting effective customer care polices .this can involve a number

Tasks are achieved enabled application.improving remediation function process

This not only removes inconsistency in the way of tickets are resolved. Improve

Productivity , now what to do exactly agent.

2.1 Existing problem

2.2 References

Literature Survey on Customer Care Registry						
Serial Number	Journal Name	Author Name	Year	Technology Used	Existing System	Proposed System
1	A Proposed Cloud Based Solutionfor Customer Satisfaction in Telecommunication Industry	Nurulhuda Mustafa ,Lew Sook Ling, S ti Fatimah Abdul Razak	2019	Cloud based framework, Data Analytics	In existing cloud based solution framework, user found it difficult to communicate with customer service representative during faulty experience, and follows traditional way of acquiring and managing datasor information.	A proposed cloud-based customer supports solution for telecommunication industry. The proposed enhancements are as follows: Mutual agreement between customer and company during making restoration appointment, Real time and status tracking enabled, Enhance customer trust by gettings signature using apps to confirm job done, Job done summary, Introduce loyalty program such as variety of vouchers are given for redemption using accumulated points by customers.
2	Using SMS and Web Technology in Mobile Government Information Services Platform	Hui Zhang ,Fayu Wang	2010	Ajax web technology,multi threat technology	In existing system traditional electronic system, it usually employs the wired network communication or handles on the spot. In case, the government servants leave the office, they can not obtain the timely information of the government and related departments which causes the delay of decision-making and lack of information. Moreover, the low penetration of the computer terminal restricts people's receive of the government information. With development of the mobile communication technology, especially the roaring increasing of mobile phone users, a kind of mobile wireless administration based on the mobile network platform has emerged. It is taken seriously by many municipalities and regarded asthe promoter to build an efficient andtransparent government.	A proposed SMS Technology in mobile service platform is a new kind electronic office platform which collates the traditional electronic administration and mobile communication. It collects the public proposals and advices, tracks handling of complaints and checks all kinds of information by means of mobile phone messages and websites. With the constant progress of mobile communication technology and coming of 3G, it will accelerate the development of our country'smobile information further. More and more companies will establish their own message platform, thus the exploitation andapplication of message platform makes great sense in society and reality. The research based on the short message mobile administration platform brings aboutan efficient, friendly, people-oriented government administration mode, which bridges the government and the public and meets the inner requirement of building a harmonious socialist society.
3	Real World Smart Chatbot for Customer Care using a Software asa Service (SaaS) Architecture	Godson Michael D'silva, Sanket Thakare,Shraddha More and Jeril Kuriakose	2017	Ejabberd,AWS Lambda,Machine Learning, LUIS,Chatbot,API Gateway, Cognitive Services.	As many customers may be using this streams to reach out to companybecause they need help. The company have setup social marketing team to monitor this stream. But due to huge volumes of users it's very difficult to analyses each and every social message and take a relevant action to solve users grievances, which lead to many unsatisfied customers or may even lose a customer. This papers proposes a system architecture which will try to overcome the above shortcoming by analyzing messages of each ejabberd users to check whether it's actionable or not. If it's actionable then an automated Chatbot will initiates conversation with that user and help the user to resolve the issue by providing a human way interactions using LUIS and cognitive services.	A proposed Real World Smart Chatbot system architecture focus on analyzing this social chats by identifying whether the messages from the customers are actionable or not. All the actionable messages are send to the Chatbot which tries to resolve the issues faced by the user by initiating the conversation with the customer in a more human way. This save lots of money and resources of the companyused for customer service and even making the customer more and satisfied. As this proposed system is implemented on AWS public cloud, it make this system capable of handling enormous amount of user base.

Literature Survey on Customer Care Registry						
Serial Number	Journal Name	Author Name	Year	Technology Used	Existing System	Proposed System
4	Virtual Customer Service Agents: Using Social Presence and Personalization to Shape Online Service Encounter	Tibert Verhagen, Jaap van Nes, Frans Feldberg, Willemijn van Dolen, Ph.D	2014	Data Analysis	In Existing system, we empirically investigate the role of VCSAs to shape more social and personalized online service encounters. Empirical studies on VCSAs are scarce and openly demanded, and a focus on the ability of VCSAs to provide service encounters with a human touch deals with conventional wisdom that social and personal approaches are critical to customer service delivery. Within this inquiry we address the direct influence of VCSA characteristics on online customer service evaluations and are among the first to extrapolate whether employing cues deemed important in traditional service encounter literature. This enables us to evaluate the cross-channel applicability of traditional customer service thought and provide further directions to the academic field of online customer service. It also reduces their effort, time, and cost to design, implement, and maintain such an agent as well as to shape the service process	In proposed system, First, to provide theoretical foundations for the employment of VCSAs, we encourage researchers to experiment with more technically advanced agents that will appear in the near future. By adding and combining elements such as motion, natural speech, lip synchronization, and 3D representation to virtual agent design, new insights into the value of mimicking humanlike service personnel online is gained. Second, more in-depth research on the role of emotions in VCSA settings is encouraged. While we did not find any effect of smiling, VCSAs may still express (positive) emotions that contribute to more positive customer evaluations of the service encounter. An interesting area of future research would be to examine whether affective real-time interactive facial expressions, and more emotional communication styles would influence the sociali ness and personalization perception of the agent.
5	Online Complaint Registration System to Municipality	A.Prassana, Dr. A.V. Senthil Kumar	2020	Android Studio, Java	In existing system, CMS (Complaint Management System) is used. Manual systems put pressure on people to be correct in all details of their work at all times, the problem being that people aren't perfect, however much each of us wishes we were. With manual systems the level of service is dependent on individuals and this puts a requirement on management to run training continuously for staff to keep them motivated and to ensure they are following the correct procedures. It can be all to easy to accidentally switch details and end up with inconsistency in data entry or in hand written orders. This has the effect of not only causing problems with customer service but also making information unable be used for reporting or finding trends with data discovery.	In proposed syystem, by using android application people can register their complaints in easy and proper format. Mainly they can mark their location in Google Map while plading the complaint so that it will help the people in easy manner. They will also well aware about their complaints progress. They can also provide feedback about their complaints progress weather they are satisfied or not. Also they user can post their requirements through this system and they will receive needed items by admin within couple of hours ,its depending on the needed item and you can also look your status about your requirements. These user complaints, needs requirements maintain by admin. The User post feedback of these CMS system and admin can view this feedback.
6	Implementation Of 'ASR4CRM': An Automated Speech Enabled Customer Care Service System	Aderemi A. Atayero, Charles K. Ayo, Ilihu-Omoregbe Nicholas and Azeta Ambrose	2009	VoiceXML, PHP and Apache, MySQL	The main disadvantage of existing system is the human presence in the Call-centers of GSM service providers is poor response time.	The proposed system describes the implementation of ASR4CRM - an automated customer care service system that obviates the need for a human operator, reduces the budget allocation of corporate bodies for CCS and most importantly, improves the business to customer (B2C) relationship, which is often damaged by inevitable flaws in the human character.

Literature Survey on Customer Care Registry						
Serial Number	Journal Name	Author Name	Year	Technology Used	Existing System	Proposed System
7	A Blockchain and AutoML Approach for Open and Automated Customer Service	Zhi Li, Hanyang Guo, Wai Ming Wang, Yijiang Guan, Ali Vatankhah Barenji, George Q. Huang, Kevin S. McFall, and Xin Chen	2019	Blockchain, AutoML, IoT	In existing system,As for small and medium enterprises (SMEs), it is difficult for them to have sufficient data, in particular, in the early days of establishment. Due to the lack of data, some SMEs purchase data from third parties or crowdsourcing. They may not know the authenticity and the source of the data. A legal agreement is difficult and loose among multiple parties in crowdsourcing. It may consist of many conflicts of interest issues, including intellectual property rights. Moreover, the quality of data is difficult to assure and control. The quality of customer services is also difficult to control and ensure.	The proposed system describes the implementation of blockchain and AutoML which incorporates the open and distributed advantages of blockchain and the automation advantage of AutoML. A new concept on open and automated customer service was proposed. Compared with traditional customer service methods, the proposed platform constructs a shared and trustless environment for data trading, which is particularly helpful for SMEs to acquire sufficient data for achieving automatic customer services as well as developing their core customer service competence.
8	Using Authentic Leadership and Mindfulness as Internal Marketing Mechanism for Enhancing Proactive Customer Service Performanc	C. M. Wu ,T. J. Chen ,Y. D. Lee, T.F.Chen	2016	Data analysis, confirmatory factor analysis, structural equation modeling	In existing system, internal marketing is critical for enhancing superior service quality in service marketing domain. Internal marketing is also critical for providing superior service and promoting external marketing success. Even the hospitality supervisors' leadership behavior is critical in influencing employee behavior and work performance. With the more and more complex and competitive hotel external business environment, the issue of how to satisfy consumer needed and not necessarily demand have become critical for hotel to establish competitive advantage.	The proposed model integrates authentic leadership, mindfulness, and proactive customer service performance. According to the analysis, as expected, authentic leadership can positively influence mindfulness, authentic leadership and mindfulness can be used in internal marketing for promoting employees proactive customer service performance. Specifically, we highlight that mindfulness also has a partially mediating role between authentic leadership and proactive customer service performance.
9	An Application of SMS Technology for Customer Service Centre	Ariff Idris, Abd. Samad Hasan Basari, Nur Hanisah Zubir,	2009	Smart Message System Technology, PHP, MySQL	In existing system, LAP is a semi-government organization in Perak which is responsible in managing the water supply service and distribution for Perak citizens. However LAP has only had a hotline number for their customers to make a complaint. The existing method of handling customers' complaint is delaying the action taken.	The proposed system Ces-LAP allow LAP customer to make complaints easier. The proposed system is very much help when there are many complaints at one time. This system can be used by everyone that have accessed to internet and hand phone. Furthermore the system helps LAP to manage all the complaints faster and effective via SMS and web. The prototype of the system is under testing phase. An initial feedback from users shows that the system is quite good in term of its mobility.

Literature Survey on Customer Care Registry						
Serial Number	Journal Name	Author Name	Year	Technology Used	Existing System	Proposed System
10	Online Helpdesk Support System for Handling Complaints and Service	Cadelina Cassandra , Sugiarto Hartono , Marisa Karsen	2019	Web Technology	In existing system, The customer's complaint often not documented because the customer services record all complaint manually one by one and the amount of complaint increase day by day. The customer service often answers the same question rom different customer. There is no information for the customer about the progress of the complaint and it is difficult to monitor the complaint and report.	In proposed system, by Using Online helpdesk, customer can submit complaint in the web. The customer services will find a notification for new complaint in the system. There is a Frequently Ask Question in the system, so customer can find the common answers and solve the problem by themselves. The customer service also can solve same problems quickly because all history of complaint and solution is recorded. The customer can find the notification for the progress of complaint by status in the system. With this integrated system between front end and back end system, all of the complaint history will be recorded easily, so manager can monitor everything. Thus the utilization of helpdesk system can reduce task, problem, and improve the customer satisfaction rather than waiting for the customer service to solve the problem. The benefit of this online helpdesk system for this service company will reduce complexity of customer complaints and needs using one system, and for the impact, it will increase customer satisfaction.

2.3 Problem Statement Definition

3

Customer Problem Statement:

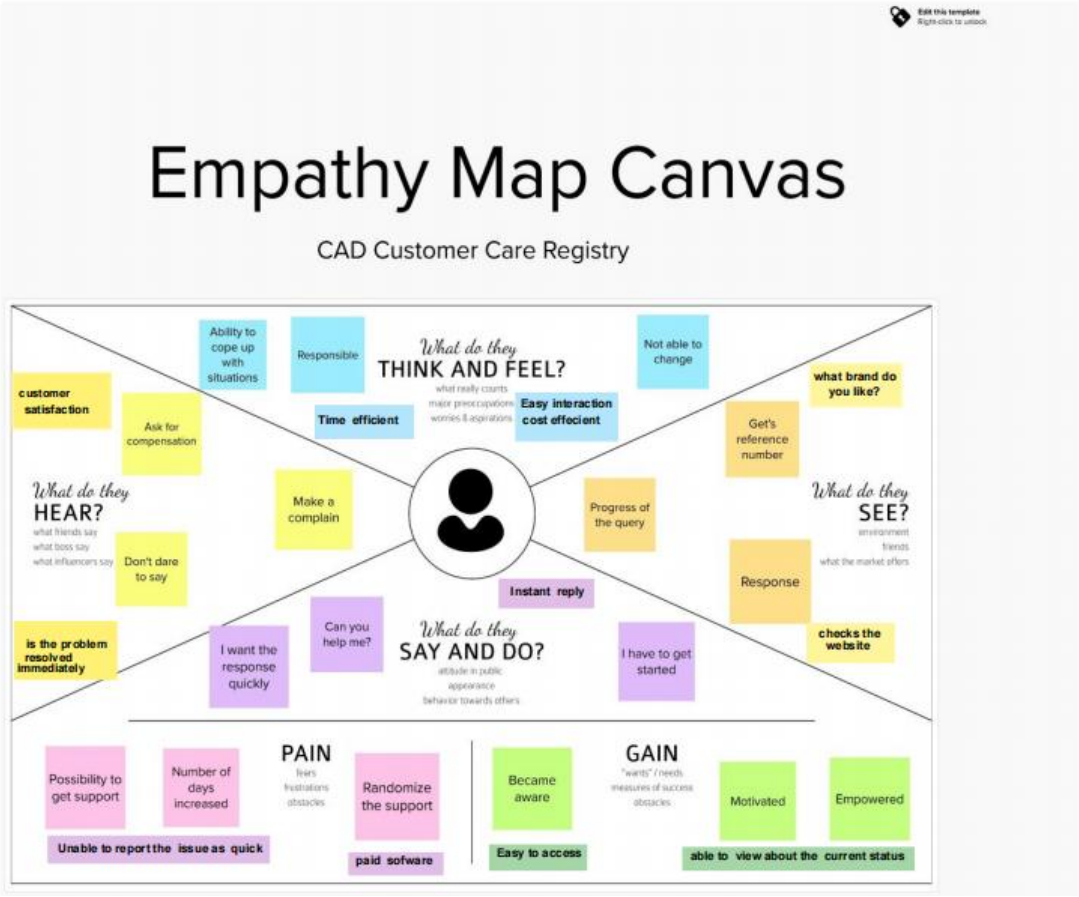
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	In need of help	Find assistance to fix my issue about issue	I can't find the right people who can help me	Too much complexity in finding the right solution	Frustrated and helpless
PS-2	Having language issues	Explain my problem to the customer care registry	They can't seem to understand my problem	Of Language barriers	Irritated
PS-3	Having a hard time explaining the issue	Conceptualize and convey my issue	I can't seem to find the right way to explain it	Of the complexity and the niche domain of the issue	Hopeless
PS-4	Satisfied with the assistance and want to convey my thanks	Give my review about the interaction which happened with my agent	The chat window gets closed after resolution of the issue at hand	The issue got resolved	customer expectation customer satisfaction

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming

Brainstorm & Idea Prioritization :

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping1Brainstorm priorizationWrite down any ideas that come to mind that addressyour rules problem statement2Group ideasTakes turns sharing your ideas while clustering similar or related noteskanimozhi user feedback customer privacy solution tocustomer askng for Ratingkaviyacustomer satisfactiontracking serviceallocating agentagents detail email notificationcustomer queries providing service.1. 2.listen carefully to queries live chat**customer Expectation**Customer QueriesCustomer noifytingproviding service on timeAsking for Rating**security** security solution to customer **Feedback**User feedback customer privacy customer satisfactionAsking Ratingproviding service detail

3.3 Proposed Solution

Proposed Solution :

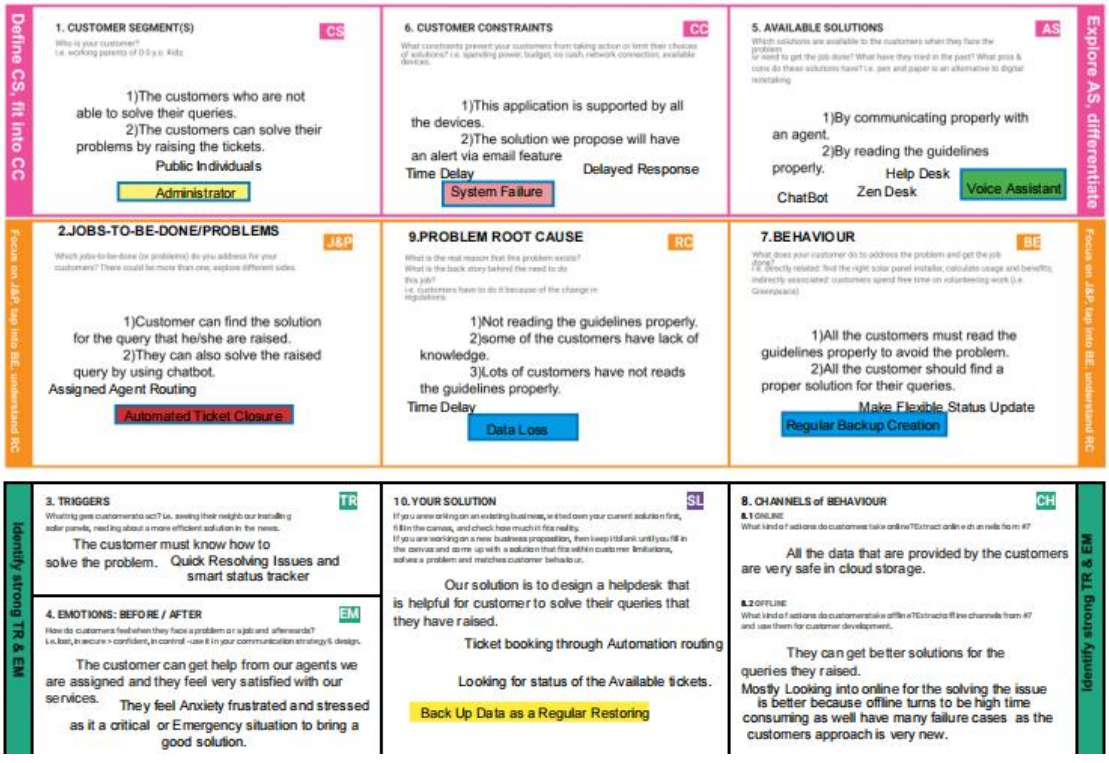
Project team shall fill the following information in proposed solution

S.no	Parameter	Description

1	Problem statement(solved)	To solved the customer issues help of the website use application
2	Proposed solution (idea)	Once the customer raise the ticket then admin assigned agent Routing can be solved By directly route to the specific agent using email,status shown to the customer can display the status of the ticket
3	Specific features	Assigned agent Routing automatic ticket status shown to the customer .backup data failure
4	Customer satisfaction Impact of social service	Customer satisfaction Customer track the status and easy agent communication.then understand the issue customer satisfied Check of status of current on time Whole process can see the customer's status Communication Service are customer satisfaction

5	Business Model	Key reasources, knowleage based channel cloud platform services ,offices
6	Scalablity	An environment where they will be able to spend less time on grunt work More time required solving customer issues, the scaling customer service is efficient as possible

3.4 Problem Solution fit



4 REQUIREMENT ANALYSIS

4:3Functional Requirements :

Following are the functional requirements of the proposed solution.

FR No	Functional Requirements	Sub Requirement (sub task)
FR-1	User registration	Registration Form Registration link Registration gmail Register valid mobile number
FR-2	User confirmation	Confirmation via Email Confirmation via OTP Two step authentication security

Functional Requirements :

Following are the functional requirements of the proposed solution.

FR No	Functional Requirements	Sub Requirement (sub task)
FR-1	User registration	Registration Form Registration link Registration gmail Register valid mobile number
FR-2	User confirmation	Confirmation via Email Confirmation via OTP Two step

		authentication security

4.2 Non-Functional requirements

Non- Functional Requirements:

Following are the non- functional requirements of the proposed solution

FR no	Non functional	Description
NFR -1	Usablity	To provide the solution to the problem for user they can register with unique vaild email id Also, made our web application flexible
NFR -2	Reliability	Tracking of decade status to the Email.since we had split the agent into categories, system response time for each every individual will be lesser.
NFR -3	performance	Effective Development web in order to

		bring best performance ,we have concerntrated on over the user request.so every individual user will be allotted with individual agents.
NFR -4	scalability	Agents scalability as per the number of customers, with respect to increase in user request ,allotment will be increased, recalling is always adaptable
NFR -5	security	Track of login verification way of authentication before any user trying to login their account to any new device ,only after entering their code they will be

		allowed to login verify code is also made expire within particular time limit
NFR -6	Availability	24 /7 service is available on the 24 hours per 7 days user can interact with their respective agents 24*7 following proper user-agent guildelines

5:PROJECT DESIGN

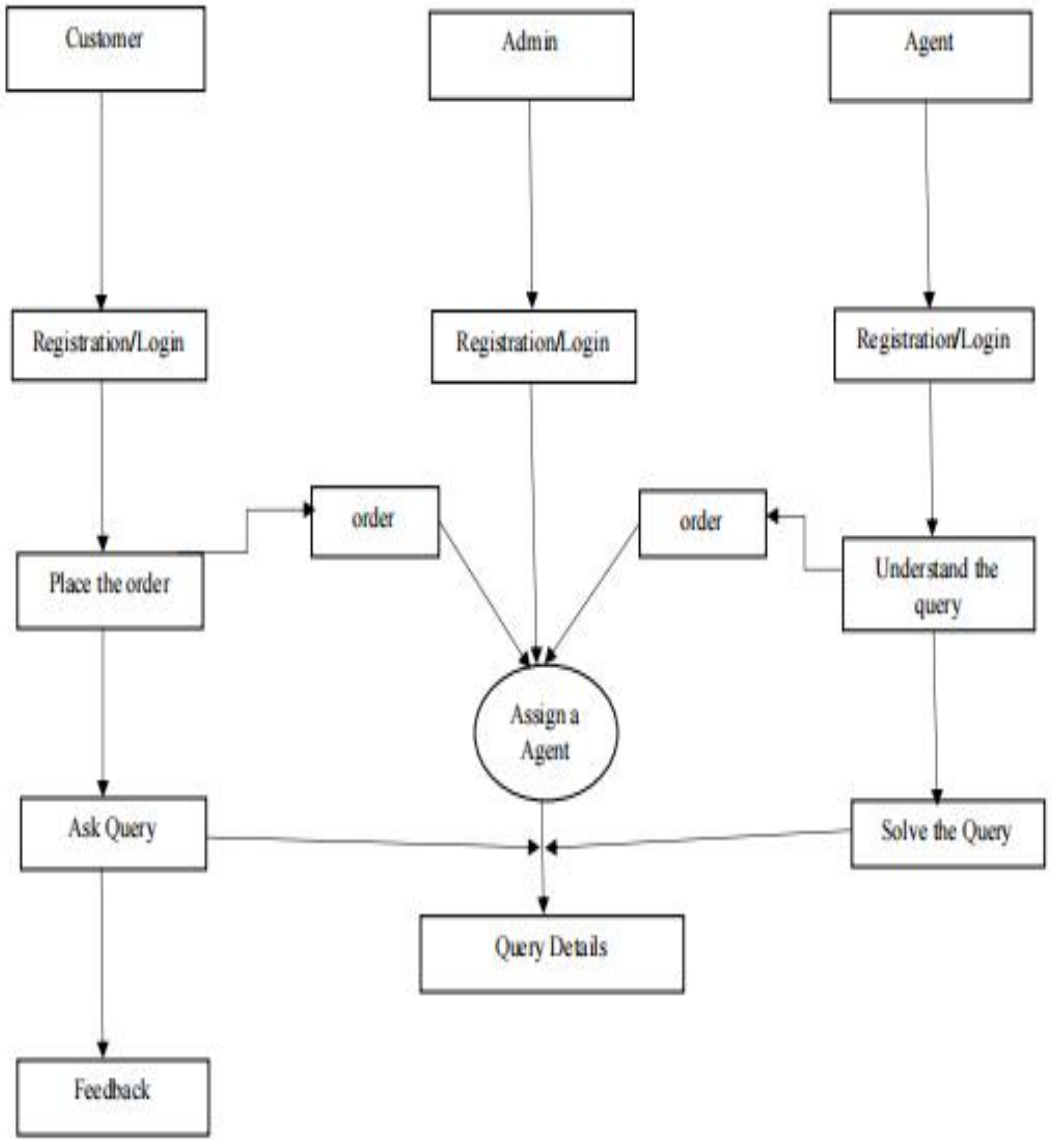
5:1Data Flow Diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the
Right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored

User Stories:

A User Stories is the about the product using after
Experience of the table of the story always elaborates
An advantage for the user, customer or user

Data Flow Diagram:



6:PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story**	**Story Points**	**Priority**	**Team Members**
Sprint -1	Customer (Web User)	Registration	USN -1	As a customer, I can register for the application by entering my email, password, and confirming my password.	2	High	Kamalasri Kanimozhi Kaviya Kirubanithi
Sprint -1		Login	USN -2	As a customer, I can login to the application by entering correct email and password	1	High	Kamalasri Kanimozhi Kaviya Kirubanithi
Sprint -1		Dashboard	USN -3	As a customer, I can see all the tickets raised by me and lot more	3	High	Kamalasri Kanimozhi Kaviya kirubanithi
Sprint-2		Ticket creation	USN -4	As a customer, I can create a new ticket with the detailed description of my query	2	High	Kamalasri Kanimozhi Kirubanithi
Sprint -3		Address Column	USN -5	As a customer, I can have conversations with the assigned agent and get my queries clarified	3	High	Kamalasri Kaviya kirubanithi

Sprint -3		Forgot password	USN -6	As a customer, I can reset my password by this option in case I forgot my old password	2	Mediu m	Kamalasri Kaviya Kirubanithi
Sprint -4		Ticket details	USN-7	As a customer, I can see the current status of my tickets	2	Mediu m	Kamalasri Kanimozhi Kaviya Kirubanithi
Sprint -3	Agent (Web user)	Login	USN-1	As an agent, I can login to the application by entering correct email and password	2	High	Kamalasri Kaviya Kirubanithi
Sprint -3		Dashboa rd	USN-2	As an agent, I can see all the tickets assigned to me by the admin	3	High	Kamalasri Kaviya Kirubanithi
Sprint -3		Address Column	USN-3	As an agent, I get to have conversations with the customer and clear his/her queries	3	High	Kamalasri Kaviya Kirubanithi
Sprint -4		Forgot password	USN-4	As an agent, I can reset my password by this option in case I forgot my old password	2	Mediu m	Kamalasri Kanimozhi Kaviya Kirubanithi
Sprint -1	Admin (Web user)	Login	USN-1	As an admin, I can login to the application by entering correct email and password	1	High	Kamalasri Kanimozhi Kaviya Kirubanithi
Sprint -1		Dashboa rd	USN-2	As an admin, I can see all the tickets raised in the entire system and lot more	3	High	Kamalasri Kanimozhi Kaviya Kirubanithi
Sprint -2		Agent creation	USN-3	As an admin, I can create an agent for clarifying the customer's queries	2	High	Kamalasri Kanimozhi Kirubanithi
Sprint		Assigning	USN-	As an admin, I	3	High	Kamalasri

-2		agent	4	can assign an agent for each ticket created by the customer			Kanimozhi Kirubanithi
Sprint -4		Forgot password	USN-4	As an admin, I can reset my password by this option in case I forgot my old password	2	Mediu m	Kamalasri Kanimozhi Kaviya Kirubanithi

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart :(4marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End (Planned) Date	Story Points Completed (as on planned End date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 oct 2022	29 oct 2022	10	29 oct 2022
Sprint-2	7	6 Days	31 oct 2022	5 Nov 2022	7	5 Nov 2022
Sprint-3	11	6 Days	7 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	8	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

Velocity:

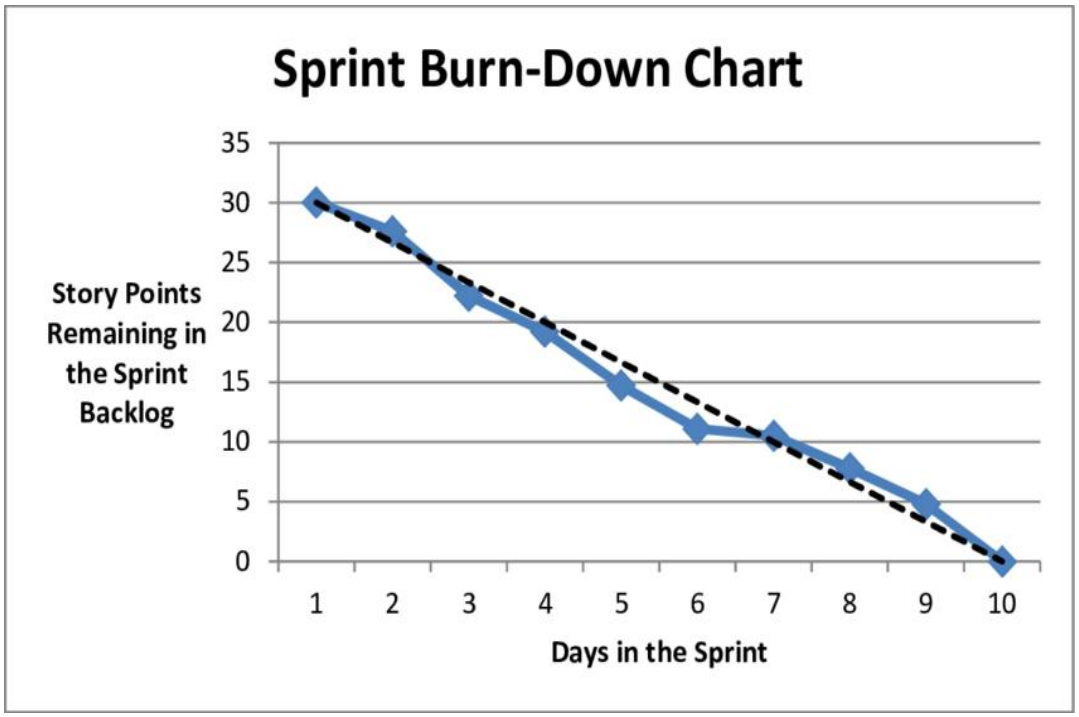
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used.in agile software development methodologies such as scrum.

6.3 Reports from JIRA



7:CODING & SOLUTIONING (Explain the features added in the project along with code)

```
#!venv/bin/python3
import random
import string
from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db


def Upper_Lower_string(length):
   result = ''.join((random.choice(string.ascii_uppercase)
                     for x in range(length)))
   return result


conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURI
TY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dml49039;PWD=oviOR6wj
SnNPcRC2", '', '')
app = Flask(__name__)
app.config['SECRET_KEY'] = 'helloworld'
```

```
@app.route('/')
def home():
    return render_template('index.html')


@app.route('/register', methods=['POST', 'GET'])
def register():
    if request.method == "POST":
        global rs
        name = request.form.get('name')
        email = request.form.get('email')
        password = request.form.get('password')
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.execute(stmt)
        rs = ibm_db.fetch_assoc(stmt)
        print(rs)
        if rs:
            msg = 'Account already Exists'
            return render_template('register.html', msg=msg)
        else:
            reg_stmt = ibm_db.prepare(
                conn, 'INSERT INTO user ("USERNAME","EMAIL","PASSWORD") VALUES(?,?,?)')
            ibm_db.bind_param(reg_stmt, 1, name)
            ibm_db.bind_param(reg_stmt, 2, email)
            ibm_db.bind_param(reg_stmt, 3, password)
            ibm_db.execute(reg_stmt)
            msg = 'Successfully Registered'
            return render_template('register.html', msg=msg)
    else:
        return render_template('register.html')


@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == "POST":
        customer = list()
        agent = list()
        name = request.form['name']
        password = request.form['password']
        log_stmt = ibm_db.prepare(
            conn, 'SELECT * FROM user WHERE username=? and password=?')
        ibm_db.bind_param(log_stmt, 1, name)
        ibm_db.bind_param(log_stmt, 2, password)
        ibm_db.execute(log_stmt)
        rs = ibm_db.fetch_assoc(log_stmt)
        if rs:
            session['role'] = 'user'
            session['customer'] = rs
            print(rs)
            return render_template('dashboard.html')
        log_stmt = ibm_db.prepare(
```

```
        conn, 'SELECT * FROM agent WHERE username=? and password=?')
    ibm_db.bind_param(log_stmt, 1, name)
    ibm_db.bind_param(log_stmt, 2, password)
    ibm_db.execute(log_stmt)
    rs = ibm_db.fetch_assoc(log_stmt)
    if rs:
        cms = ibm_db.exec_immediate(conn, 'SELECT * FROM user')
        agt = ibm_db.exec_immediate(conn, 'SELECT * FROM agent')
        customers = ibm_db.fetch_assoc(cms)
        agents = ibm_db.fetch_assoc(agt)
        while customers:
            customer.append(customers)
            customers = ibm_db.fetch_assoc(cms)
        while agents:
            agent.append(agents)
            agents = ibm_db.fetch_assoc(agt)
        print(customer)
        print(agent)
        session['role'] = 'agent'
        session['name'] = rs['USERNAME']
        session['customer'] = customer
        session['agent'] = agent
        return render_template('dashboard.html')
    log_stmt = ibm_db.prepare(
        conn, 'SELECT * FROM admin WHERE username=? and password=?')
    ibm_db.bind_param(log_stmt, 1, name)
    ibm_db.bind_param(log_stmt, 2, password)
    ibm_db.execute(log_stmt)
    rs = ibm_db.fetch_assoc(log_stmt)
    if rs:
        cms = ibm_db.exec_immediate(conn, 'SELECT * FROM user')
        agt = ibm_db.exec_immediate(conn, 'SELECT * FROM agent')
        customers = ibm_db.fetch_assoc(cms)
        agents = ibm_db.fetch_assoc(agt)
        while customers:
            customer.append(customers)
            customers = ibm_db.fetch_assoc(cms)
        while agents:
            agent.append(agents)
            agents = ibm_db.fetch_assoc(agt)
        print(customer)
        print(agent)
        session['role'] = 'admin'
        session['customer'] = customer
        session['agent'] = agent

        return render_template('dashboard.html', agent=agent, customer=customer)
    else:
        msg = 'UID/Password is incorrect'
        return render_template('login.html', msg=msg)
else:
    return render_template('login.html')
```

```
@app.route('/dashboard', methods=['POST', 'GET'])
def dashboard():
    return render_template('dashboard.html')


@app.route('/success', methods=['POST', 'GET'])
def success():
    if request.method == "POST":
        ticket = session['ticket'] = Upper_Lower_string(16)
        print(ticket, session['ticket'])
        query = request.form['query']
        sql = "UPDATE user SET QUERY=?,TICKET=? WHERE USERNAME=?"
        out = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(out, 1, query)
        ibm_db.bind_param(out, 2, session['ticket'])
        ibm_db.bind_param(out, 3, session['name'])
        status = ibm_db.execute(out)
        if status:
            msg = 'Success ! Your Ticket Nno is :', ticket, 'You can now return to the home page'
            return render_template('success.html', msg=msg)
        else:
            msg = 'Error Submitting your Query'
            return render_template('success.html', msg=msg)


@app.route('/redirect')
def redir():
    return redirect(url_for('home'))


@app.route('/querying', methods=['POST'])
def admin_query():
    msg = ""
    agent = request.form.getlist('agent_name')
    usr_name = request.form.getlist('cus_name')
    for i in range(0, len(agent)):
        if agent[i] != 'none':
            qr = ibm_db.prepare(
                conn, "UPDATE USER SET ASSIGNED_AGENT=? WHERE USERNAME=?")
            ibm_db.bind_param(qr, 1, agent[i])
            ibm_db.bind_param(qr, 2, usr_name[i])
            result = ibm_db.execute(qr)
            print(agent[i], "    ", usr_name[i])
            if result:
                msg = '<h1>queries executed</h1>'
    return render_template('done.html', msg=msg)


@app.route('/executing...', methods=['POST', 'GET'])
def agent_submit_reply():
    names = request.form.getlist('name')
    text = request.form.getlist('text')
    print(names)
    print(text)
```

```
    for i in range(0, len(names)):
        if not text[i] == '':
            try:
                sql = 'UPDATE USER SET REPLY=?,REVIEW_STATUS=1 WHERE USERNAME=?'
                query = ibm_db.prepare(conn, sql)
                ibm_db.bind_param(query, 1, text[i])
                ibm_db.bind_param(query, 2, names[i])
                ibm_db.execute(query)
            except:
                print('an error occured')
    return '<html><body>done</body></html>'


if __name__ == '__main__':
    app.run(debug=True)
```

10. TESTING

11. 8.1 Test Cases

Test Cases:

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Pass / Fail
15.	Customer creating a new ticket with empty query	1. Go to site 2. Customer login using email and password 3. Click "New Ticket" option in the Dashboard 4. Clicking the "New Ticket" button without typing any query in the given text area	Query = NULL	Customer should get an alert saying "Query cannot be empty!"	As expected	Pass
16.	Customer creating a new ticket with a valid query	1. Go to site 2. Customer login using email and password 3. Click "New Ticket" option in the Dashboard 4. Typing the query in the given text area 5. Clicking the "New Ticket" button	Query = "Hi. My I Phone 14 pro max is not turning on. It is a new unit I bought it just 2 days back. I don't know what happened. Can you help me please?"	The ticket gets inserted in the database. After that customer gets an alert saying 'Ticket created'	As expected	Pass

17.	Customer seeing all the tickets raised by him/her	1. Go to site 2. Customer login using email and password 3. Click "Tickets" option in the Dashboard	Tickets created by the customer which are already being inserted in the database	Customer should see the list of all the tickets raised by him/her	As expected	Pass
18.	Customer seeing all the tickets raised by him/her	1. Go to site 2. Customer login using email and password 3. Click "Tickets" option in the Dashboard	-	Customer should see a message "You are yet to raise a ticket"	As expected	Pass
19.	Customer seeing the query of a ticket	1. Go to site 2. Customer login using email and password 3. Click "Tickets" option in the Dashboard 4. Click "View" option in a ticket from the list of tickets	Tickets created by the customer which are already being inserted in the database	An alert should be shown having the actual query posted by the customer	As expected	Pass
20.	Customer seeing the assigned agent for a ticket	1. Go to site 2. Customer login using email and password 3. Click "Tickets" option in the Dashboard	• Tickets created by the customer which are already being inserted in the database • Admin assigned the agent for the ticket	Customer should be able to see the first name of the agent assigned	As expected	Pass
21.	Customer seeing the assigned agent for a ticket	1. Go to site 2. Customer login using email and password 3. Click "Tickets" option in the Dashboard	• Tickets created by the customer which are already being inserted in the database • Admin is yet to assign the agent	Customer should be able to see the "N/A" message displayed	As expected	Pass
22.	Admin seeing all the unassigned tickets	1. Go to site 2. Admin login using email and password 3. Click "Tickets" option in the Dashboard	• Tickets created by the customers which are already being inserted in the database • Admin did not assign agent for the tickets	Showing the tickets that are yet to be assigned an agent by the admin	As expected	Pass
23.	Admin seeing all the unassigned tickets	1. Go to site 2. Admin login using email and password 3. Click "Tickets" option in the Dashboard	• Tickets created by the customers which are already being inserted in the database • Admin assigned agents for all the tickets	Admin should just see the message "There is nothing left to assign"	As expected	Pass
24.	Admin assigning an agent for a ticket	1. Go to site 2. Admin login using email and password 3. Click "Tickets" option in the Dashboard 4. Select an agent from the dropdown given	• Tickets created by the customers which are already being inserted in the database • Admin did not assign the agent yet	Admin should get an alert saying "Do you really want to assign the agent for this ticket?". If admin clicks OK, then the agent is assigned for the ticket. The list gets updated	As expected	Pass
25.	Admin seeing the requests section	1. Go to site 2. Admin login using email and password 3. Click "Requests" option in the Dashboard	• Agent details in the database • Admin is yet to accept the agent	Admin should be able to see the list of all the requests made by the agents to the admin	As expected	Pass

8.2 User Acceptance
Testin

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the **Customer Care Registry** project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	0	0	2	7
External	0	2	0	0	2
Fixed	12	11	35	45	103
Not Reproduced	0	5	0	0	5
Skipped	0	0	0	0	0
Totals	17	18	35	47	117

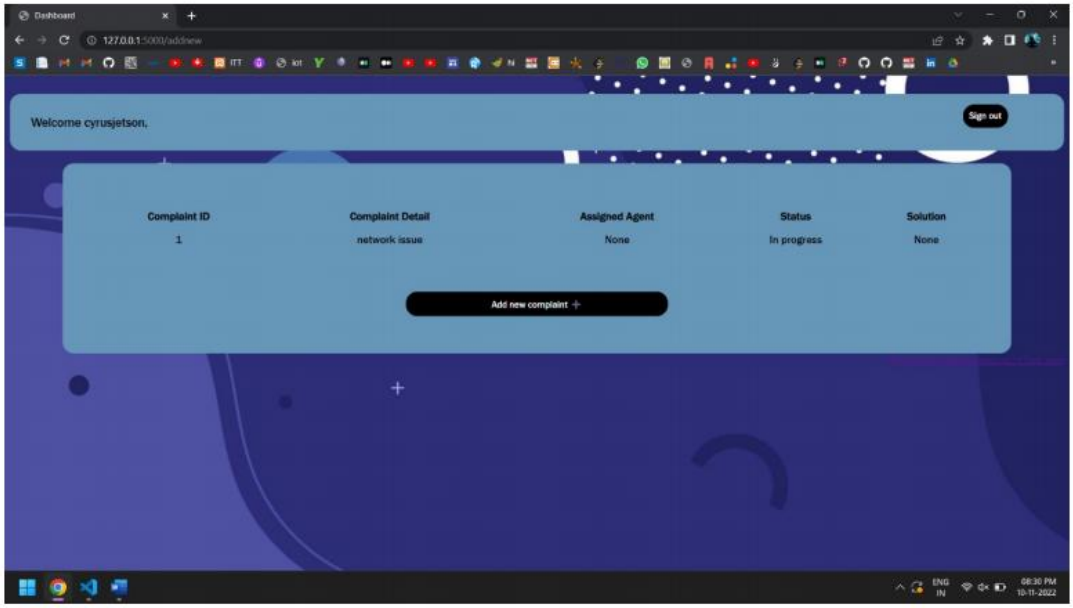
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Client Application	72	0	0	72
Security	7	0	0	7
Exception Reporting	5	0	0	5
Final Report Output	4	0	0	4

11. RESULTS

12. 9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

11. Brainstorm priorizationWrite down any ideas that come to mind that addressyour rules problem statement2Group ideasTakes turns sharing your ideas while clustering similar or related noteskanimozhi user feedback customer privacy solution tocustomer askng for Ratingkaviyacustomer satisfactiontracking serviceallocating agentagents detail email notificationcustomer queries providing service.1. 2.listen carefully to querieslive chat**customer Expectation**Customer QueriesCustomer noifytingproviding service on timeAsking for Rating**security** security solution to customer **Feedback**User feedback customer privacy customer satisfactionAsking Ratingproviding service detail

11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX

Source Code
GitHub & Project Demo Link