IBM-NALAIYA THIRAN 2022

Loyola - ICAM College of Engineering and Technology

News Tracker Application

Team ID : PNT2022TMID27467

Team Members

Jesinthan J

Anisha Immaculate A

Devi Priya P

Crystal Darling B

FACULTY MENTOR : Mary Virgil Nithya S

INDUSTRY MENTOR : Sai Priya

CONTENTS

INTRODUCTION

Project Overview

Mobile app ecosystems are transforming patterns of news consumption. As news is increasingly accessed on smartphones and tablets, the need for personalizing news app interactions is apparent. As our lives are very busy these days, we often feel we need more than 24 hrs a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day. Only read what you feel is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds, etc.… as on official websites.

Purpose

As the world's technology is rapidly growing we have fast connections and networks to instantly connect to other people. Day to day use in mobile, tablets and laptop is increasing, most of the people already have these facilities. In this fast and information oriented world we need to stay updated with every incident and news too. The main focus of this application is to connect news articles from all around the world and deliver it to users as fast as possible in the best visualized way.

LITERATURE SURVEY

Existing problem

As our lives are very busy these days, we often feel we need more than 24 hrs a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help.In this fast and information oriented world we need to stay updated with every incident and news too.

References

Several experiments have been carried out over the years by different groups of researchers. Here are some of the following groups:

[1] Exploring mobile news reading interactions for news app personalisation,Marios Constantinides, John Dowell, David Johnson, Sylvain Malacria ,July 2018
Report a series of studies addressing key issues in the development of adaptive news app interfaces.

[2] Topic Detection and Tracking in News Articles,Sagar Patel Nehal Patel Sandeep Patel March 2017
A system is developed to detect and track topics within news articles. Agglomerative clustering has been used successfully for topic detection. Used VSM model for finding similarity between texts

[3] An Improved Method for Multi-Lingual News Feed Application,Regonda Nagaraju, Mohammed Farhan Pasha, Mohammed Abdul Majeed, AdapaSujith October 2019
The news will be fetched and played based on the country's national language & the news is categorized into 7 different categories. This application also supports. translation and the news can be translated into any language.

[4] Tracking Digitally Consumer News,Martijn Kleppe & Irene Costera MeijerMay 2015
Monitoring the behavior of online news users in real-time via tools such as Google Analytics and Chartbeat.

[5] Android News App,Brijesh Joshi Nehal Patel ,March 2018
"Newsapi" provides API that.mkts (JavaScript Object Notation) metadata for headlines and articles. Android structure provides. capability with frameworks, libraries and APIs, with the help of it. we can provide better user.experience and combine this. data at one place while maintaining integrity of its.

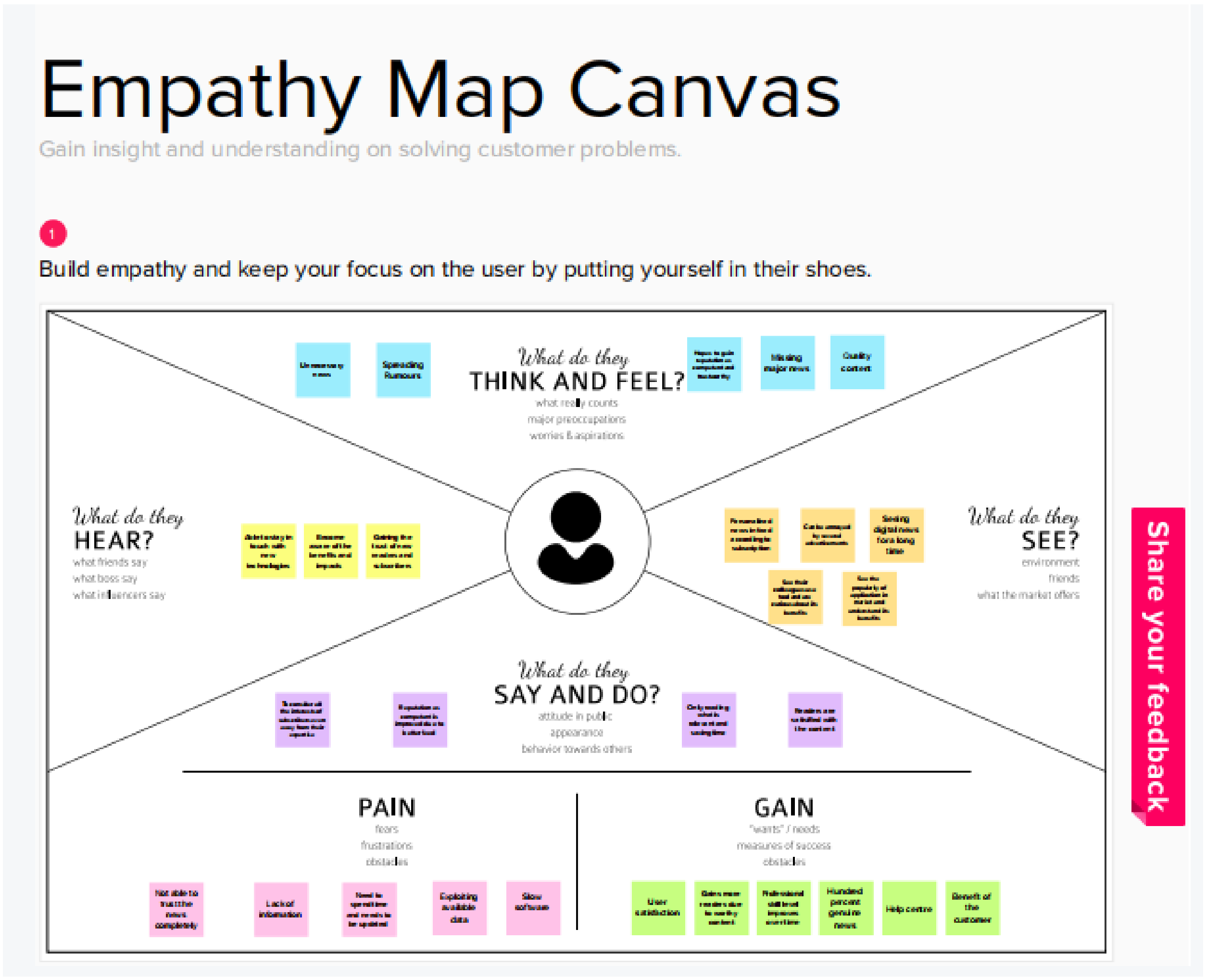
[6] Research on Development Strategy of News App under the Background of artificial Intelligence,Wei Guo1 , Bo Zhang2
Artificial Intelligence and Machine learning is used to extract,sort and transfer large amounts of information

Problem Statement Definition

The main objective of the project is to provide people a handy android application Through which people can access all types of news and information.Through This Application,any user can gain technical knowledge the world and it surrounding With just one click ahead.User Does not have to visit multiple sites for different related information.All information was going to be in One place. Many people generally get the redundancy in information.Sometimes,people Even spread fake news,which circulates and spread more like a disease of false Information in WhatsApp and other social media.Various myths are also likely to Spread as soon as possible than good to the people.This App while cross-checks the redundancy in the information along with the false and Misleading information,which later results in panic in the people.

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



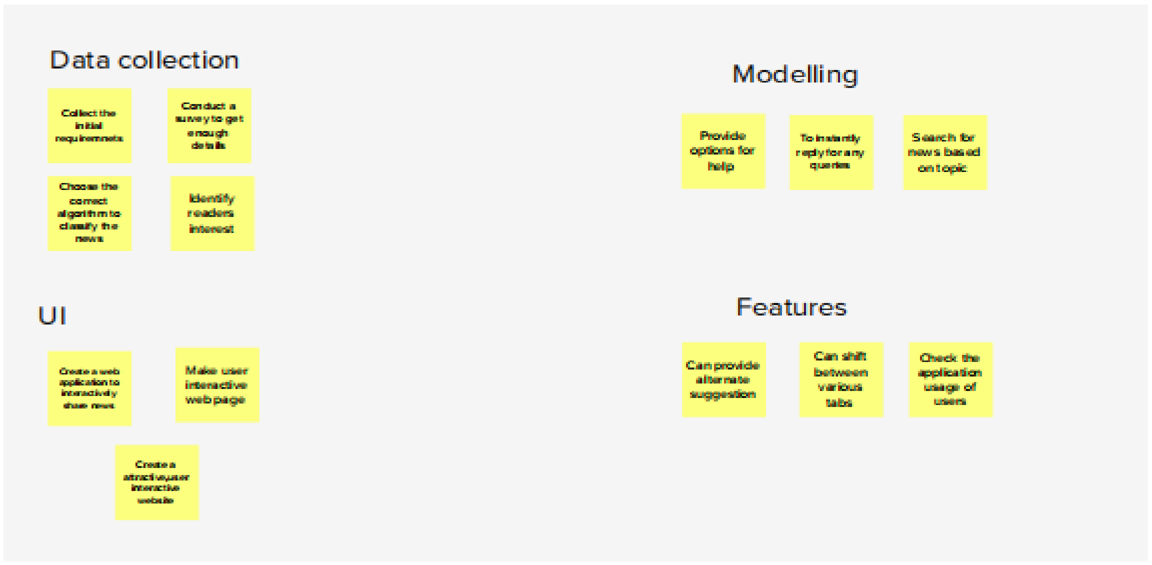
Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

Ideation & Brainstorming

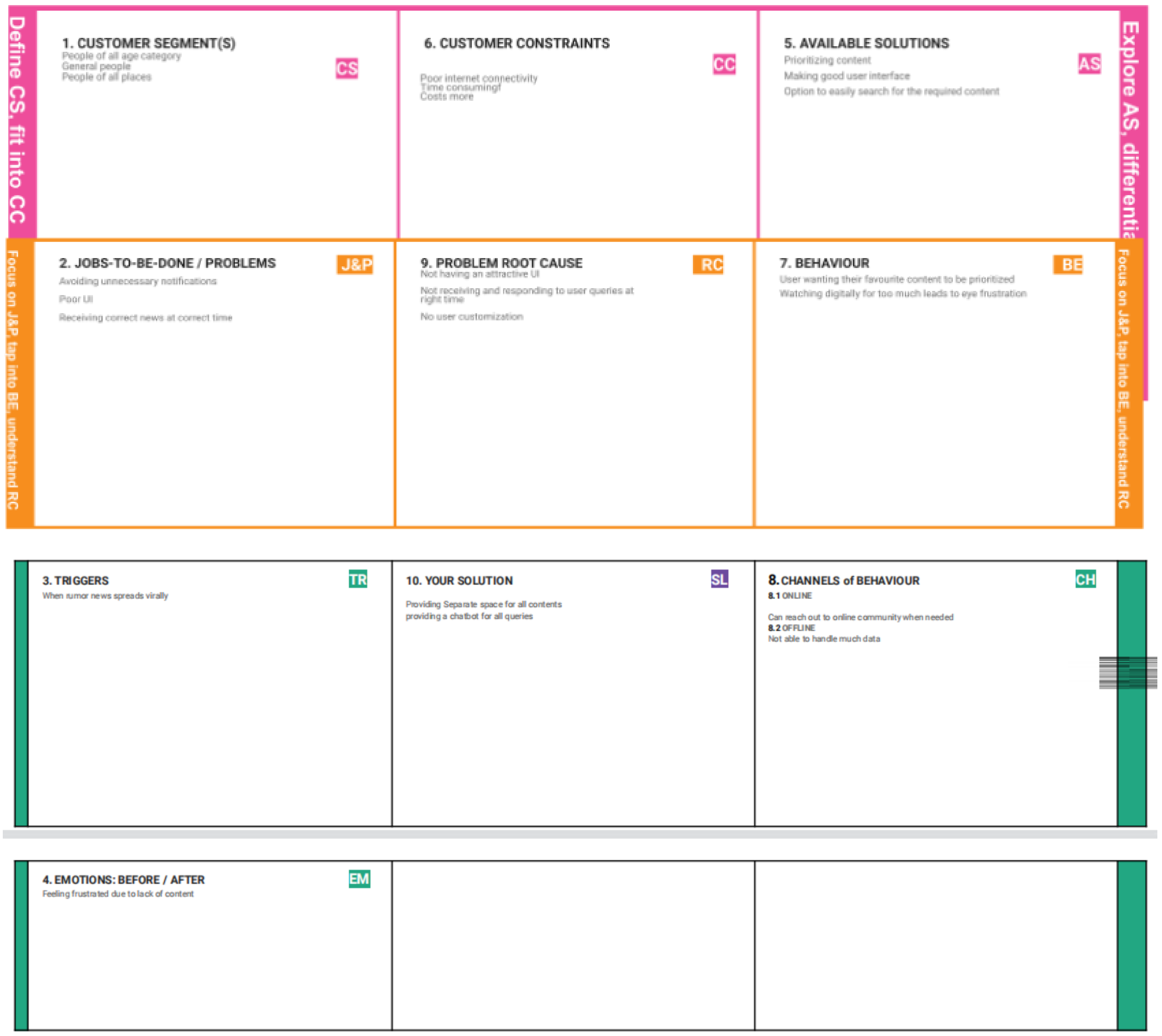

Data collection
Collect the initial requiremnets
Conduct a survey to get enough details
Choose the correct algorithm to classify the news
Identify readers interest
UI
Create a web application to interactively share news
Make user interactive web page
Create a attractive,user interactive website
Modelling
Provide options for help
To instantly reply for any queries
Search for news based on topic
Features
Can provide alternate suggestion
Can shift between various tabs
Check the application usage of users

Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	As our lives are very busy these days, we often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help.
2.	Idea / Solution description	In order to avoid this you could just say what market news you're interested in and get a quick peek for the day. Only read what you feel is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds, etc.… as on official websites.
3.	Novelty / Uniqueness	Has a very unique user interface and makes users search for their value content very easily. It prevents the popping of unnecessary ads
4.	Social Impact / Customer Satisfaction	Since we are using the News API, the news will be reliable and accurate. The customer can be aware of recent happenings. Users can get the required news at the right time easily without any delays

5.	Business Model (Revenue Model)	The features include seeing stories from the user’s favorite publishers, regional news, few regional languages. The default language will be English.
6.	Scalability of the Solution	This can handle multiple users at a time. The user will go through a seamless experience and it enables them to view the news according to their interests and choices. Users from all age categories can use the application and the news can also be filtered according to their age. No internet lag will be there and the UI will be very friendly

Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

People of all age category
General people
People of all places

6. CUSTOMER CONSTRAINTS CC

Poor internet connectivity
Time consumingf
Costs more

5. AVAILABLE SOLUTIONS AS

Prioritizing content

Making good user interface

Option to easily search for the required content

Explore AS, differentia

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Avoiding unnecessary notifications

Poor UI

Receiving correct news at correct time

9. PROBLEM ROOT CAUSE RC

Not having an attractive UI

Not receiving and responding to user queries at right time

No user customization

7. BEHAVIOUR BE

User wanting their favourite content to be prioritized

Watching digitally for too much leads to eye frustration

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

When rumor news spreads virally

10. YOUR SOLUTION SL

Providing Separate space for all contents
providing a chatbot for all queries

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Can reach out to online community when needed
8.2 OFFLINE
Not able to handle much data

4. EMOTIONS: BEFORE / AFTER EM

Feeling frustrated due to lack of content

REQUIREMENT ANALYSIS

Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through online application Registration through Gmail Registration through website
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through browser directly by entering username and password
FR-4	User Interaction	Done through user interface between client and server View the related news by subscripted or requested page

Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	End users can receive push updates for new content on a site by subscribing to the site's news feed
NFR-2	Security	How well are the system and its data protected against attacks
NFR-3	Reliability	How often does the system experience critical failures? How much time does it take to fix the issue when it arises ?And how is user availability time compared to downtime?

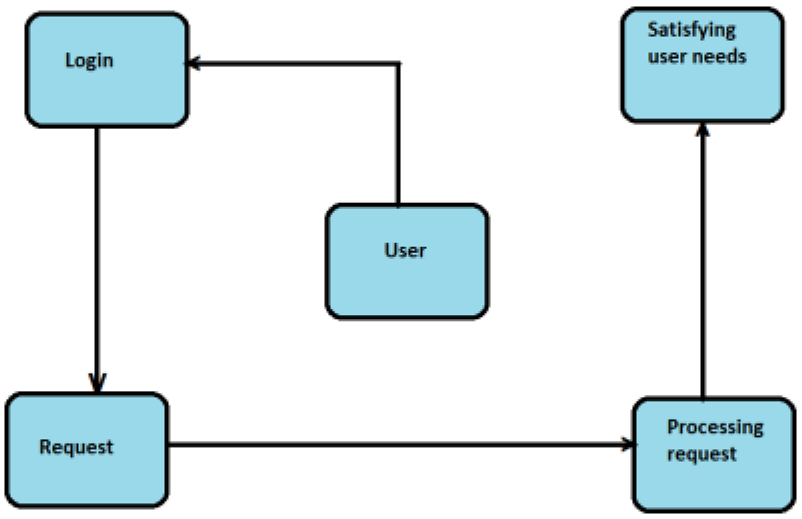
NFR-4	Performance	Performance is the core non-functional requirements no system can do without.It defines how fast a software system or a particular piece of it responds to certain users actions under a certain workload. In most cases, this metric explains how long a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment. But it’s not always like that. Performance requirements may describe background processes invisible to users, e.g. backup. But let’s focus on user-centric performance.
NFR-5	Availability	Availability describes how likely the system is accessible to a user at a given point in time. While it can be expressed as an expected percentage of successful requests, you may also define it as a percentage of time the system is accessible for operation during some time period. For instance, the system may be available 98 percent of the time during a month. Availability is perhaps the most business-critical requirement, but to define it, you also must have estimations for reliability and maintainability.
NFR-6	Scalability	Scalability assesses the highest workloads under which the system will still meet the performance requirements. There are two ways to enable your system scale as the workloads get higher: horizontal and vertical scaling.

PROJECT DESIGN

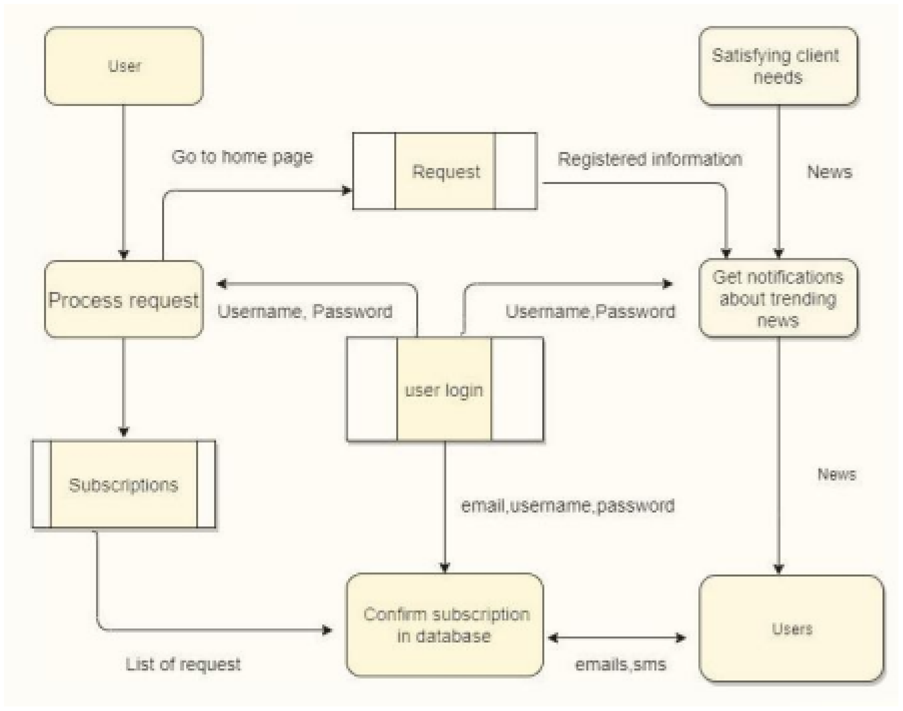
Data Flow Diagram

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.

Simplified



Data flow diagram

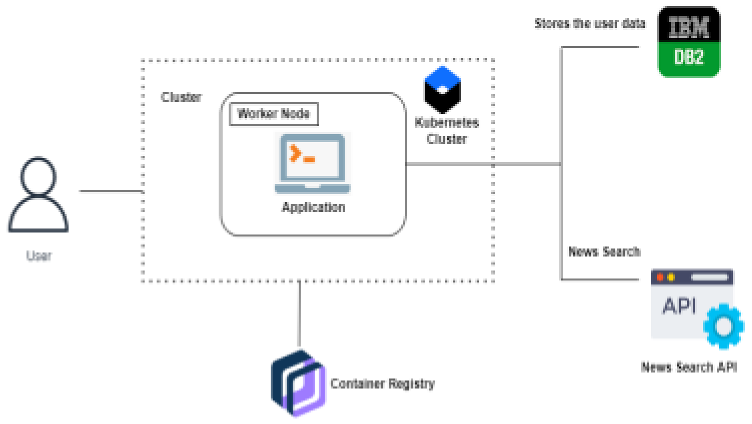


Solution architecture

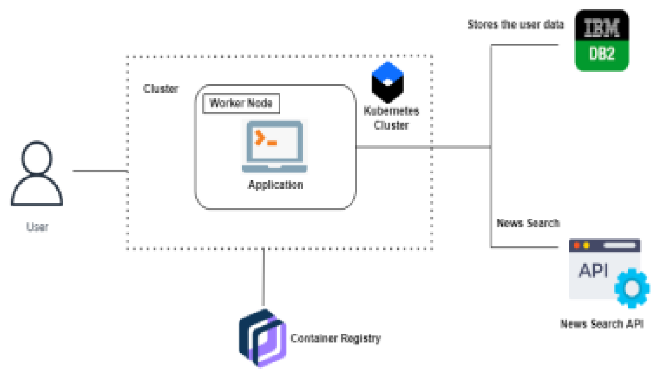
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

1. Find the best tech solution to solve existing business problems
2. Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
3. Define features, development phases, and solution requirements.
4. Provide specifications according to which the solution is defined, managed, and delivered.

Solution architecture diagram



Technical architecture



Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

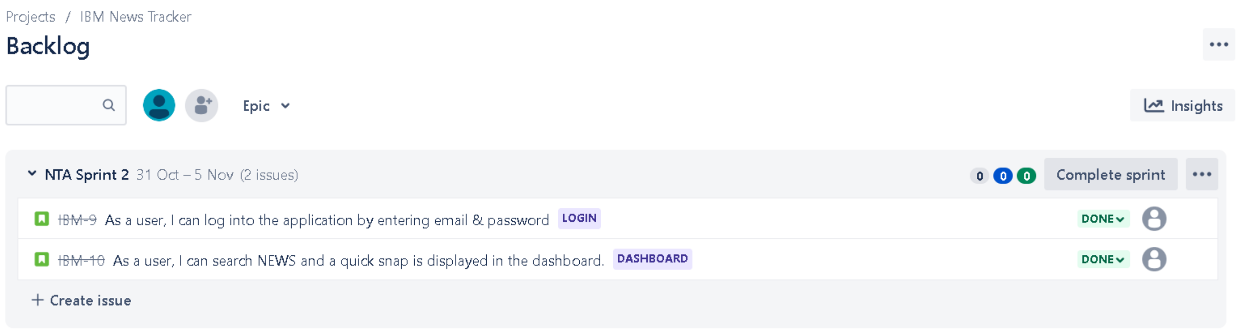
User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Custom er (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
		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
		USN-3	As a user, I can register for the application through Gmail	I can register through Gmail by OTP authenti cation	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can view all types of information through this application	High	Sprint-1

	Dashboard	USN-5	To see their histories about recently viewed, updates for search related news, current progress, feedback		Medium	Sprint-2
Customer (Web user)	Browser	USN-6	Works as an interactive medium between client and server	I can access the resources through browser	High	Sprint-1
Customer Care Executive	Chat bot	USN-7	Rectify the customer’s issues related to account, subscription and customization	Chat bot can resolve simple issues for customers	Low	Sprint-2
Feedback	Feedback Form	USN-8	Getting feedback from customers helps application’s administrator to improve the quality of the application	Customers can tell their opinions	High	Sprint-1
Administrator	Admin module	USN-9	As an admin, I will modify the application as per customer requirements and fix the bugs to give customers a bug free service	I can modify the entire application	High	Sprint-2

PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation



Projects / IBM News Tracker

Backlog

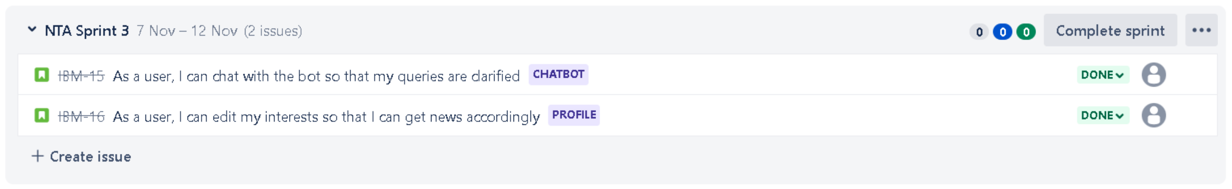
Epic | Insights

NTA Sprint 2 31 Oct – 5 Nov (2 issues) 0 0 0 Complete sprint

~~IBM-9~~ As a user, I can log into the application by entering email & password LOGIN DONE

~~IBM-10~~ As a user, I can search NEWS and a quick snap is displayed in the dashboard. DASHBOARD DONE

+ Create issue

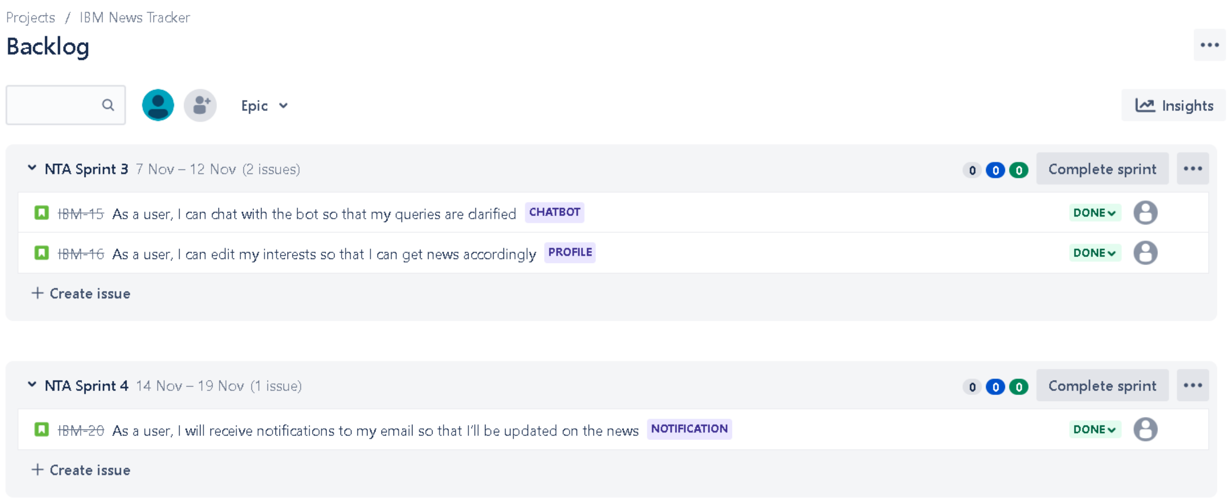


NTA Sprint 3 7 Nov – 12 Nov (2 issues) 0 0 0 Complete sprint

~~IBM-15~~ As a user, I can chat with the bot so that my queries are clarified CHATBOT DONE

~~IBM-16~~ As a user, I can edit my interests so that I can get news accordingly PROFILE DONE

+ Create issue



Projects / IBM News Tracker

Backlog

Epic | Insights

NTA Sprint 3 7 Nov – 12 Nov (2 issues) 0 0 0 Complete sprint

~~IBM-15~~ As a user, I can chat with the bot so that my queries are clarified CHATBOT DONE

~~IBM-16~~ As a user, I can edit my interests so that I can get news accordingly PROFILE DONE

+ Create issue

NTA Sprint 4 14 Nov – 19 Nov (1 issue) 0 0 0 Complete sprint

~~IBM-20~~ As a user, I will receive notifications to my email so that I'll be updated on the news NOTIFICATION DONE

+ Create issue

Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	15	High	Jesinthan,Devi Priya,Anisha, Crystal
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	5	Medium	Jesinthan,Devi Priya,Anisha, Crystal
Sprint-2	Login	USN-3	As a user, I can log into the application by entering email & password	10	High	Jesinthan,Devi Priya,Anisha, Crystal
Sprint-2	Dashboard	USN-4	As a user, I can search NEWS and a quick snap is displayed in the dashboard.	10	High	Jesinthan,Devi Priya,Anisha, Crystal
Sprint-3	Chatbot	USN-5	As a user, I can chat with the bot so that my queries are clarified	10	High	Jesinthan,Devi Priya,Anisha, Crystal

Sprint-3	Profile	USN-6	As a user, I can edit my interests so that I can get news accordingly	10	High	Jesinthan,Devi Priya,Anisha, Crystal
Sprint-4	Notifications	USN-7	As a user, I will receive notifications to my email so that I'll be updated on the news	20	Medium	Jesinthan,Devi Priya,Anisha, Crystal

Project Tracker, Velocity & Burndown Chart:

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	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)

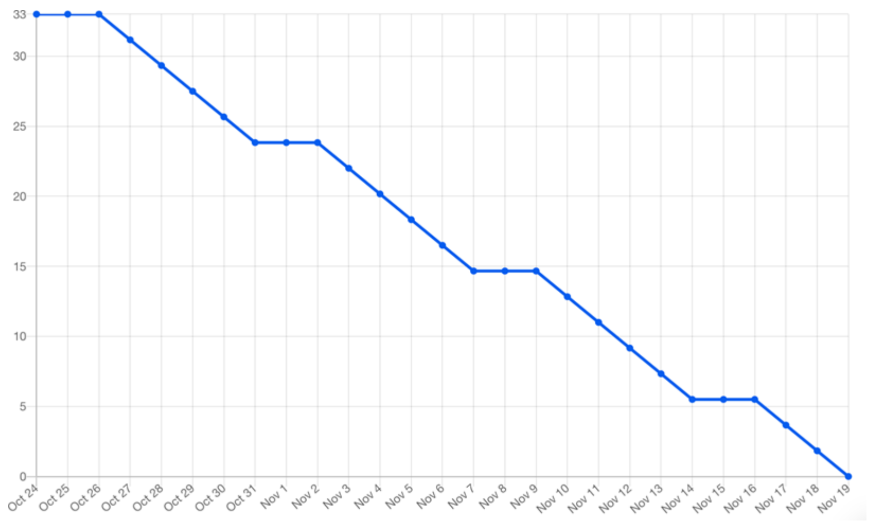
$$AV = \frac{\textit{sprint duration}}{\textit{velocity}} = \frac{20}{10} = 2$$

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)	Average Velocity (AV)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022	3.33
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022	3.33
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022	3.33
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022	3.33

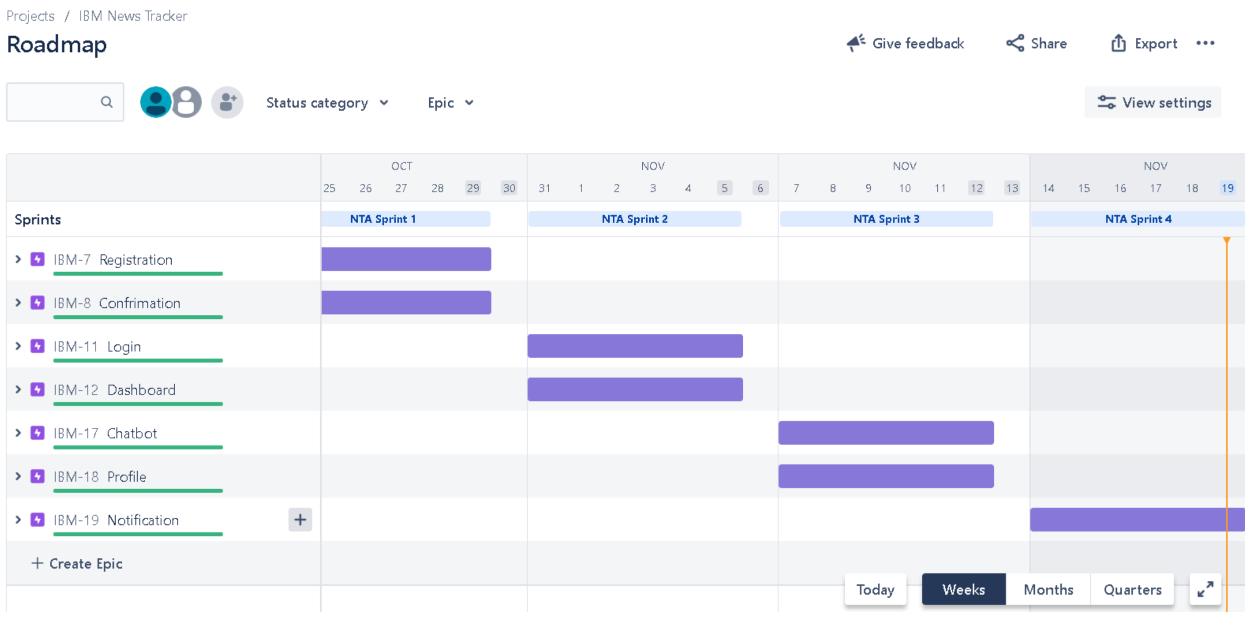
Total no of days = 6+6+6+6= 24
Total Story points = 20+20+20+20= 80
Average Velocity per Sprint = 80/24= 3.33

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



Reports from Jira


Projects / IBM News Tracker

Roadmap

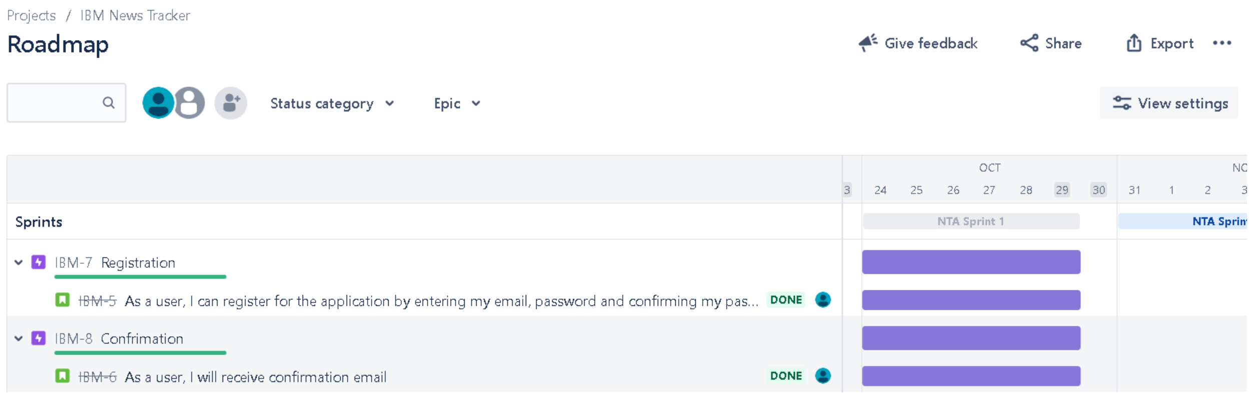
Give feedback Share Export

Status category Epic View settings

Sprints	OCT 25–30	NOV 31–6	NOV 7–13	NOV 14–19
	NTA Sprint 1	NTA Sprint 2	NTA Sprint 3	NTA Sprint 4
IBM-7 Registration	▬			
IBM-8 Confirmation	▬			
IBM-11 Login		▬		
IBM-12 Dashboard		▬		
IBM-17 Chatbot			▬	
IBM-18 Profile			▬	
IBM-19 Notification				▬

+ Create Epic

Today Weeks Months Quarters


Projects / IBM News Tracker

Roadmap

Give feedback Share Export

Status category Epic View settings

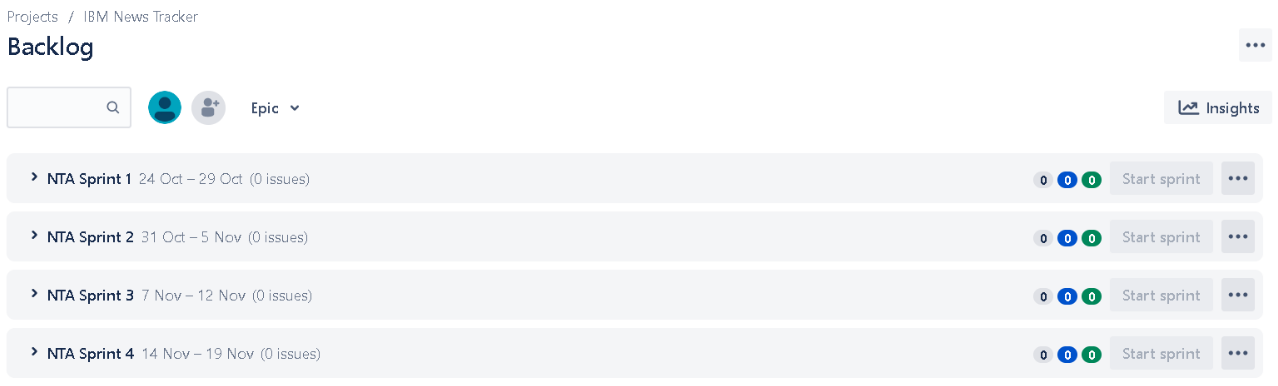
Sprints: NTA Sprint 1 (OCT 24–30), NTA Sprint 2

IBM-7 Registration

~~IBM-5~~ As a user, I can register for the application by entering my email, password and confirming my pas... DONE

IBM-8 Confirmation

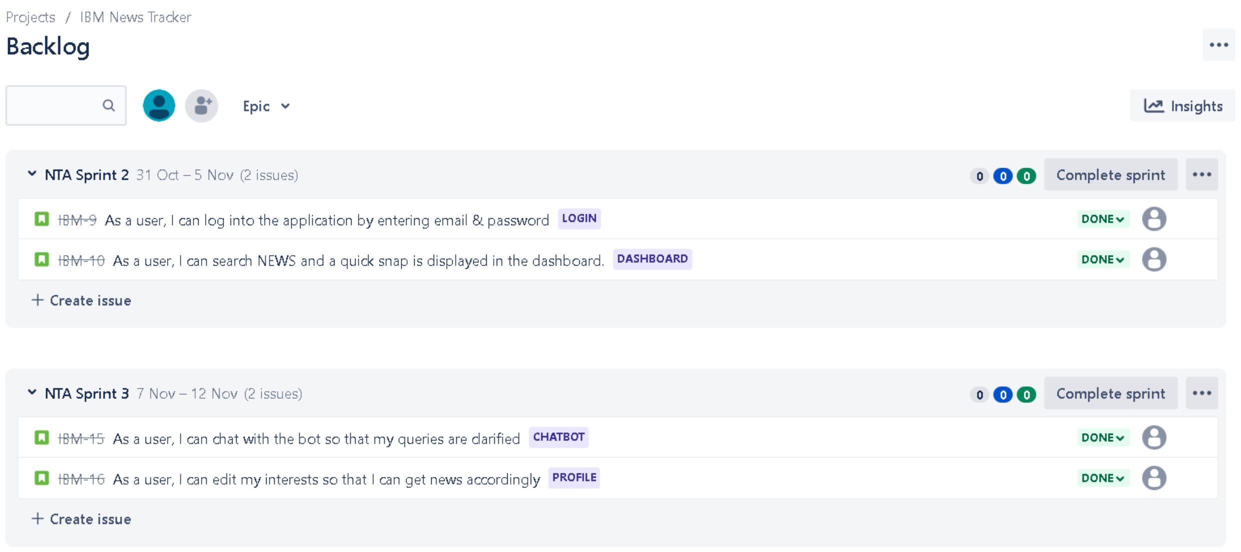
~~IBM-6~~ As a user, I will receive confirmation email DONE


Projects / IBM News Tracker

Backlog

Epic Insights

Sprint	Dates	Issues				
NTA Sprint 1	24 Oct – 29 Oct	(0 issues)	0	0	0	Start sprint
NTA Sprint 2	31 Oct – 5 Nov	(0 issues)	0	0	0	Start sprint
NTA Sprint 3	7 Nov – 12 Nov	(0 issues)	0	0	0	Start sprint
NTA Sprint 4	14 Nov – 19 Nov	(0 issues)	0	0	0	Start sprint


Projects / IBM News Tracker

Backlog

Epic Insights

NTA Sprint 2 31 Oct – 5 Nov (2 issues) 0 0 0 Complete sprint

~~IBM-9~~ As a user, I can log into the application by entering email & password LOGIN DONE

~~IBM-10~~ As a user, I can search NEWS and a quick snap is displayed in the dashboard. DASHBOARD DONE

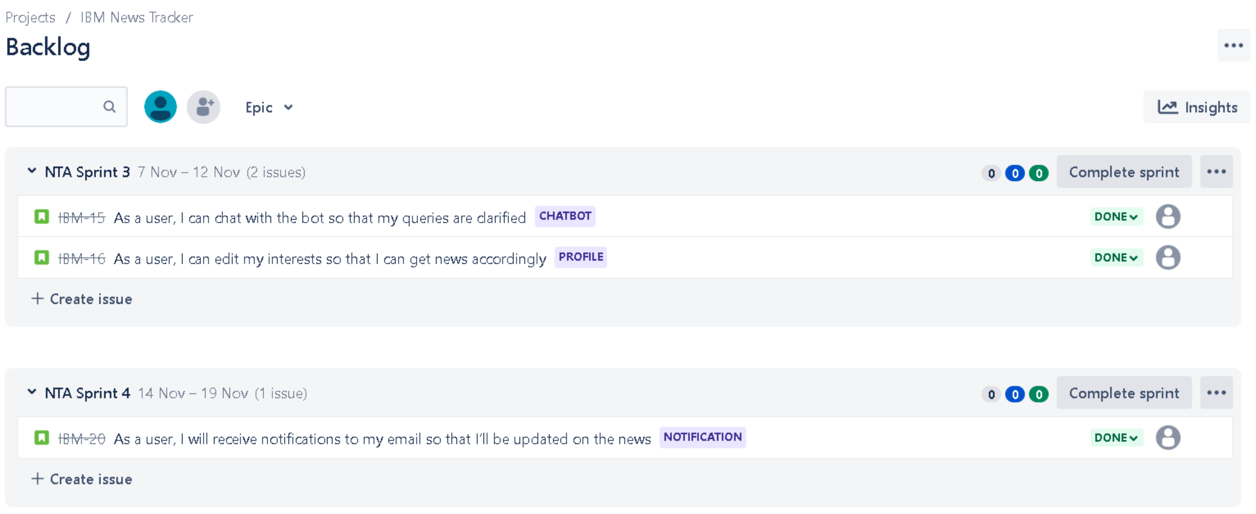
+ Create issue

NTA Sprint 3 7 Nov – 12 Nov (2 issues) 0 0 0 Complete sprint

~~IBM-15~~ As a user, I can chat with the bot so that my queries are clarified CHATBOT DONE

~~IBM-16~~ As a user, I can edit my interests so that I can get news accordingly PROFILE DONE

+ Create issue


Projects / IBM News Tracker

Backlog

Epic Insights

NTA Sprint 3 7 Nov – 12 Nov (2 issues) 0 0 0 Complete sprint

~~IBM-15~~ As a user, I can chat with the bot so that my queries are clarified CHATBOT DONE

~~IBM-16~~ As a user, I can edit my interests so that I can get news accordingly PROFILE DONE

+ Create issue

NTA Sprint 4 14 Nov – 19 Nov (1 issue) 0 0 0 Complete sprint

~~IBM-20~~ As a user, I will receive notifications to my email so that I'll be updated on the news NOTIFICATION DONE

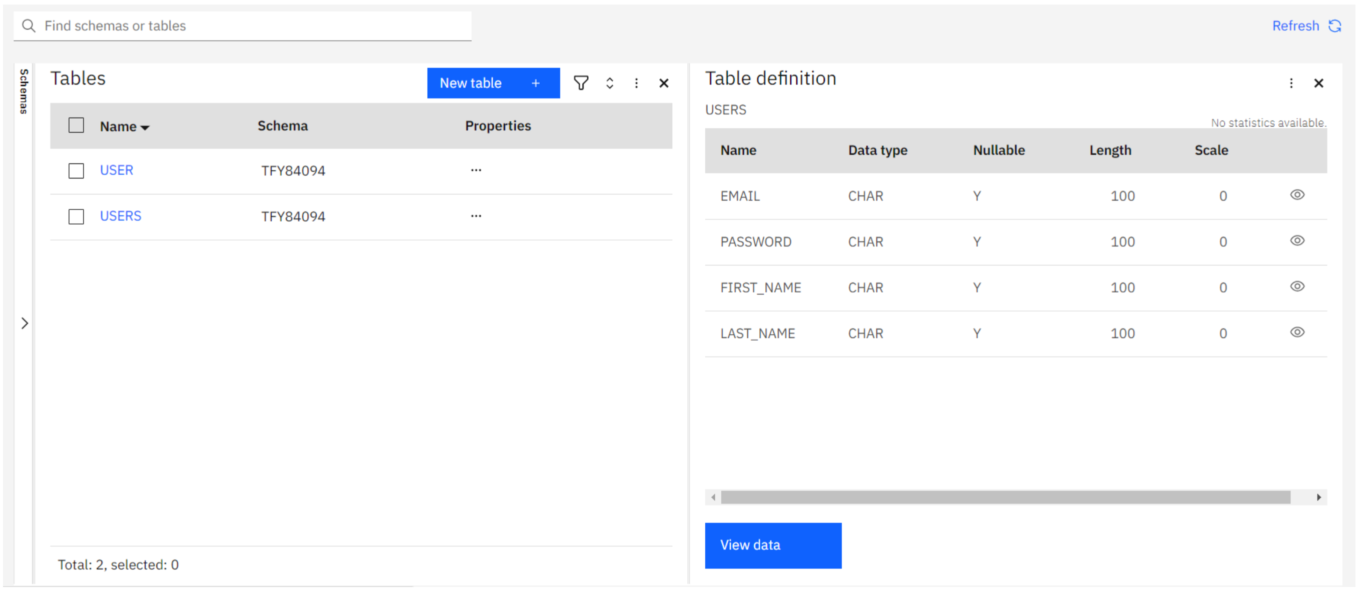
+ Create issue

CODING AND SOLUTIONING

Feature-Chatbot

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "bbb2eec8-ab45-4b81-84c5-844556da314c", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "17bd5ac5-d205-4fc1-9bef-948c171b1aa2", // The ID of
your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
</script>
```

Database Schema



Find schemas or tables

Refresh

Schemas

Tables

New table +

Name	Schema	Properties
USER	TFY84094	···
USERS	TFY84094	···

Total: 2, selected: 0

Table definition

USERS

No statistics available.

Name	Data type	Nullable	Length	Scale
EMAIL	CHAR	Y	100	0
PASSWORD	CHAR	Y	100	0
FIRST_NAME	CHAR	Y	100	0
LAST_NAME	CHAR	Y	100	0

View data

TESTING
Test Cases

Test Case	Feature	Component	Test Scenario	Expected result	Actual Result	Status	Comments	Bug	Executed By
Sign in	Functional	Login page	Verified user can see the sign in page	Visible	Yes, Visible	Pass	Successful	-	Anisha
Sign up	Functional	Login page	Verified user has the option to sign up	Visible	Yes, Visible	Pass	Successful	-	Jesinthan
Fetch news	Functional	Home page	Verified user can get the news	News will be fed to app	404 error	Fail	Unsuccessful	App integration problem	Jesinthan
Types of news available in fetch news page	Functional	Fetch news	Types of news available	Weather, Sport, Economy	Hover button is shown	Yes	Successful	-	Crystal

User Acceptance Testing

Purpose of this Document

The purpose of this document is to briefly explain the test coverage and open issues of the news tracker application project at the time of the release to User Acceptance Testing (UAT).

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	7	4	2	1	14
Duplicate	3	1	1	1	6
External	2	1	1	0	4
Fixed	9	4	3	3	19
Not Reproduced	2	2	1	1	6
Skipped	3	0	1	0	4
Won't Fix	1	2	0	1	4
Totals	27	14	9	7	

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	46	0	0	46
Security	4	0	0	4
Outsource Shipping	2	0	0	2
Exception Reporting	5	0	0	5
Final Report Output	6	0	0	6
Version Control	2	0	0	2

RESULTS

News tracker application using cloud is developed and executed at the level of completed progress .

ADVANTAGES AND DISADVANTAGES

Advantages:

- Get personalized news based on interest
- Easily portable and accessible
- Get interesting news instantly
- Good user experience
- Can help the users to share news in any social media

Disadvantages:

- Fake news might mislead people
- Limited by time
- Inefficiencies in detecting reliable vs unreliable news

CONCLUSION

The idea and motive behind this project is that people with the same interest can interact with each other. However, they can even share more information on the topic. This app cross-checks the redundancy in the information along with the false and misleading information,which later results in panic in the people.

FUTURE SCOPE

Offline Reading can be improved and explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps.

APPENDIX
Source code

App.py

```python
from flask import  Flask,render_template, request, redirect, url_for,
session
import ibm_db
import os
from dotenv import load_dotenv
import pandas as pd
import smtplib
from email.message import EmailMessage
import requests
import json
from flask_session import Session
from apscheduler.schedulers.background import BackgroundScheduler
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
from email.mime.application import MIMEApplication
from email.mime.multipart import MIMEMultipart
from jinja2 import Environment




app = Flask(__name__)
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)

# Loading up the values
load_dotenv()
#DB Creds
database = os.environ.get("DATABASE")
db_hostname = os.environ.get("HOSTNAME")
db_port = os.environ.get("PORT")
db_uid = os.environ.get("UID")
db_pwd = os.environ.get("PWD")
```

```
email_pwd = os.environ.get("email_password")



# Database Connection
try:
    conn = ibm_db.connect(

f'DATABASE={database};HOSTNAME={db_hostname};PORT={db_port};SECURITY=SSL;S
SLServerCertificate=DigiCertGlobalRootCA.crt;UID={db_uid};PWD={db_pwd}',
'', '')
    print("Connected to database: ", conn)
except Exception as e:
    print (e)


def message(subject="Python Notification",
            text="", img=None, attachment=None):

    # build message contents
    msg = MIMEMultipart()

    f = open("./templates/notifications.html", "r", errors="ignore")
    html_content = f.read()

    html_contentt = Environment().from_string(
        html_content).render(msg=text)

    # Add Subject
    msg['Subject'] = subject

    # Add text contents
    msg.attach(MIMEText(html_contentt, 'html'))
    return msg


def mail():

    # initialize connection to our email server,
    # we will use gmail here
    smtp = smtplib.SMTP('smtp.gmail.com', 587)
    smtp.ehlo()
    smtp.starttls()

    # Login with your email and password
    smtp.login("jesinthan0703@gmail.com", email_pwd)

    # url = "https://newscatcher.p.rapidapi.com/v1/search_free"
    # querystring = {"q":"Russia","lang":"en","media":"True"}
    # headers = {
    #     "X-RapidAPI-Key":
"78394ce5f7msh148449ce3836679p1239b2jsnb6fb656b52e5",
    #     "X-RapidAPI-Host": "newscatcher.p.rapidapi.com"
    # }
    # response = requests.request("GET", url, headers=headers,
params=querystring)
    # json_object = json.loads(response.text)

    f = open("sample.json", "r")
    news_data = f.read()
    json_object = json.loads(news_data)

    data = json_object["articles"]

    # Call the message function
    msg = message("Exciting news today!", data[:10])

    sql = "SELECT email FROM users"
    stmt = ibm_db.prepare(conn, sql)
    # ibm_db.bind_param(stmt, 1, "shirleychristabel.23it@licet.ac.in")
    ibm_db.execute(stmt)
    users = []
    # List of emails
    while ibm_db.fetch_row(stmt) != False:
        users.append(ibm_db.result(stmt, 0).strip())

    print(users)
    smtp.sendmail(from_addr="jesinthan0703@gmail.com",
                            to_addrs="jesinthan.23cs@licet.ac.in",
msg=msg.as_string())
```

```
    print("Email sent successfully")

    smtp.quit()


sched = BackgroundScheduler(daemon=True)
sched.add_job(mail, 'interval', hours=24)
sched.start()

# Home Route
@app.route('/',methods = ['POST', 'GET'])
def home():

    def send_mail(r_mail, content):
        s_mail = "jesinthan0703@gmail.com"
        s_pass = email_pwd
        msg=EmailMessage()
        msg['Subject'] = f"Registration Successful"
        msg['From'] = s_mail
        msg['To'] = r_mail
        msg.set_content(content,subtype="html")

        server = smtplib.SMTP_SSL("smtp.gmail.com",465)

        try:
            server.login(s_mail,s_pass)
            print("Logged In Successfully")
            server.send_message(msg)
            print("Mail Sent")
            server.quit()
        except Exception as e:
            print(e)

    if request.method == 'POST':

        first_name = request.form['first_name']
        last_name = request.form['last_name']
        email = request.form['email']
        password = request.form['password']
        confirm_password = request.form['confirm_password']
        print(first_name, last_name, email, password, confirm_password)

        if(password==confirm_password):
            sql = "SELECT * FROM users WHERE email = '"+email+"' "
            print(sql)
            stmt = ibm_db.exec_immediate(conn, sql)

            account = ibm_db.fetch_assoc(stmt)
            print(account)

            if account:
                print("User already exists")
                return render_template('register.html', msg="User already
exists, Please login")
            else:
                print("User does not exist")
                try:
                    insert_query = "INSERT INTO users
VALUES('"+email+"','"+password+"','"+first_name+"','"+last_name+"')"
                    ibm_db.exec_immediate(conn, insert_query)
                    print("You are successfully registered")
                    with open('mail.html', 'r') as f:
                        mail_content= f.read()
                        send_mail(email, mail_content)
                    return render_template('login.html')

                except Exception as e:
                    print(e)


        else:
            print("Password does not match")
            return render_template('register.html', msg="Password does not
match")


    return render_template('register.html',title="Register")

@app.route('/login',methods = ['POST', 'GET'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        print(email,password)

        try:
            sql = "SELECT password FROM users WHERE email = '"+email+"' "
            print(sql)
            stmt = ibm_db.exec_immediate(conn, sql)
            print(stmt)
            pwd = ibm_db.fetch_assoc(stmt)
            print(pwd)
            key = pwd.get('PASSWORD')

            if password==key.strip():
                print("User exists, Logged in successfully")
                session["email"] = email
                print(session["email"])
                return redirect("/dashboard", code=302)
            else:
                print("Password is incorrect")
                return render_template('login.html', msg="Password is
incorrect")
        except Exception as e:
            print(e)
            return render_template('login.html', msg="User does not exist,
Please register")


    return render_template('login.html')

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

    if not session.get("email"):
            return redirect("/login")
    else:
        try:
            #PRODUCTION
            # url = "https://newscatcher.p.rapidapi.com/v1/search_free"
            # querystring = {"q":"Russia","lang":"en","media":"True"}
            # headers = {
            #     "X-RapidAPI-Key":
"78394ce5f7msh148449ce3836679p1239b2jsnb6fb656b52e5",
            #     "X-RapidAPI-Host": "newscatcher.p.rapidapi.com"
            # }
            # response = requests.request("GET", url, headers=headers,
params=querystring)
            # json_object = json.loads(response.text)

            #DEV
            f = open("sample.json", "r")
            news_data = f.read()
            json_object = json.loads(news_data)

            try:
                sql="SELECT * FROM users WHERE email =
'"+session["email"]+"' "
                print(sql)
                stmt = ibm_db.exec_immediate(conn, sql)
                print(stmt)
                user = ibm_db.fetch_assoc(stmt)
                first_name = user.get('FIRST_NAME').strip()
                last_name = user.get('LAST_NAME').strip()
                return
render_template('dashboard.html',students=json_object,
first_name=first_name, last_name=last_name)
            except Exception as e:
                print(e)
                return
render_template('dashboard.html',students=json_object)


        except Exception as e:
            print(e)

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

```
def profile():
    if not session.get("email"):
            return redirect("/login")
    else:
        try:
            sql = "SELECT first_name, last_name FROM users WHERE email =
'"+session["email"]+"' "
            print(sql)
            stmt = ibm_db.exec_immediate(conn, sql)
            print(stmt)
            profile = ibm_db.fetch_assoc(stmt)
            first_name = profile.get('FIRST_NAME').strip()
            last_name = profile.get('LAST_NAME').strip()
            print(first_name, last_name)
            return render_template('profile.html',first_name=first_name,
last_name=last_name, email=session["email"])
        except Exception as e:
            print(e)
    return render_template('profile.html')


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

    session.pop("email", None)

    return render_template('login.html')



if __name__ == "main":
    app. run(debug=True, use_reloader=True)
```

Github and demonstration link

Github:

https://github.com/IBM-EPBL/IBM-Project-37822-1660327116.git

Demonstration Link:

https://drive.google.com/file/d/1LDNQS1X7nql6lyniIqTEk2B1LbO0Mgw2/view?usp=drivesdk