Customer Care Registry Web Application

TEAM MEMBERS:
E SriJayan (Team Leader)
N Mohammed Junaid Alam
G Rajesh
V Ravi

INTRODUCTION

Project Overview:

After the deployment of a product, it becomes indispensable to get feedback from the customers for the betterment of the product in the long run. We have created a web appliation to make it easier for the customers/clients to post their grievances/queries/complaints online and have them resolved by an agent as soon as possible.

Purpose:

Customers facing problems in the product purchased or service provided is very common. For getting these issues resolved, customers need to physically visit/call the organisation/company which makes it an inconvenient process. Our objective is to create a customised application which allows customers to raise their issue which will be forwarded to admin who assigns agents to rectify the issue.The customer can also keep track of issue to know the current status.

LITERATURE SURVEY

Existing Problem:

Customers facing problems in the product purchased or service provided is very common. For getting these issues resolved, customers need to physically visit/call the organisation/company which makes it an inconvenient process.

References:

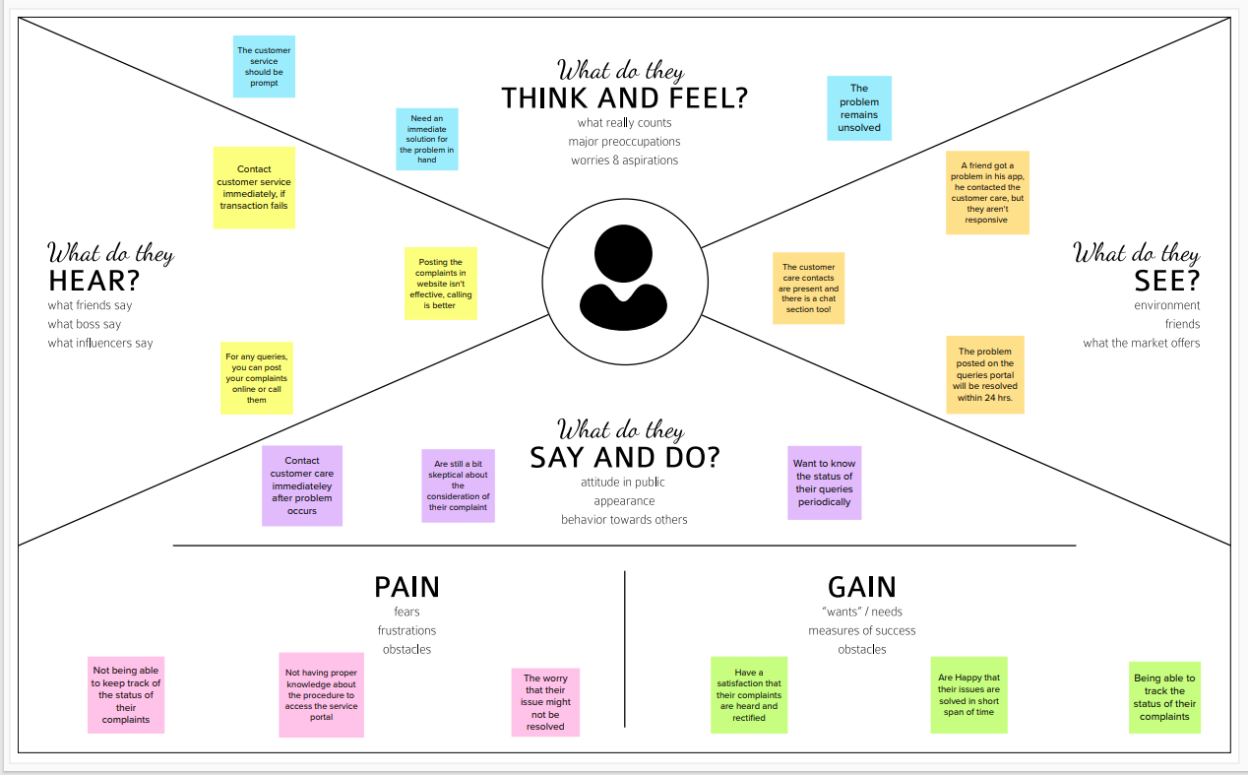
[1] https://iopscience.iop.org/article/10.1088/1757 899X/263/4/042073/pdf
[2] https://turcomat.org/index.php/turkbilmat/article/view/ 8766/6835
[3] V. K. Kandhari and K. D. Mohinani, "GPS based complaint redressal system," 2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS), 2014, pp. 51-56, doi: 10.1109/GHTC-SAS.2014.6967558.
[4] https://www.academia.edu/download/64659324/IRJET-V7I61248.pd

Problem Statement Defintion:

If a customer faces any problem with the product they purchased or in the service provided, it should be addressed and rectified properly in order to get a good user experience. To accomplish this, we need an online customer care registry to process and rectify the complaints in a faster manner. The objective is to create a Cloud based Customer Care Registry where the customer can register their complaints with a detailed description, get assigned with an agent to resolve the complaint and able to track the status of the complaint.

IDEATION & PROPOSED SOLUTION

Empathy Map:



Ideation and Brainstorming:

To create a Cloud based Customer Care Registry Web application where the customers can register their complaints with a detailed description, get assigned with an agent to resolve the complaint and are able to track the status of their complaint.

Ideas:

1. Acknowledge customer's complaint using digital signature.

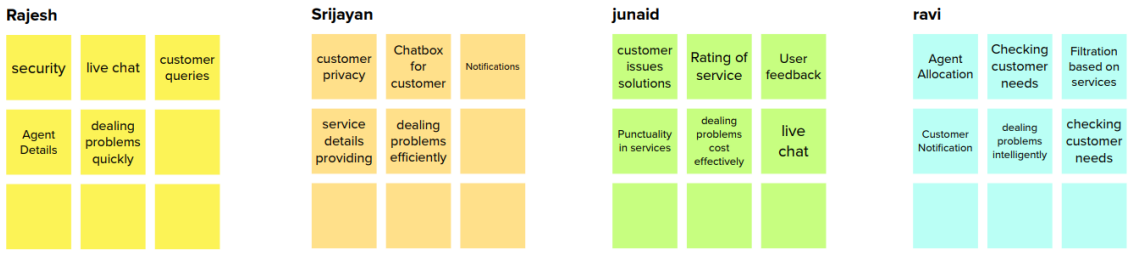
2. Allowing customer to upload proofs such as images or videos or any relevant documents.

3. Add captcha verification to prevent spam bots that attempt DoS (Denial of Service) attacks

Brainstorm solo

Have each participant begin in the "solo brainstorm space" by silently brainstorming ideas and placing them into the template. This "silent-storming" avoids group-think and creates an inclusive environment for introverts and extroverts alike. Set a time limit. Encourage people to go for quantity.

10 minutes

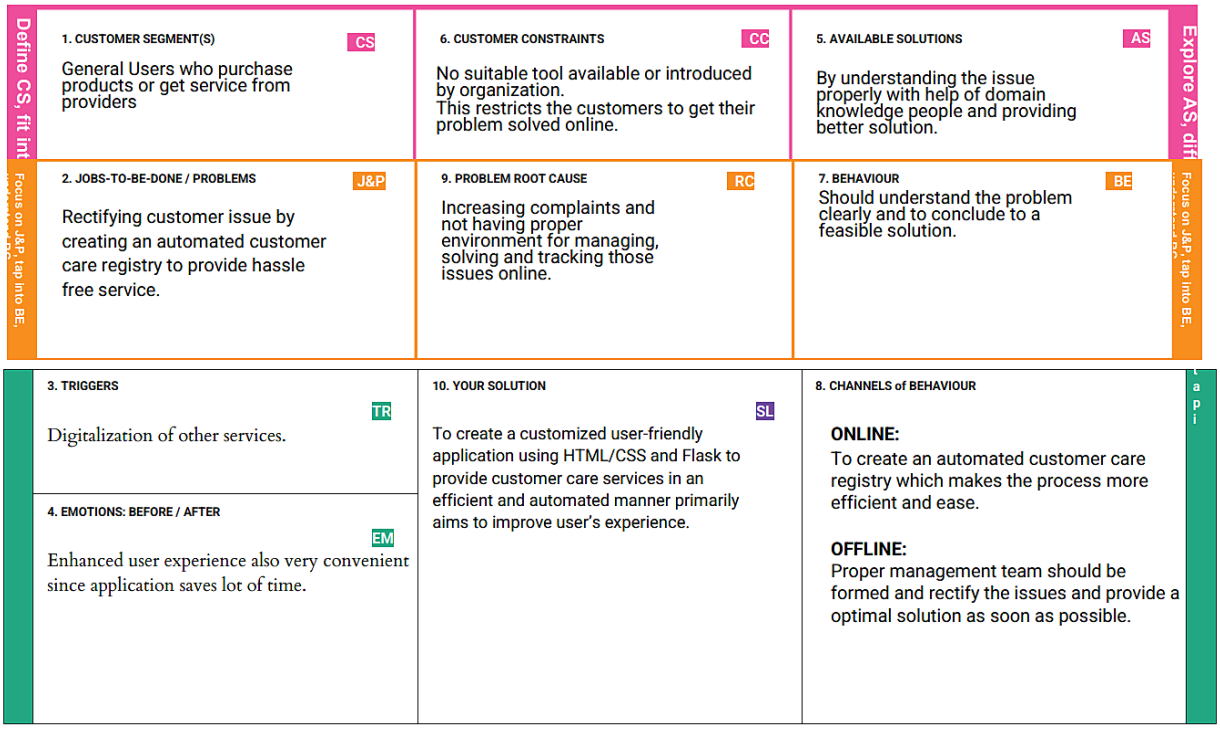


Proposed solution:

To create a customised application which allows customers to raise their issue which will be forwarded to admin who assigns agents to rectify the issue and customer can also keep track of issue to know the current status.

Problem Solution fit:

Team ID: PNT2022TMID35699



REQUIREMENT ANALYSIS

Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login using email and Password
FR-4	Complaint registration	Registering complaint using the query form available in the dashboard
FR-5	Tracking Status	Fetching the status of query using unique id
FR-6	Notification Email	Receiving email from the executive

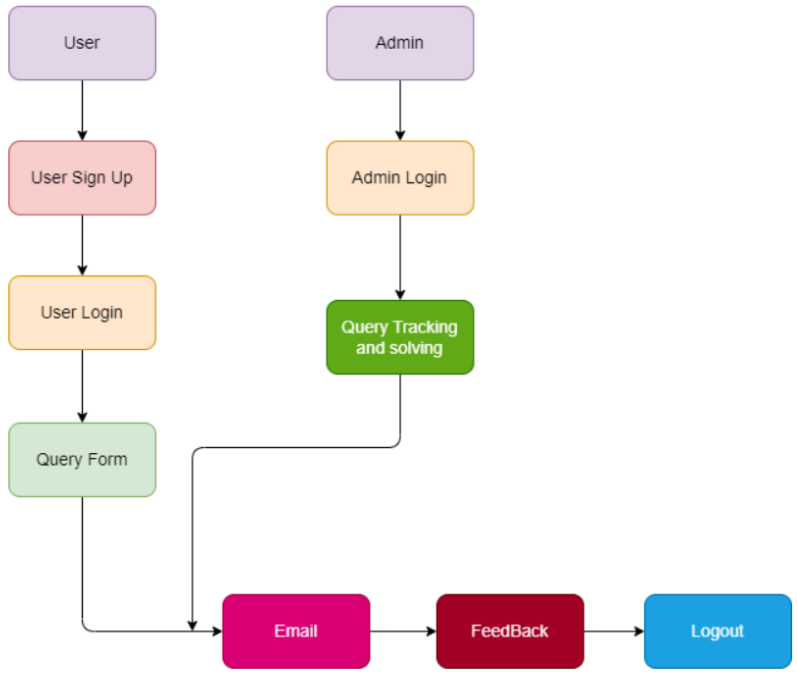
Non Functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application must be easy to use
NFR-2	**Security**	Encryption standards must be used in database
NFR-3	**Reliability**	Queries must be regularly tracked
NFR-4	**Performance**	It must be a low-latency application
NFR-5	**Availability**	Must have multiple servers to avoid traffic
NFR-6	**Scalability**	The data traffic in the network must be predictable.

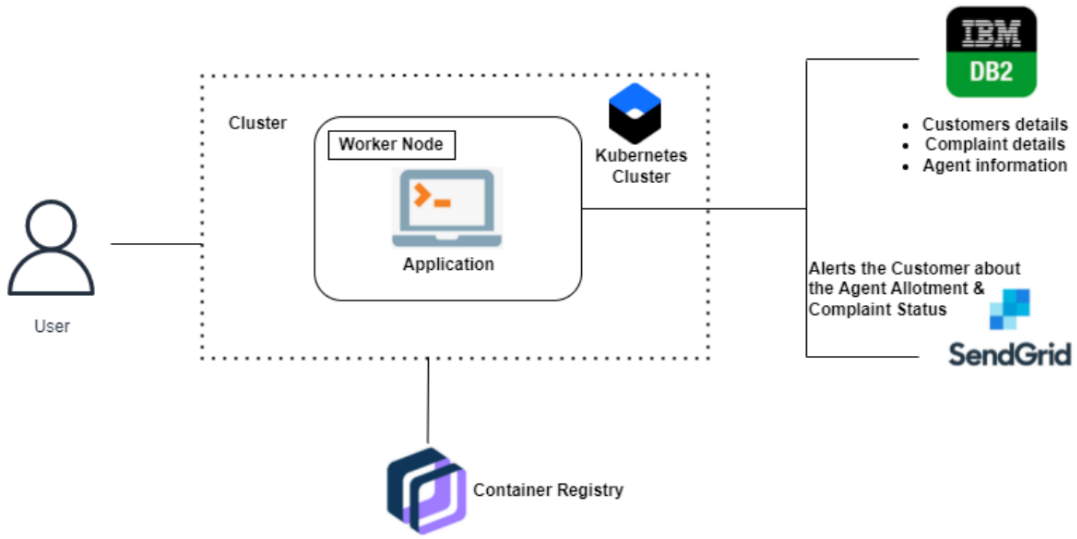
PROJECT DESIGN

Dataflow Diagram:

Data Flow Diagram:



Solution & 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 user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Mobile User)	Verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Laptop User)	Cross-platform authentication.	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Laptop User)	Cross-platform authentication.	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
Customer (Web user)	Login	USN-5	As a user, I can log into the application by entering email & password	I can login using email and password	High	Sprint-1
Customer Care Executive	Solving Queries	USN-6	As an executive, I can check the pending queries, solve them and notify the user	I can solve the queries and notify the user	High	Sprint-1
Administrator	Tracking, Analysis	USN - 7	As an admin, I can track the status of the queries.	I can track and analyse the queries	Medium	Sprint-2

PROJECT PLANNING & SCHEDULING

Sprint 1:

- Registraion page created
- Loin page created

Sprint 2:

- Database integration done with IBM Db2
- Tables created for storing users' data
- Validation for user registration/login completed

Sprint 3:

- Created facility for an individual user to post a complaint.
- Created separate table for complaints in Database
- The complaints along with other relevant data (id) will be stored in database.

Sprint 4:

- Facility is provided for admin to assign complaints to staff (agent).
- Separated table is created to map the agent to the complaint assigned for that particular staff.
- The staff can resolve the issue stated and can provide essential feedback to the client.
- The status of the complaint can be tracked by the client.

Sprint Delivery Schedule:

Sprint 1 -> 24 - 29 October 2022

Sprint 2 -> 31 October – 5 November 2022

Sprint 3 -> 7 – 12 November 2022

Sprint 4 -> 14 – 19 November 2022

CODING & SOLUTIONING:

Feature 1 (Posting complaints):

The client can post their complaints in the web application to have them addressed by the staff. The complaint consists of two sections: The 'subject' and the 'content'. The subject has the brief description about the complaint under consideration and the content section has the detailed explanation about the problem. As soon as a complaint is posted, it is sent to the admin for consideration. The admin verifies if it is a valid complaint and assigns it to an agent/staff for resolving.

Feature 2 (Agent/staff sending feedback to the client):

In some cases, the staff may require extra information about the situation to resolve the query, or they would also want to notify the client about the time they need to resolve the query. To serve those purposes, we have added a feature wherein the staff can leave a message for the client while attempting to resolve the query, so that the client knows that his complaint has been acknowledged and is being processed.

The code for the above features is present in our Github repository:

https://github.com/IBM-EPBL/IBM-Project-5044-1658746471

Database Schemas:

Schema of ‘users_table’ table:

NAME	COLTYPE	LENGTH
EMAIL	VARCHAR	100
MOBILE	VARCHAR	15
NAME	VARCHAR	100
PASSWORD	VARCHAR	100
STATUS	INTEGER	4
USER_ID	INTEGER	4

Schema of ‘complaints’ table:

NAME	COLTYPE	LENGTH
AGENT_FEEDBACK	LONGVAR	32700
AGENT_ID	INTEGER	4
COMP_ID	INTEGER	4
CONTENT	LONGVAR	32700
DATE	DATE	4
STATUS	BOOLEAN	1
SUBJECT	VARCHAR	150
USER_ID	INTEGER	4

Schema of ‘tasks’ table:

NAME	COLTYPE	LENGTH
AGENT_ID	INTEGER	4
COMP_ID	INTEGER	4

TESTING

Testing:

Test Case Id	Component	Test Scenario	Expected Result	Actual Result	Status	Bug Id
Resgister_TC_001	Registration page	Verify if the user is able to see the registration page	The website should appear as soon as the route is entered in	Working as expected	PASS	

			url bar			
Register_TC_002	Registration page	Verify if the entered email is valid	Error message should popup when wrong email format is used, otherwise it should not.	Working as expected	PASS	
Register_TC_003	Registration page	Verify if it is a valid phone number	Error message should popup when an invalid phone number is entered, otherwise it should not	Working as expected	PASS	
Register_TC_004	Registration page	Verify if all the details required are entered before submitting	An error popup should appear if any of the details in left blank	Working as expected	PASS	
Register_TC_005	Registration page	Verify if the registration is successful	The user data must be stored in the database if no faults are found	Working as expected	PASS	

Login_TC_001	Login page	Verify if the login page exists	The login page should appear when the route is entered in the url bar	Working as expected	PASS	
Login_TC_002	Login page	Verify if the user exists	An error popup should appear of the user doesn't exist in the database	Working as expected	PASS	
Login_TC_003	Login page	Verify if the password is correct	An error should popup if the password is incorrect	Working as expected	PASS	
Login_TC_004	Login page	Verify if the login is successful	The user must be redirected to the homepage if the login is successful	Working as expected	PASS	
Homepage_TC_001	Client Homepage	Verify if the client can write a complaint	The client must be able to enter text in the columns specified for the 'subject' and 'complaint'	Working as expected	PASS	

Homepage_TC_002	Client Homepage	Verify if the complaint is valid	An error should popup if the complaint section is left empty	Working as expected	PASS	
Homepage_TC_003	Client Homepage	Verify if the complaint is registered	The complain must be stored in the database if it is devoid of any faults	Working as expected	PASS	
Homepage_TC_004	Client Homepage	Verify that the status of the complain is displayed to the user	The client must be able to view the list of all his complaints along with their status.	Working as expected	PASS	
Admin_TC_001	Login Page	Verify that a user is identified as admin when logging in.	The status value of the user in the database must be checked and the admin should be redirected to the admin homepage	Working as expected	PASS	
Admin_TC_002	Admin Homepage	Verify if the admin can view all the complaints	All the complaints and their status must be visible to	Working as expected	PASS	

			the admin			
Admin_TC_003	Admin Homepage	Verify if the admin can assign tasks	The admin must be able to select the appropriate staff using the dropdown box for every complaint.	Working as expected	PASS	
Staff_TC_001	Login page	Verify that the user is identified as a staff when logging in.	The status value of the user in the database must be checked and the admin should be redirected to the admin homepage	Working as expected	PASS	
Staff_TC_002	Staff Homepage	Verify that the staff is able to view all the issues that are assigned to him	The complaints assigned to him along with other related details are displayed in a list	Working as expected	PASS	

Staff_TC_003	Staff Homepage	Verify that the message feature is working	A message box opens and the staff is able to enter his feedback about the issue to the client.	Working as expected	PASS	
Staff_TC_004	Staff Homepage	Verify that the status button is toggling	When the staff has resolved the issue, he must be able to mark the complaint as 'completed'	Working as expected	PASS	

User Acceptance Testing:

1. Purpose of Document

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

2. Defect Analysis

This reportshows the numberof 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	10	5	2	3	21
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	6	20	39
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	80

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Login/Register	7	0	0	7
Client Homepage	6	0	0	6
Security	2	0	0	2

Database	3	0	0	3
Staff Homepage	4	0	0	4
Admin Homepage	4	0	0	4
Version Control	2	0	0	2

ADVANTAGES & DISADVANTAGES

Advantages:

- Clients can post their complaints easily from their own place without having to go in person
- Clients can track status of their complaints
- The staff can send the feedback about the complaints or communicate any other information to the client through the message box feature
- It is easy to keep a digital record of all the complaints posted and resolved

Disadvantages:

- The clients who can't afford a mobile phone and a stable internet connection can't access this service
- Since the communication is through text messages misinterpretation of information is possible.

CONCLUSION

We have implemented a Customer care service application using Reactjs as frontend and Python Flask module as the backend. We have implemented all the features that are expected. This application was containerized using Docker and has been hosted online using IBM cloud service. Proper testing has been performed to ensure that the application works flawlessly in all the possible

scenarios.

FUTURE SCOPE

- An AI algorithm can be deployed to predict whether the complaint posted is a genuine complaint or a spam message.
- Blockchain can be deployed to keep track of the status of the complaints and to store information in a more secure manner.
- Credit points can be awarded to the staff based on how quick they are in resolving the queries.

APPENDIX:

Github – Project id: IBM-Project-5044-1658746471

Github Link: https://github.com/IBM-EPBL/IBM-Project-5044-1658746471