

# CUSTOMER CARE REGISTRY

**TEAM ID: PNT2022TMID47962**

## **TEAM MEMBERS:**

<b>Pandiselvi R (TEAM LEAD)</b>	<b>912219104023</b>
<b>Chithra banu M</b>	<b>912219104005</b>
<b>Fathimakani M</b>	<b>912219104011</b>
<b>Raarajeshwari S</b>	<b>912219104025</b>

<b>S.NO</b>	<b>Table of content</b>	<b>PAGE NO.</b>
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Project Overview	3
	1.2 Purpose	3
<b>2</b>	<b>LITERATURE SURVEY</b>	
	2.1 Existing problem	5
	2.2 References	5
	2.3 Problem Statement Definition	7
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	
	3.1 Empathy Map Canvas	8
	3.2 Ideation & Brainstorming	9
	3.3 Proposed Solution	11

	3.4 Problem Solution fit	13
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	
	4.1 Functional requirement	14
	4.2 Non-Functional requirements	14
<b>5</b>	<b>PROJECT DESIGN</b>	
	5.1 Data Flow Diagrams	15
	5.2 Solution & Technical Architecture	15
	5.3 User Stories	16
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	
	6.1 Sprint Planning & Estimation	17
	6.2 Sprint Delivery Schedule	17
	6.3 Reports from JIRA	17
<b>7</b>	<b>CODING &amp; SOLUTIONING</b>	
	7.1 Feature 1	19
	7.2 Feature 2	21
	7.3 Database Schema (if Applicable)	22
<b>8</b>	<b>TESTING</b>	
	8.1 Test Cases	24
	8.2 User Acceptance Testing	24
<b>9</b>	<b>RESULTS</b>	
	9.1 Performance Metrics	26
<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	28
<b>11</b>	<b>CONCLUSION</b>	29
<b>12</b>	<b>FUTURE SCOPE</b>	30
<b>13</b>	<b>APPENDIX</b> (source code ,GitHub and Demo link)	32

- PROJECT OVERVIEW
- 
- INTRODUCTION

Customer care and customer service together help create a positive customer experience, or the overall impression a person has when interacting with your company. Both are vital, but there are subtle differences in how they are implemented. High-quality customer care is proactive. The needs of customers throughout the buyer's journey are anticipated, making customers feel supported. That, in turn, helps create an emotional connection between the customer and the

company. Customer service is reactive. Here, the focus is on helping customers solve problems or answer questions before purchase, either in a self-serve fashion or via the customer support team. Customer care is more than just providing great customer service. It's a proactive approach to providing information, tools, and services to customers at each point they interact with a brand. If a company neglects customer care, it can negatively impact the customer service experience. For example, when a website chatbot can't provide key information about a product, customers are more likely to get frustrated and reach out to a customer service agent for help. Consumer expectations are extremely high, putting increased pressure on companies to improve their customer relationships. This can lead to lost information when the same person reaches out via multiple channels. When a customer service agent doesn't know the whole story and the customer has to repeatedly share the problem, it leaves both people frustrated. They can register for an account. After the login, they can create a complaint with a description of the problem they are facing. Each user will be assigned an agent. They can view the status of their complaint.

- Customers get the insights they need to make an informed purchase.
- Customer satisfaction can increase and customer loyalty can improve.
- Customer service agents spend less time on routine tasks and answering commonly asked questions, enabling agents to do more meaningful task.

- **PURPOSE**

There are two sides to customer service objectives. First, there are the goals and KPIs customer service teams attempt to achieve. Then, there's customer service resume objectives. It's important to understand the connection between the two: Writing a strong customer service resume objective starts with understanding the objectives of the field and its depth and possibilities. To provide insight into both levels of customer service objectives. The prime objective of customer service is to answer customer questions quickly and effectively, resolve issues with empathy and care, document pain points to share with internal teams, nurture relationships, and improve brand credibility. Great customer service can make people loyal to your brand, products, and services for years to come.

A strong customer service resume objective underscores your skills and experiences in contributing to customer service's overall goals and objectives.

Meeting key customer service KPIs doesn't just involve answering phones and emails. It's a whole world of solutions development, intuition, empathy, brand management, time management-and the soft skills that help connect people and create trust. I guide my team toward giving the best service possible. Sometimes, we're not delivering good news. But the objective is to do that with compassion and empathy and in a way that we give the customer constructive next steps to move forward. We also know that as a newer, younger brand, customers may be wary of our credibility. It usually takes a few consistently excellent customer experiences to feel connected and loyal to the brand. That awesome experience starts from the very first touchpoint, whether it be web, email, brick and mortar, or Instagram, and carries through to when they're wearing our product

- EXISTING PROBLEM
- 
- **LITERATURE SURVEY**

A strong customer problem statement should provide a detailed description of your customer's current situation. Consider how they feel, the financial and emotional impact of their current situation, and any other important details about their thoughts or feelings.

Customer Satisfaction is an attitude that is decided based on the experience obtained. Satisfaction is an assessment of the characteristics or privileges of a product or service, or the product itself, that provides a level of consumer pleasure with regard to meeting consumer consumption needs.

Customer Satisfaction is the customer's response to the evaluation of perception of differences in initial expectations prior to purchase (or other performance standards) and the actual performance of the product as perceived after wearing or consuming the product in question.

The level of complaint is how high the complaint or delivery of dissatisfaction, discomfort, irritation, and anger over the service of the service or product. The dimension or indicator of complaint level is the high level of complaint.

Product Quality affects Customer Satisfaction, where the dimensions or indicators of Product Quality are quality products, in accordance with the price

offered, and ease of use affects the dimensions or indicators of Customer Satisfaction in relation to subscription decisions.

## • REFERENCES

1] Uddi Executive Overview: Enabling Service Oriented Architecture, 2004 Oct.

- “Web Services Architecture”, <http://www.w3.org/TR/ws-arch/>.

Date Accessed:11/02/2004.

- UDDI V3 Specification. <http://uddi.org/pubs/uddi-v3.00-published-20020719.html>.19/07/2002.

- Christensen E, Curbera F, Meredith G, Weerawarana S, Web Services DescriptionLanguage (WSDL) 1.1, W3C Note, 2001

- Ali A S, Rana O F, Ali R A, Walker D W, UDDIe: an extended registry for Webservices, SAINT-w’03 Proceedings of the 2003 Symposium on Applications and the Internet Workshops, 2003 Jan, pp .85-89.

- Tsai W T, Paul R, Cao Z, Yu L, Saimi A, Xiao B, Verification of Web Services Usingan Enhanced UDDI Server ,Proceedings of The Eighth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems ,2003 Jan, pp 131-38.

- Liu J, Gu N, Zong Y Ding Z, Zhang S, Zhang Q Service Registration and Discovery ina Domain-Oriented UDDI Registry ,

Proceedings of the 2005 The Fifth International Conference on Computer and InformationTechnology (CIT’05),2005, pp .276-83.

- Jian W, Zhaohui W, Similarity-based Web Service Matchmaking Proceedings of the2005 IEEE International Conference on Services Computing (SCC’05), 2005.

- Tretola G, Zimeo E , Structure Matching for Enhancing UDDI Queries Results , IEEEInternational Conference on Service-Oriented Computing and Applications(SOCA’07),2007.

- Ayorak E, Bener a, Super Peer Web Service Discovery Architecture, IEEE Proceedings

- International

Conference on Data

Engineering, 2007 pp

.287-94.

- Libing He W, Wu Y, Jianqun D A novel interoperable model of distributed UDDI -proceedings of the 2008 IEEE International

Conference on Networking, Architecture, and Storage - IEEE NAS 2008 Jun, pp .153-54.

- Nawaz F, Qadir K, Ahmad H F, and SEMREG-Pro: A Semantic based Registry for Proactive Web Service Discovery using Publish Subscribe Model, Fourth International Conference on Semantics, Knowledge and Grid, IEEE Xplore, 2008 Dec, pp .301-08.
- LIANG1 Q, CHUNG2 J A Federated UDDI System for Concurrent Access to ServiceData, IEEE International Conference on e-Business Engineering, ICEBE'08 - Workshops: AiR'08, EM2I'08, SOAIC'08, SOKM'08, BIMA'08, and DKEEE'08, 2008 Oct, pp .71-78.
- Vandan T, Nirmal D, Inderjeet Garg S, Garg N, Soni P. An Improved Discovery Engine for Efficient and Intelligent discovery of Web Service with publication facility, SERVICES 2009 - 5th 2009 World Congress on Services, 2009, pp .63-70.
- Ma C, Song M, Xu K, Zhang X. Web Service Discovery Research and Implementation Based on Semantic Search Engine, IEEE 2nd Symposium on Web Society, Beijing. 2010 Aug 16-17, pp .672-77.
- Rajendran T, Balasubramanie P, Flexible and Intelligent Architecture for Quality Based Web Service Discovery with an Agent- Based Approach, International Conference on Communication and Computational Intelligence (INCOCCI), Erode. 2010 Dec 27-29, pp .617-22.
- Johnsen F T, Hafsøe T, Eggen A, Griwodz C, Halvorsen P, Web Services Discovery across Heterogeneous Military Networks, IEEE Communications Magazine, 2010 Oct, 48(10), pp. 84-90.
- Ourania H, Georgios B, Mara N, Dimosthenis A, "A Specialized Search Engine for Web Service Discovery" IEEE 19th International Conference on Web Services, USA. 2012 June 24-29, pp. 448-55.
- Raj R J R, Sasipraba T., Web Service Recommendation Framework Using Qos Based Discovery and Ranking Process 3rd International Conference on Advanced Computing, ICoAC. Chennai. 2011 Dec 14-16, pp .371-77.
- Ren X, Hou R, Extend UDDI Using Ontology for Automated Service Composition 2nd International Conferences on Mechanic Automation and Control Engineering, MACE, 2011 July, pp .298-301.
- **PROBLEM STATEMENT DEFINITION**

A customer problem statement outlines problems that your customers face. It helps you figure out how your product or service will solve this problem for them.

The statement helps you understand the experience you want to offer your customers. It can also help you understand a new audience when creating a new product or service.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

A Customer Problem Statement is a detailed description of an issue that needs to be addressed. This document thoroughly elaborates on the problem that your product or your service solves for your particular customers. It takes into consideration your customer's unique pain points and how your product goals about solving their situation. A customer problem statement helps you and your team understand the detailed experience you are attempting to transform by analyzing and empathizing with your customers.

The customer problem statement is a critical component of a project. It benefits everyone involved with the project because it helps people understand why they're working on the project, providing clarity on the reasons behind the product or service. Team members will consider how your customers will be impacted by your project, what their thoughts and needs are, and thus come up with truly effective and valuable ways to improve their experience.

- **IDEATION & PROPOSED SOLUTION**
- **EMPATHY MAP CANVAS**





3

### Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Tip

Use a timer to keep track of time and to ensure you have enough time to brainstorm.

Category 1	Category 2	Category 3	Category 4
1. 2. 3. 4. 5. 6. 7. 8. 9. 10.	1. 2. 3. 4. 5. 6. 7. 8. 9. 10.	1. 2. 3. 4. 5. 6. 7. 8. 9. 10.	1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

4

### Group ideas

Take time sharing your ideas with clustering similar or related ideas as you go. Once all sticky notes have been grouped, give each cluster a common title label. It is useful to bigger than six sticky notes, try and use if you need break it up into smaller sub-groups.

10 minutes

### CUSTOMER'S EXPECTATIONS

Tip

Use a timer to keep track of time and to ensure you have enough time to group ideas.



### SECURITY



### SERVICES



### 4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

**Importance**  
A scale of how important an idea is to the business. It's a subjective measure, but it's a good one to use.

**Feasibility**  
A scale of how easy it is to implement an idea. It's a subjective measure, but it's a good one to use.

### 5 After you collaborate

You can expect the round to be longer or go faster when you collaborate and your company will benefit from it.

1. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

2. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

3. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

4. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

5. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

6. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

7. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

8. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

9. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

10. **Share the round**  
Share your ideas with the team and discuss them. It's a good idea to share your ideas with the team and discuss them.

## • PROPOSED SOLUTION

S.NO.	PARAMETER	DESCRIPTION
01	Problem Statement (Problem to be solved)	To solve customer issues using Cloud Application Development.
02	Idea / Solution description	Assigned Agent routing can be solved by directly routing to the specific agent about the issue using the specific Email. Automated Ticket closure by using daily sync of the daily database. Status Shown to the Customer can display the status of the ticket to the customer. Regular data retrieval in the form of retrieving lost data.
03	Novelty / Uniqueness	Assigned Agent Routing, Automated Ticket Closure, Status Shown to the Customer, and Backup data in case of failures.

S.NO.	PARAMETER	DESCRIPTION
04	Social Impact / Customer Satisfaction	Customer Satisfaction, Customer can track their status and Easy agent communication.
05	Business Model (Revenue Model)	<ul style="list-style-type: none"> <li>● Key Partners are Third-party applications, agents, and customers.</li> <li>● Activities held as Customer Service, System Maintenance.</li> <li>● Key Resources support Engineers, Multi-channel.</li> <li>● Customer Relationship have 24/7 Email Support, Knowledge-based channel.</li> <li>● Cost Structure expresses Cloud Platform, Offices</li> </ul>

S.NO.	PARAMETER	DESCRIPTION
06	Scalability of the Solution	The real goal of scaling customer service is providing an environment that will allow your customer service specialists to be as efficient as possible. An environment where they will be able to spend less time on gruntwork and more time on actually resolving critical customer issues

• **PROBLEM SOLUTION FIT**

## PROJECT DESIGN PHASE -I (PROBLEM-SOLUTION FIT)

### Problem-Solution fit canvas 2.0

Define CS, fit into	<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b> Who is your customer? 1) Customers who are not able to solve them Own complaints of what they are facing. 2) Customers who do not know the solution of their questions they get.	<b>6. CUSTOMER</b> <b>CC</b> What constraints prevent your customers from <u>adapting</u> to their choices of solutions? <u>spending power, budget, network, connection, available devices.</u> 1) This application will be supported by almost all the devices. 2) The solution we propose will have an alert via email feature, <u>if</u> expense exceed the given limit. 3) This solution also provides insights in a graphical way.	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> Which solutions are available to the customers when they have the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? <u>pen and paper is an alternative to digital notetaking</u> 1) By reading the guidelines properly. 2) Offer a solution and give options whenever possible. 3) Address to issue within the company. 4) By communicating properly	Explore AS.
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b> Which jobs to be done (or problems) do you address for your customers? There could be more than one; explore different sides. 1) The application <u>allow</u> the customers to find the solution for their queries. 2) They <u>will</u> be able to categorize their expenses. 3) They will be also given option for the general <u>questions.</u> 4) They also get the free solution where we provide our agents.	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> What is the real reason that this problem exists? What is the back story behind the need to do this job? <u>customers have to do because of the change in regulations.</u> 1) Lot of customers don't know the guidelines for their problems. 2) Some customers have of lack of <u>knowledge.</u> 3) Not knowing the answer to a question. 4) Not reading the guidelines properly	<b>7. BEHAVIOUR</b> <b>BE</b> What does your customer do to address the problem and get the job done? <u>directly related: find the right solar panel installer, calculate usage and benefits;</u> <u>indirectly associated: customers spend less time on collecting work (i.e. Googlepay)</u> 1) Make sure he/she reads the guidelines properly. 2) Make sure they find a proper solution <u>for</u> their queries.	
<b>3. TRIGGERS</b> <b>TR</b> What triggers customers to act? <u>seeing their <del>neighbour</del> installing</u> <u>solar panels, reading about a more efficient solution in the news.</u> 1) Customers can know to solve their solutions.	<b>10. YOUR SOLUTION</b> <b>SL</b> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits really. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer <u>behaviour.</u> 1) To design a personal help desk using flask. 2) To provide insights on their queries in a graphical way.	<b>8. CHANNELS OF BEHAVIOUR</b> <b>CH</b> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7 1) All their data are secured and being updated to cloud storage <b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. 1) Make sure they find the best solutions for their complaints.	Extract online & offline CH of BE	
<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> How do customers feel when they face a problem or a job and afterwards? <u>frustrated, insecure - confident, in control - use it in your communication strategy &amp; design.</u> 1) Customers can get the from the help desk.				

Problem-Solution Fit

4

## • REQUIREMENT ANALYSIS

### • FUNCTIONAL REQUIREMENT

FR No	Functional Requirement(Epic)	Sub Requirement(Story/ Sub-Task)
1	User Registration	Registration through Form Registration through Gmail Registration through Google
2	User Confirmation	Confirmation via Email Confirmation via OTP
3	User Login	Login via Google Login with Email id and Password
4	Admin Login	Login via Google Login with Email id and Password
5	Query Form	Description of the issues Contact information
6	E-mail	Login alertness
7	Feedback	Customer feedback

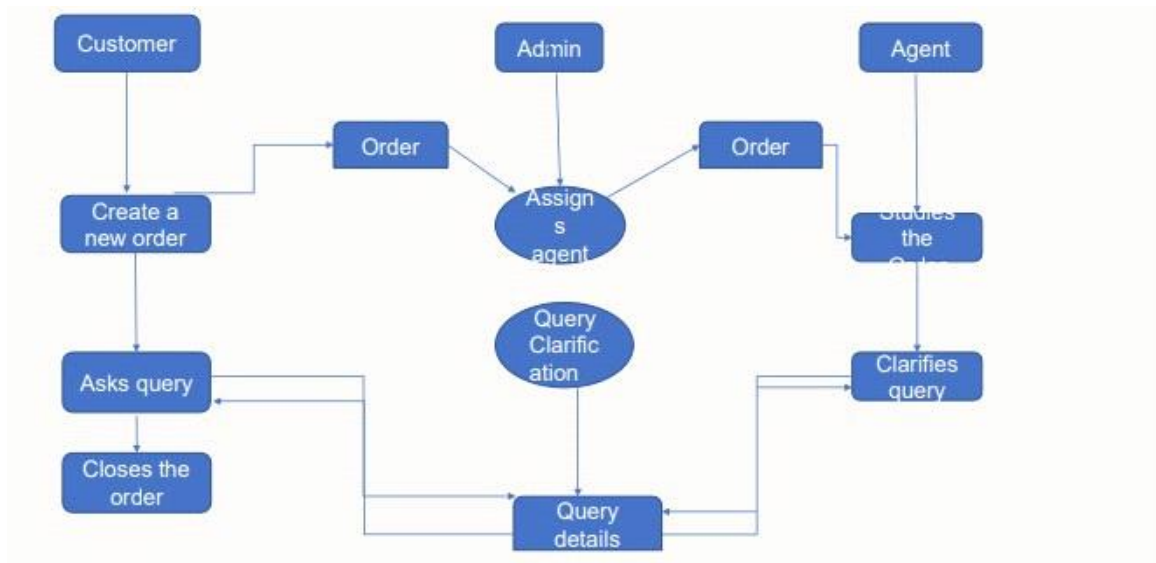
- **NON-FUNCTIONAL REQUIREMENT**

FR No	Non-Functional Requirement	Description
1	Usability	To provide the solution to the problem
2	Security	Track of login authentication
3	Reliability	Tracking of decade status through email
4	Performance	Effective development of web application
5	Availability	24/7 service
6	Scalability	<u>Agents</u> scalability as per the number of customers

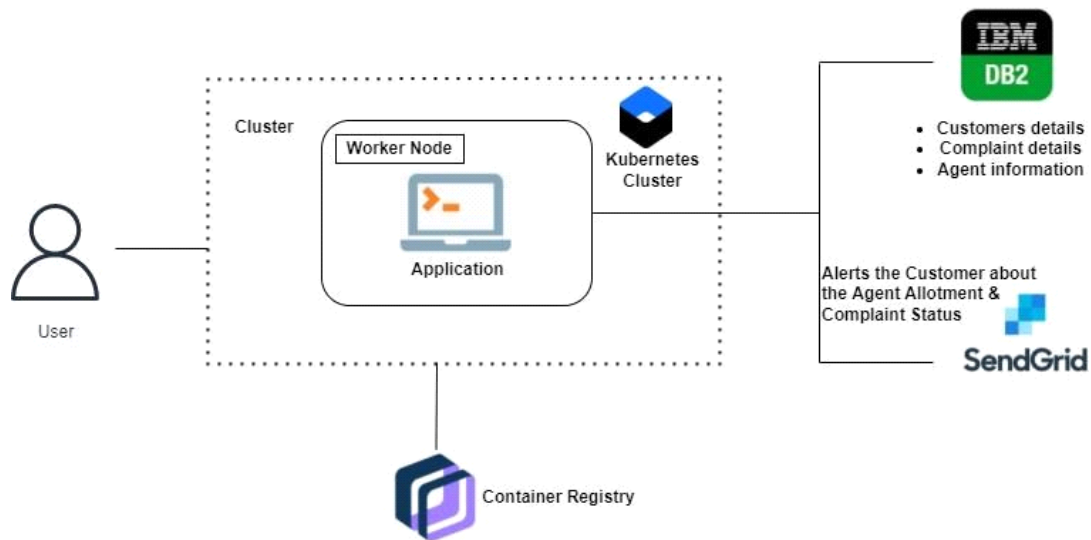
- **DATA FLOW DIAGRAMS**

- 

- **PROJECT DESIGN**



## • SOLUTION AND TECHNICAL ARCHITECTURE



## • USER STORIES



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a customer, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	login	USN-2	As a customer, I can login to the application by entering correct email and password.	I can access my account/dashboard.	High	Sprint-1
	Dashboard	USN-3	As a customer, I can see all the orders raised by me.	I get all the info needed in my dashboard.	Low	Sprint-2
	Order creation	USN-4	As a customer, I can place my order with the detailed description of my query	I can ask my query	Medium	Sprint-2
	Address Column	USN-5	As a customer, I can have conversations with the assigned agent and get my queries clarified	My queries are clarified.	High	Sprint-3
	Forgot password	USN-6	As a customer, I can reset my password by this option incase I forgot my old password.	I get access to my account again	Medium	Sprint-4
	Order details	USN-7	As a Customer I can see the current stats of order.	I get abetter understanding	Medium	Sprint-4
Agent (web user)	Login	USN-1	As an agent I can login to the application by entering Correct email and password.	I can access my account / dashboard.	High	Sprint-3
	Dashboard	USN-2	As an agent, I can see the order details assigned to me by admin.	I can see the tickets to which I could answer.	High	Sprint-3
	Address column	USN-3	As an agent, I get to have conversations with the customer and clear his/er dobuts	I can clarify the issues.	High	Sprint-3
	Forgot password	USN-4	As an agent I can reset my password by this option in case I forgot my old password.	I get access to my account again.	Medium	Sprint-4

Admin (Mobile user)	Login	USN-1	As a admin, I can login to the appliaction by entering Correct email and password	I can access my account/dashboard	High	Sprint-1
	Dashboard	USN-2	As an admin I can see all the orders raised in the entire system and lot more	I can assign agents by seeing those order.	High	Sprint-1
	Agent creation	USN-3	As an admin I can create an agent for clarifying the customers queries	I can create agents.	High	Sprint-2
	Assignment agent	USN-4	As an admin I can assign an agent for each order created by the customer.	Enable agent to clarify the queries.	High	Sprint-1
	Forgot password	USN-5	As an admin I can reset my password by this option in case I forgot my old password.	I get access to my account.	High	Sprint-1

## • PROJECT PLANNING & SCHEDULE
















### • SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the website and go through the services available on the webpage.	20	Medium	Shyam R Suiindhar C
Sprint-2	Agent Panel	USN-2	The role of the agent is to check out the complaint tickets and to contact the user and solve the complaint they raise.	20	High	Shankar P Jayasurya C
Sprint-3	Admin Panel	USN-3	The role of the admin is to check out the database about the availability and have a track of all the things that the users are going to experience and manage the agent and complaint tickets.	20	High	Shankar P
Sprint-4	Chat Bot	USN-4	The user can directly talk to Chatbot regarding the services. Get the recommendations based on information provided by the user.	20	High	Jayasurya S
Sprint-5	Final Delivery	USN-5	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	Suiindhar C Shyam R Shankar P Jayasurya S

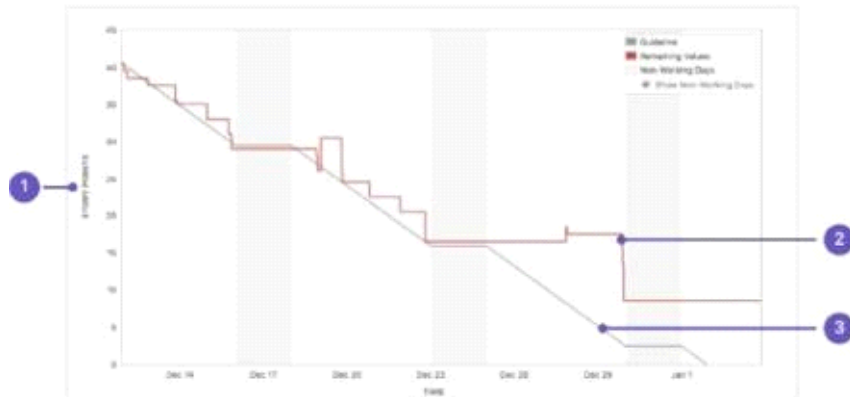
## • SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	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	6	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	6	3 Days	07 Nov 2022	09 Nov 2022	6	09 Nov 2022
Sprint-4	5	3 Days	09 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-5	8	6 Days	13 Nov 2022	19 Nov 2022	8	19 Nov 2022

## • REPORTS FROM JIRA

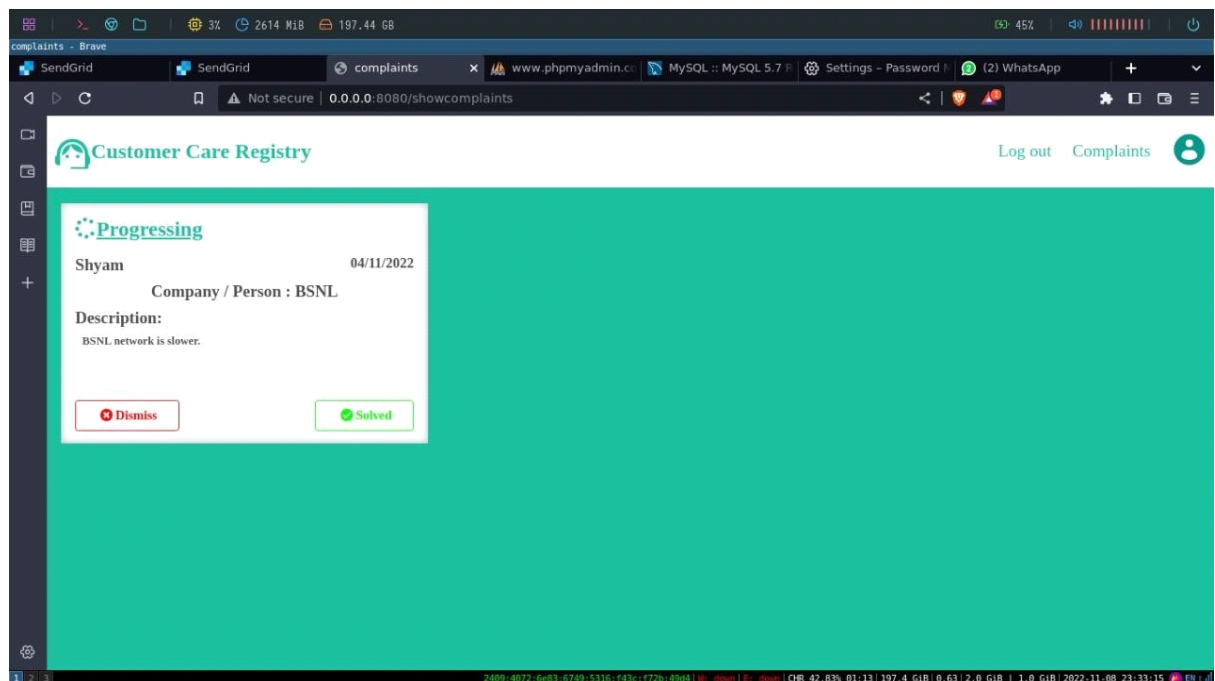
 ECR-3 The user will login into the website and go through...	DONE  
 ECR-4 The role of the agent is to check out the complaint...	DONE  
 ECR-5 The role of the admin is to check out the database...	DONE  
 ECR-6 he user can directly talk to Chatbot regarding the ...	DONE  
 ECR-7 Container of applications using docker kubernetes...	DONE  



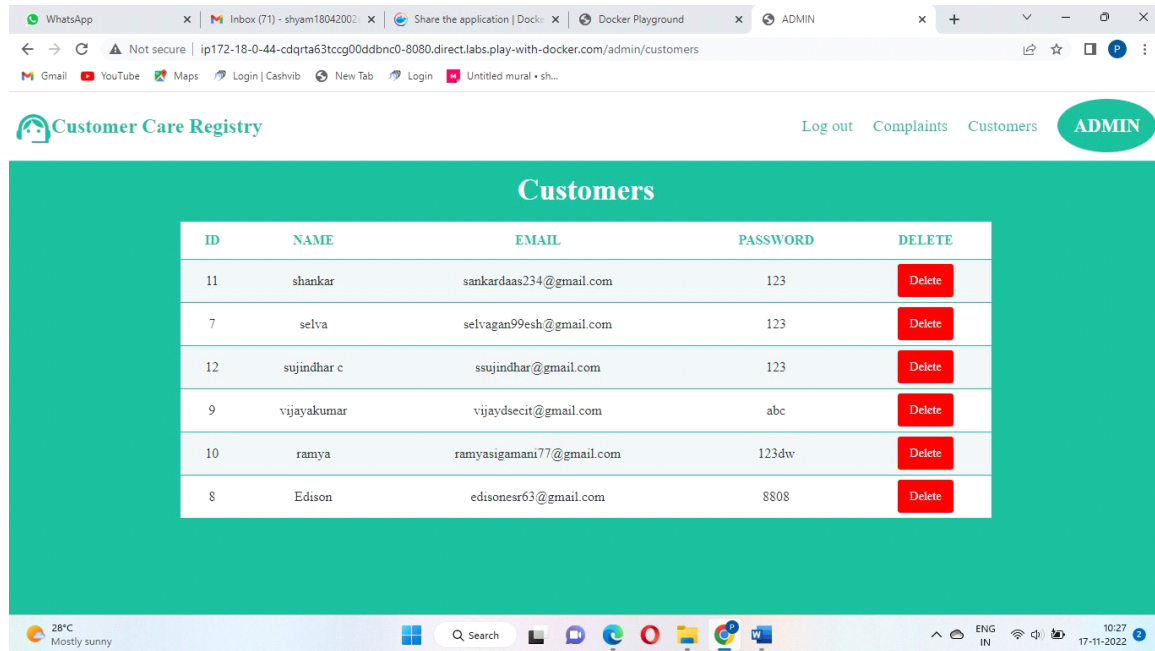


Burndown Graph

- **FEATURE 1**
- 
- **CODING & SOLUTIONING**

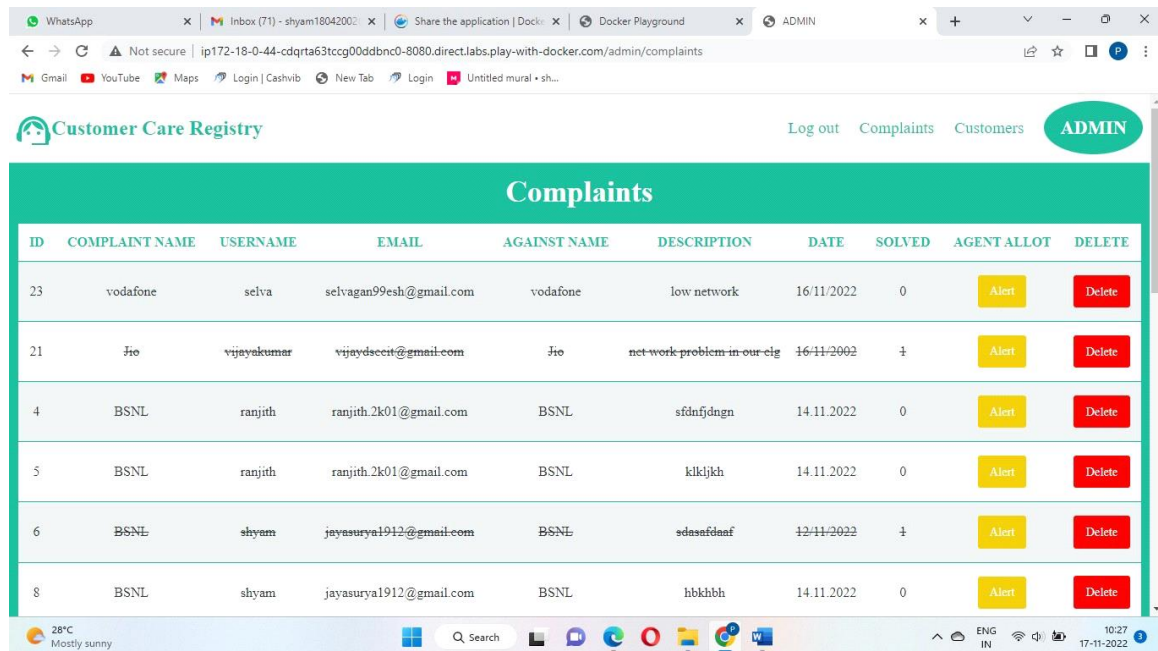


- **FEATURE 2**



The screenshot shows a web browser window with the URL `ip172-18-0-44-cdqta63tccg00ddbnc0-8080.direct.labs.play-with-docker.com/admin/customers`. The page is titled "Customer Care Registry" and has a navigation bar with "Log out", "Complaints", "Customers", and an "ADMIN" button. The main content area is a teal box with the heading "Customers" and a table with the following data:

ID	NAME	EMAIL	PASSWORD	DELETE
11	shankar	sankardaas234@gmail.com	123	<button>Delete</button>
7	selva	selvagan99esh@gmail.com	123	<button>Delete</button>
12	sujindhar c	ssujindhar@gmail.com	123	<button>Delete</button>
9	vijayakumar	vijaydsecit@gmail.com	abc	<button>Delete</button>
10	ramya	ramyasigamani77@gmail.com	123dw	<button>Delete</button>
8	Edison	edisonesr63@gmail.com	8808	<button>Delete</button>



The screenshot shows a web browser window with the URL `ip172-18-0-44-cdqta63tccg00ddbnc0-8080.direct.labs.play-with-docker.com/admin/complaints`. The page is titled "Customer Care Registry" and has a navigation bar with "Log out", "Complaints", "Customers", and an "ADMIN" button. The main content area is a teal box with the heading "Complaints" and a table with the following data:

ID	COMPLAINT NAME	USERNAME	EMAIL	AGAINST NAME	DESCRIPTION	DATE	SOLVED	AGENT ALLOT	DELETE
23	vodafone	selva	selvagan99esh@gmail.com	vodafone	low network	16/11/2022	0	<button>Alert</button>	<button>Delete</button>
21	Jio	vijayakumar	vijaydsecit@gmail.com	Jio	net-work problem in our eig	16/11/2022	1	<button>Alert</button>	<button>Delete</button>
4	BSNL	ranjith	ranjith.2k01@gmail.com	BSNL	sfdnfdngn	14.11.2022	0	<button>Alert</button>	<button>Delete</button>
5	BSNL	ranjith	ranjith.2k01@gmail.com	BSNL	kikijkh	14.11.2022	0	<button>Alert</button>	<button>Delete</button>
6	BSNL	shyam	jayasurya1912@gmail.com	BSNL	edasaafdaaf	12/11/2022	1	<button>Alert</button>	<button>Delete</button>
8	BSNL	shyam	jayasurya1912@gmail.com	BSNL	hbkhhb	14.11.2022	0	<button>Alert</button>	<button>Delete</button>

- **DATABASE SCHEMA**

## Table definition



COMPLAINTS

No statistics available.

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	
USERNAM E	VARCHAR	N	150	0	
EMAIL	VARCHAR	N	150	0	
AGAINST_ PERSON	VARCHAR	N	150	0	
DES	VARCHAR	N	150	0	

## Table definition



COMPLAINTS

No statistics available.

Name	Data type	Nullable	Length	Scale	
AGAINST_ PERSON	VARCHAR	N	150	0	
DES	VARCHAR	N	150	0	
DATE	VARCHAR	N	150	0	
SOLVED	VARCHAR	N	150	0	

## Table definition

: ✕

CUSTOMERDEATILS

No statistics available.

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	👁
USERNAME	VARCHAR	N	150	0	👁
EMAIL	VARCHAR	N	150	0	👁
PASSWRD	VARCHAR	N	150	0	👁

- **TEST CASES**

- 

- **TESTING**

- **FUNCTIONAL TESTING**

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

- **WHITE BOX TESTING:**

Testing based on an analysis of internal workings and structure of a piece of software. This testing can be done using the percentage value of load and energy. The tester should know what exactly is done in the internal program. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing.

- **BLACK BOX TESTING:**

Testing without knowledge of the internal workings of the item being tested. Tests are usually functional. This testing can be done by the user who has no knowledge of how the shortest path is found.

- **USER ACCEPTANCE TESTING**

Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages. The function of performance characteristics to specification and is accepted. A deviation from specification is uncovered and a deficiency list is created. User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

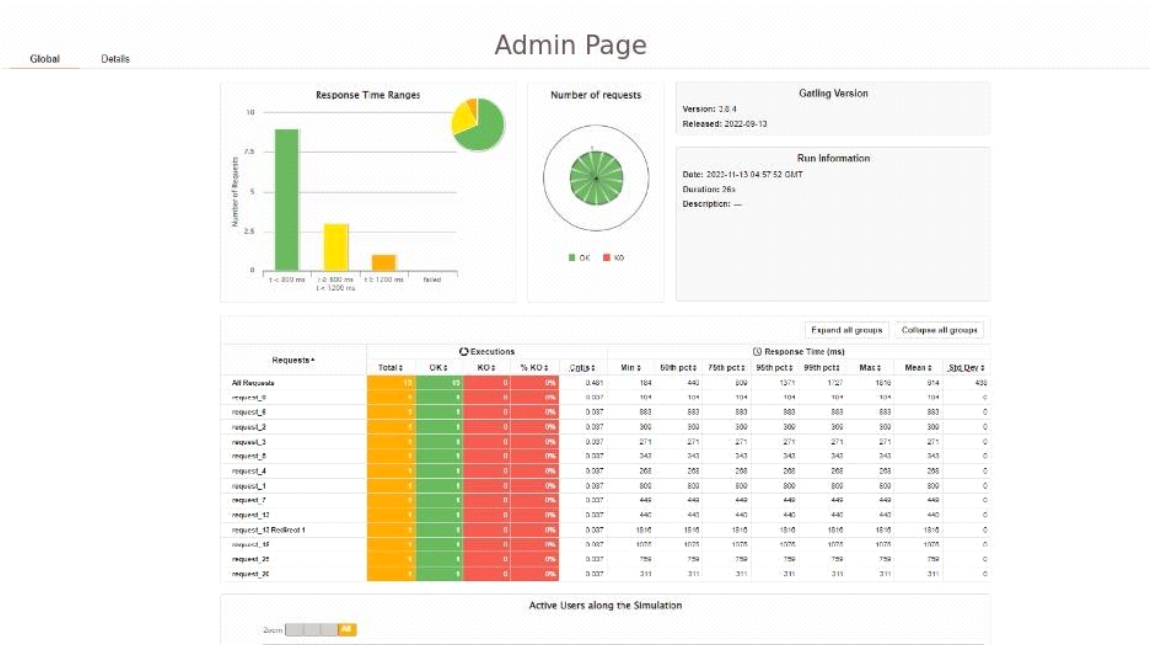
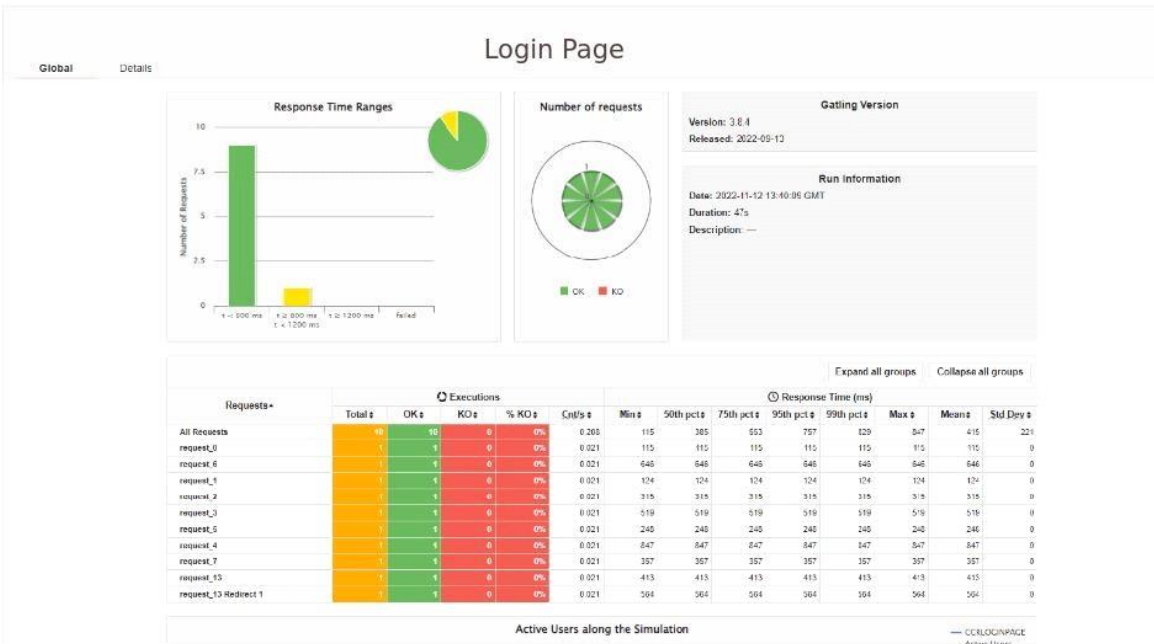
- **TEST RESULTS**

A	B	C	D	E	F	G	H	I
Sprint 1 UI/UX								
Testcase	Type	Component	Scenario	Step to execute	Expected result	Actual result	Status	Executed by
1	UI	login/signup page	clicking on site link	click in site link	login/signup page loads	page load	PASS	Shyam
2	Functional	login/signup page	login in to user account	enter credentials	login to home page	home page loads	PASS	
3	Functional	login/signup page	signup a user	enter user details	login to home page	home page loads	PASS	
4	Functional	home page	logout of the home page	logout the user	back to login page	login page loads	PASS	
5	Functional	login/signup page	login with unregistered user	redirect to signup page	back to signup page	signup page loads	PASS	
6	Functional	login/signup page	signup a existing email	use a existing user email	back to sign up page	signup page loads	PASS	
7	Functional	wrong password	sign in with wrong password	login with wrong password	back to sign up page	signup page loads	PASS	
8	Functional	wrong email	sign in with wrong email	login with wrong email	back to signup page	signup page loads	PASS	
Sprint 2 db2								
1	Functional	complaint page	display registered complaints	click on complaint tab	complaint list down	complaint list down	PASS	
2	Functional	complaint page	clicking on solve	click on solve button	solve the complaint	complete the complaint	PASS	
3	Functional	complaint page	click on the dismiss	click dismiss button	delete the complaint	deletes the complaint	PASS	
4	Functional	complaint page	fill up the complaint form	click submit	create the complaint	create a new complaint	PASS	

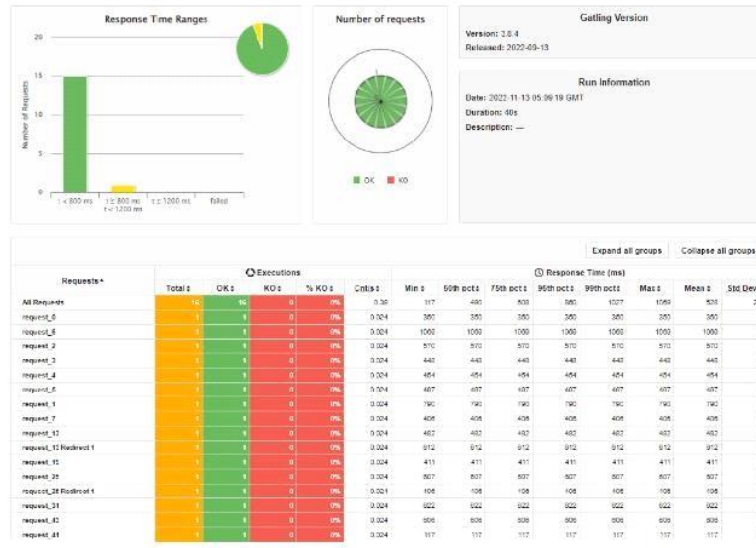
## 9.1 PERFORMANCE METRICES

-

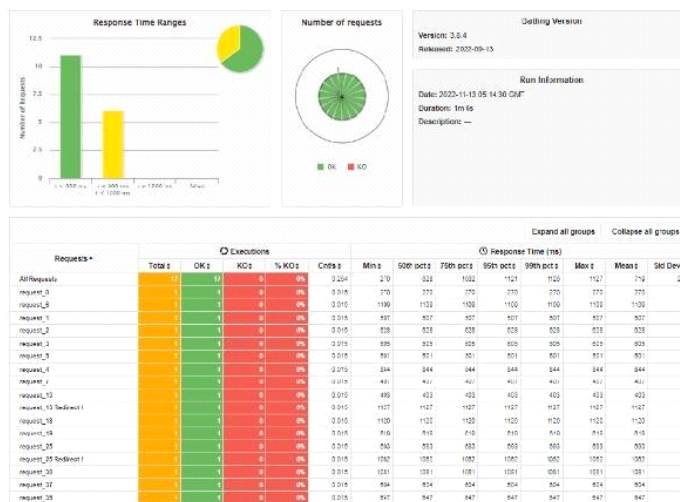
• RESULTS



## Complaint page



## Complaint form



### Advantage

- 
- **ADVANTAGES & DISADVANTAGES**

- ☐ Flow sheet is a powerful tool to monitor clinical data and track trends
- ☐ Provides a dashboard of who needs what
- ☐ Provides total population data reporting with no chart abstraction

- ☐ Generates revenue (it shows when services are needed)
- ☐ Provides outreach information at fingertips
- ☐ Improves team-based care
- ☐ Smaller software package than EHRs
- ☐ Creating loyal customers through good customer service can provide businesses with lucrative long-term relationships.
- ☐ Customer loyalty. Loyal customers have many benefits for businesses

#### **Disadvantage**

- ☐ Disease-specific, not longitudinal
- ☐ Does not include information necessary for billing
- ☐ Requires hardware, software and maintenance
- ☐ Requires data entry and data maintenance
- ☐ Parallel documentation system (i.e., some information has to be entered in two systems)
- ☐ Can't stand alone, must have an additional documentation system.
- ☐ Experience burnout and stress. Working as a customer service representative requires you to maintain a friendly demeanour at all times, regardless of how customers act or how you personally feel

### **• CONCLUSION**

Companies today are modernizing customer care, using advanced AI to ensure a positive customer experience starting from the first interaction and throughout the buyer's journey. To properly manage customer care, companies must understand how they are succeeding and what needs improvement. This requires establishing key performance indicators (KPIs) for customer service and creating a system of gathering metrics across channels. In conclusion, customer care, involves the use of basic ethics and any company who wants to have success and grow, needs to remember, that in order to do so, it must begin with establishing a code of ethics in regards to how each



employee is to handle the dealing with customers. Customers are at the heart of the company and its growth or decline. Customer care involves, the treatment, care, loyalty, trust the employee should extend to the consumer, as well in life. This concept can be applied to so much more than just customer care. People need to treat others with respect and kindness; people should try to take others into consideration when making any decision. If more people were to practice this policy, chances are the world would be a better, more understanding place for all to exist. Thereby, the customer care registry would be far helpful and approachable. It offers easy tracking, recording and notification than any other means.

### • FUTURE SCOPE

The current state of customer care registry, in so many companies, looks something like this:

- ❑ Customer acquisition is prioritised over retention
- ❑ Customer service investment projects are sidelined.
- ❑ Departmental efficiency is of highest priority.
- ❑ Businesses see employees in the customer service department as short-term and disposable. They are there to fulfil a specific, repetitive, purpose.
- ❑ Employees are considered unskilled and leaders hire accordingly.
- ❑ New agents view customer service as a 'last resort' or 'short term' job. People often see careers in customer support as unambitious.
- ❑ Agent training rarely goes beyond product and people skills.

In the next 3-5 years, we expect to see these **future customer care registry trends**:

- ❑ The shift from a primarily 'cost centre' to primarily 'growth centre' worldview.

- The job desk for a customer care registry director will focus more on leadership, innovation, and ability to drive company-wide improvement.
- Customer service will shift to become a strategic partner of marketing, sales, and product development. CS will help with direction, project prioritisation, and impact.
- A need for customer service leaders to take a highly strategic seat at the table. They'll need to argue for investment in talent, technology, and innovation.
- A shift in performance metrics. Forget of resolved tickets. In the future, we'll measure performance based on of customers saved from the precipice of churn.
- A career in customer care registry will not be a last resort. Top graduates will prioritise getting an education in strategic customer interaction.
- Focus on ticket deflection will reduce because brands will view each customer interaction as an opportunity to learn, build a relationship, and grow profits. They deserve a well-trained, human touch.

Modern and developing technology enables this future to exist. With new technology, administrative tasks will tend toward zero.

- The sole purpose of the customer service is to meet the expectations of the customers so that they are satisfied with the outcome. These services are also available to understand the queries of the customers and ensure that they enjoy a cost-effective experience after purchasing any product from the respective company.

## **SOURCE CODE**

- 

## **• APPENDIX**

```
from flask import
Flask,render_template,request,url_for,session,redirectfrom
flask_mysqlldb import MySQL
from sendmail import
sendemail,forget_password_mail,updated_password_mail,solve_mail
imp
ort
json
imp
ort
ibm
_db
imp
ort
re
from random
import randint
from datetime
import date

app = Flask(__name__)
# http://remotemysql.com/
```

```

# dsn_hostname = "b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud"

# dsn_uid = "dmt13873"

# dsn_pwd =
"740yZ1Yq8Uj2E4qm"#

dsn_database = 'bludb'

# dsn_port = 31249

conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-
46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;P
ORT=3124
9;SECURITY=SSL;SSLServerCertificate=src/DigiCertGlobalRootCA.
crt;UID=d mt13873;PWD=740yZ1Yq8Uj2E4qm",",") # type: ignore

print(conn)

print("connection successful...")


# database configuration

# app.config['MYSQL_HOST'] =
'sql12.freesqldatabase.com'#

app.config['MYSQL_USER'] = 'sql12552843'

# app.config['MYSQL_PASSWORD'] =
'zWlZHmXNi8'# app.config['MYSQL_DB'] =
'sql12552843' app.secret_key = "super secret
key"

# mysql = MySQL(app)

```

```

@a
pp.r
oute
('/')
def
ho
me(
):
    today = date.today()
    current_date =
today.strftime('%d/%m/%Y')if
"google_token" in session:
    session["current_date"] =
current_datereturn
render_template('home.htm
l')
if "username" in session:
    session["current_date"] =
current_datereturn
render_template('home.htm
l')
return render_template('index.html')

```

```

# manually registration

@app.route('/register',methods=[
"POST"])def register():
    if request.method ==
        'POST': name =
            request.form['una
            me']

            mail =
            request.form['m
            ail']pwd =
            request.form['p
            wd']
            cpwd = request.form['confirmpwd']

            if not
                re.match(r'^@[^@]+\.[^@]+'
                , mail):msg = 'Invalid email
                address !'

                return

            render_template('index.html',signupmsg=ms
            g)if pwd != cpwd:
                msg = 'Please enter correct confirm password'

```

```

        return

render_template('index.html',signupmsg=msg)

# check account is exists or not

# cursor =

mysql.connection.cursor()

rCheckQuery = "

result = ibm_db.exec_immediate(conn,f"SELECT * FROM
customerdeatils WHERE email LIKE '{mail}')"

# cursor.execute('SELECT * FROM customerdeatils WHERE email
LIKE
% s',[mail])

# existing_user =

cursor.fetchone()#

cursor.close()

existing_user =

ibm_db.fetch_row(result)

#exits

if existing_user:

    msg = 'Account already exists please login.'

    return

render_template('index.html',signupmsg =

msg)# not exists

# cursor = mysql.connection.cursor()

```

```

        # cursor.execute('INSERT INTO customerdeatils
VALUES(null,% s,% s,% s)',(name,mail,pwd))

        #
        mysql.connection.
        commit()#
        cursor.close()

        regInsertQuery = f"INSERT INTO customerdeatils
(username,email,passwd) VALUES('{name}','{mail}','{pwd}')"

        insertflag =
        ibm_db.exec_immediate(conn,regInsertQuery)

        msg = 'Your registration successfully
        completed.'

        # send mail
        sendemail(mail,'Accoun
        t_creation')

    return

    render_template('index.html',signupmsg =
    msg)# admin page
    @app.route('/admin
    /<which>')def
    admin(which):

        if which == 'customers':

            # cursor = mysql.connection.cursor()

```



```

        result = ibm_db.exec_immediate(conn,'SELECT * FROM
customerdeatils')data = []
        while ibm_db.fetch_row(result):

            temp =
[ibm_db.result(result,0),ibm_db.result(result,1),ibm_db.result(result,2),ibm_db.
re sult(result,3)]

            data.append(temp)

        return

render_template('admin.html',customers=data,complaints=None)
e)if which == 'complaints':

    # cursor = mysql.connection.cursor()

    result = ibm_db.exec_immediate(conn,'SELECT * FROM
complaints')data = []

    while ibm_db.fetch_row(result):

        temp =
[ibm_db.result(result,0),ibm_db.result(result,1),ibm_db.result(result,2),ibm_db.
re sult(result,3),ibm_db.result(result,4),ibm_db.result(result,5)]

        data.append(temp)

    return

render_template('admin.html',customers=None,complaints=data)

# admin delete

@app.route('/Delete/<type>/'<id>')def
Delete(type,id):

    if type == 'customers':

```

```

        # cursor = mysql.connection.cursor()

        result = ibm_db.exec(conn,f'DELETE FROM customerdeatils
WHERE id="{id}"')

        #

        mysql.connection.
        commit()#

        cursor.close()

        return

    redirect(url_for('admin',which='customers'
))if type == 'complaints':

        # cursor = mysql.connection.cursor()

        result = ibm_db.exec_immediate(conn,f'DELETE FROM
complaintsWHERE id = {id}')

        #

        mysql.connection.
        commit()#

        cursor.close()

        return

    redirect(url_for('admin',which='complaints'))

# manually login

@app.route('/login',methods=['POST','GET'])

def login():

    if request.method

        == 'POST': mail

```

```

=
request.form['mail']
password =
request.form['pwd1']#
login is admin or not
if mail == "admin" and password ==
    'admin@1810': return
    redirect(url_for('admin',which='customers')
    )
# check account is exists or not

# cursor = mysql.connection.cursor()

query = "SELECT * FROM customerdetails WHERE
email=? ANDpassword=?"

stmt = ibm_db.prepare(conn, query) #
type:ignore
ibm_db.bind_param(stmt,1,mail) #
type:ignore
ibm_db.bind_param(stmt,2,password) #
type:ignoreibm_db.execute(stmt) #
type:ignore
user = ibm_db.fetch_assoc(stmt) #
type:ignoreprint(user,password)

```

```
#
e
x
i
s
t
s

i
f

u
s
e
r
:

    session["username"] =
    user['USERNAME']
    session['mail'] = mail
    return
render_template('home.html',username=session["username"],mail=session["ma
il"
])

else:

    msg = 'mail or password is not valid.'
```

```

        return render_template('index.html',signinmsg=msg)

    if request.method ==

        "GET": return

        redirect(url_for('home'))

# logout
method

@app.route
('/logout'
)def
logout():
    if "username" in
        session:
            session.pop("
            username")
    if "google_token" in
        session:
            session.pop("google_token")
            session.pop("mail
            ")
    if "mail" in
        session:

```

```
session.p  
op("mail  
")  
return redirect(url_for('home'))
```

```
# complaint register
```

```
@app.route('/complaint',methods=['
```

```
POST'])def complaint():
```

```
    if request.method == 'POST':
```

```
        complaint_name =
```

```
        request.form['complaint_name']name =
```

```
        request.form['name']
```

```
        mail = request.form['email']
```

```
        against_person =
```

```
        request.form['against_person']date =
```

```
        request.form["date"]
```

```
        des =
```

```
        request.form['complaint_
```

```
        des'] # cursor =
```

```
        mysql.connection.cursor()
```

```
    if not name == session["username"] or not mail ==
```

```
        session["mail"]:msg = "please don't change username
```

```
        and mail."
```

```

        return render_template('home.html',msg=msg)

    result = ibm_db.exec_immediate(conn,f"INSERT INTO
complaints(username,email,against_person,des,date,solved)
VALUES('{name}','{mail}','{against_person}','{des}','{date}'
,{ '0'})")

    #

    mysql.connection.
commit()#
cursor.close()

sendemail(mail,'complaint_creation')

msg = 'Complaint registerd you check out
complaints section.'return
render_template('home.html',msg=msg)

# show complaints
and progress

@app.route('/showco
mplaints') def
showcomplaints():

    # cursor = mysql.connection.cursor()

    # cursor.execute("SELECT * FROM complaints WHERE
username= % sAND email=%
s",(session["username"],session["mail"]))

```

```

# details =

cursor.fetchall()

# cursor.close()

query = "SELECT * FROM complaints WHERE username=?
AND email=?" stmt = ibm_db.prepare(conn, query) # type:ignore
ibm_db.bind_param(stmt,1,session["username"]) # type:ignore
ibm_db.bind_param(stmt,2,session['mail']) # type:ignore
ibm_db.execute(stmt)

data = []

while ibm_db.fetch_row(stmt):

    temp =
[ibm_db.result(stmt,0),ibm_db.result(stmt,1),ibm_db.result(stmt,2),ibm_db.resu
lt (stmt,3),ibm_db.result(stmt,4),ibm_db.result(stmt,5),ibm_db.result(stmt,6)]

    print(te
mp)

    data.ap
pend(te
mp)

return render_template('complaints.html',complaints=data)

# update complaint

@app.route('/solve',methods=
["POST"])def
solve_complaint():

```



```

if request.method
    == "POST":c_id
    =
    request.form['c_i
    d']

    print(c_id)

    # cursor = mysql.connection.cursor()

    # cursor.execute("UPDATE complaints SET solved = % s
    WHERE id = %s",('1',c_id,))

    query = "UPDATE complaints SET solved = '1'
    WHERE id = ?"# mysql.connection.commit()

    stmt = ibm_db.prepare(conn, query) #
    type:ignore
    ibm_db.bind_param(stmt,1,c_id) #
    type:ignore ibm_db.execute(stmt)

    detail =

    ibm_db.result(stmt
    ,0)print(detail)

    # cursor.execute("SELECT * FROM complaints WHERE id =
    % s",[c_id])query2 = "SELECT * FROM complaints WHERE
    id = ?"

    stmt1 = ibm_db.prepare(conn, query2) # type:ignore

```

```

    ibm_db.bind_param(stmt1,1,c_id) #
    type:ignoreibm_db.execute(stmt1)
    details =
    ibm_db.result(stmt1,
    0)# cursor.close()
    print(details)

    # solve_mail(session['mail'],'user')

    return

    redirect(url_for('showcomplaints'
    ))return
    redirect(url_for('showcomplaints'
    ))

```

```

# # admin agent allot

```

```

#

```

```

@app.route('/solve_admin',methods=["
POST"])# def solve_admin():

```

```

#     if request.method == "POST":

```

```

#         c_id = request.form['c_id']

```

```

#         # cursor = mysql.connection.cursor()

```

```

#         cursor.execute("SELECT * FROM complaints WHERE id =

```

```

% s",[c_id])#     query = "SELECT * FROM complaints WHERE id =

```

```

?"

```

```

#     details =
cursor.fetchone()#
        cursor.close()
#     solve_mail(details[3],'admin')

#     return
redirect(url_for('admin',which='complaints'))#
        return
redirect(url_for('admin',which='complaints'))
# remove complaint
@app.route('/dismiss',methods=[
"POST"])def
dismiss_complaint():
    if request.method == "POST":

        c_id = request.form["c_id"]
        # cursor = mysql.connection.cursor()
        # cursor.execute("DELETE FROM complaints WHERE id =
        % s",[c_id])# mysql.connection.commit()
        # cursor.close()

        query = "DELETE FROM complaints
        WHERE id = ?"stmt =
        ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,c_id) #
        type:ignore ibm_db.execute(stmt)

```

```

        return
    redirect(url_for('showcomplaints'
    ))return
    redirect(url_for('showcomplaints'
    ))
# send otp in user mail id
@app.route('/send_otp',methods=["POST
", "GET"])def send_otp():
    if request.method
        == "POST":mail
        =
        request.form["m
        ail"]
        cursor = mysql.connection.cursor()

        cursor.execute("SELECT * FROM customerdeatils WHERE
email = %s",[mail])

        temp =
        cursor.fetcho
        ne()
        cursor.close()

        if not temp:

            return
    render_template('forget.html',type='otp',msg1='Your account
doesn\'t exist please register')

```

```

        otp = randint(10 **
5,10**6)
        forget_password_mail(
            mail,otp)
        session["otp"] = otp

    return

render_template('forget.html',type='update_password',tempmail=mail)
# forget password method
@app.route('/forgetpassword/<type>',methods=["POST","GET"])
def
    forgetpassword(
        type):
        if
            type ==
            'otp':
                return
            render_template('forget.html',type=type)
        elif request.method == "POST":
            mail =
            request.form["mail"]
            otp =
            request.form["otp"]

```

```

pwd =
request.form["password
"] c_pwd =
request.form["con_pwd
"]
print(otp,session['otp'])
if not pwd == c_pwd:
    msg = 'Please Enter Password properly'
    return
render_template('forget.html',type='updatePassword',msg=msg)
if not otp == str(session['otp']):
    msg = "Your OTP is Incorrect."
    return
render_template('forget.html',type='updatePassword',msg=msg)
cursor = mysql.connection.cursor()
cursor.execute("UPDATE customerdeatils SET passwrđ = %
s WHEREemail = % s",(pwd,mail))
mysql.connectio
n.commit()
cursor.close()
msg = 'password updated
successfully'
updated_password_mail(m
ail)

```

```
return render_template('forget.html',type='updatePassword',msg=msg)
```

```
if __name__ == '__main__':
```

```
    app.run(host = '0.0.0.0',port = 8080,debug=True)
```

### **GitHub & Project Demo Link**

**GitHub link:** <https://github.com/IBM-EPBL/IBM-Project-12303-1659447004>

**Demo link:** <https://youtu.be/XVklWCRjMCo>