Software Requirements Specification (SRS) Document

Project Number	14
Project Title	Gemini API Integrated With HTML UI
Document	Project Plan
Creation Date	30th January 2024
Created By	Rahul Singal, Harshit Gupta, Divyansh Pandey, Ayush Sahu
Client	Patenti Technology Solutions

Brief problem statement

To develop a web interface for a chatbot utilizing Google's Gemini LLM through an API service.The web application should accept text prompts from users and showcase the corresponding responses generated by Gemini. Additionally, the prompts must be saved in a database, and the entire system,including both the service and the database, should be hosted on AWS using EC2 instance.

System requirements

1) **Gemini integrated Chatbot:**

In this module, the focus will be on comprehending the Gemini API, followed by the development of a Python script. This script is designed to seamlessly interface with the Gemini API, accepting input and generating corresponding outputs. Also the chatbot will follow an NLP based approach ,understanding the context of the prompt to synthesize an output and keep track of conversation history and context. The chatbot should have context filtering to a dataset so that it gives response to a certain set of prompts.

2) **HTML + CSS (Bootstrap):**

The second module concentrates on the aesthetic aspect of the project. It involves the creation of a user interface (UI) through the utilization of HTML ,CSS and JS controllers along with Bootstrap integration. The primary objective is to design a visually appealing and responsive interface.

3) **Django Backend:**

This pivotal phase is dedicated to the establishment of the backend infrastructure of the project. It encompasses the integration of the UI and the ChatBot Python script, ensuring a cohesive and functional project foundation.

4) **SQL Database Linkage:**

The fourth module involves the critical task of linking the project with a MySQL-powered database. This linkage will facilitate the storage of prompts,dates, and region-related data, ensuring a robust and efficient data management system.

5) **Deployment to AWS**:

The final phase centers on the deployment of the entire project onto the AWS cloud services platform using EC2 instance. Special attention will be given to ensuring compatibility with Linux platforms to guarantee seamless functionality.

Users profile

This project is a versatile and widely applicable chatbot platform where users can submit inquiries, and the system generates and presents corresponding outputs.This is a generic and a an easy-to-use chatbot that addresses all users alike without any major specificity . The basic the requirement is that the user should know English.The system administrator plays a vital role in maintaining the platform's integrity, protecting user data, and addressing technical challenges, ultimately enhancing the user experience.

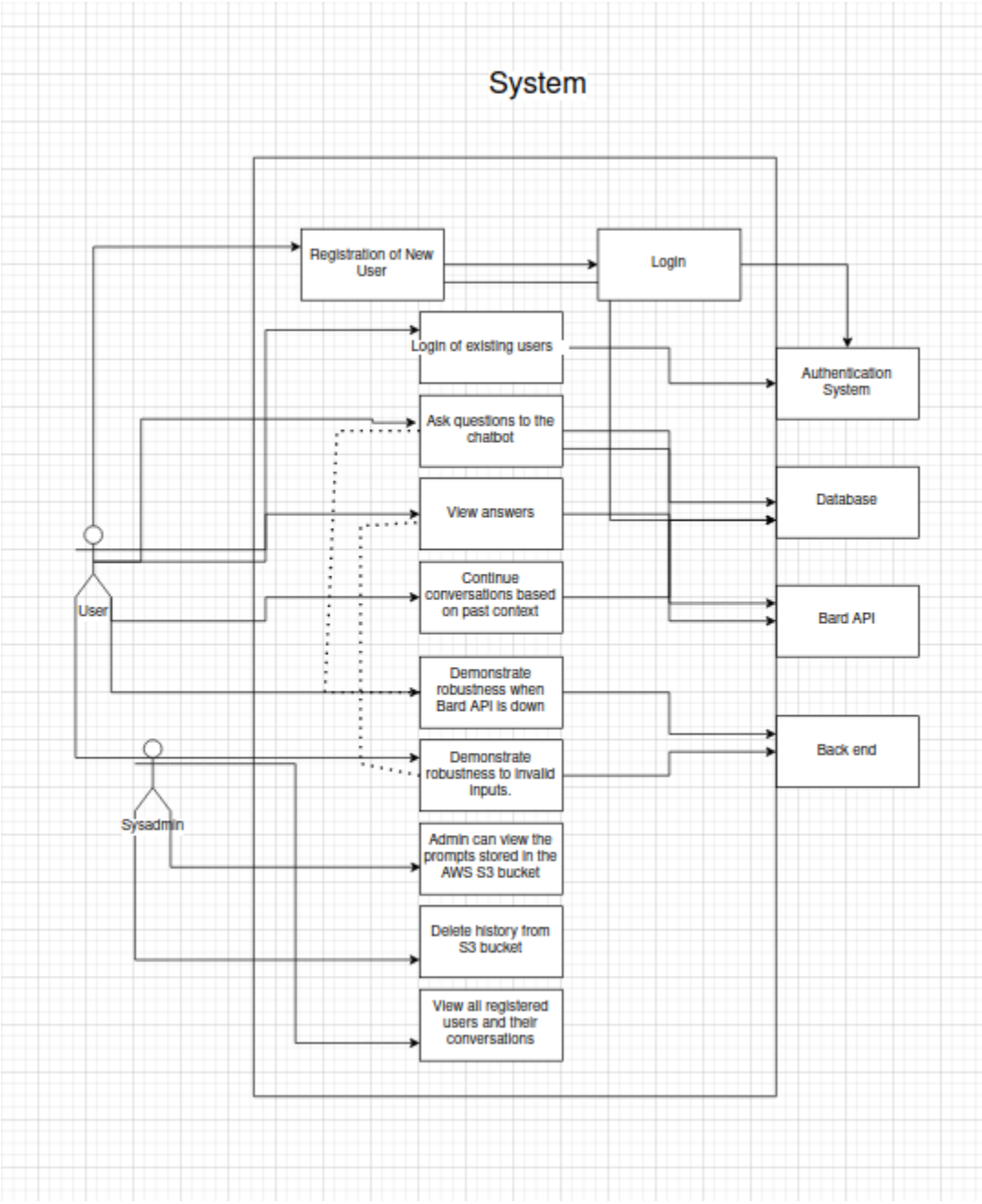
Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Registering new user	User will select the register option and be able to create a new profile on the website	R1
2.	Login existing user	Users will be able to select the login option from the landing page to log in to their existing account.	R1
3.	Ask questions to the chatbot	After entering the chatbot page , the user will be able to ask questions to the chatbot using the input text-box. The questions have to related to the context .	R1
4.	View answers	The user will be able to view chatbot responses to their prompts and decide to progress the conversation accordingly.	R1
5.	Continue conversations based on past context	The user will be able to build on current context of the conversation and carry forward the conversation with context specific questions	R1
6.	Demonstrate robustness when Bard API is down	In case of unresponsiveness of Gemini API , the chatbot goes temporarily down since it cannot generate responses	R1
7.	Demonstrate robustness to invalid inputs.	In case of inputs where Gemini cannot generate suitable outputs , the chatbot will accordingly take care of not displaying invalid outputs	R2
8.	Admin can view the prompts stored in the AWS S3 bucket	The chatbot admin will have an option to view all prompts stored in the S3 bucket.	R2
9.	Delete history from S3 bucket	The admin can delete instances from S3 bucket if need be.	R2
10.	View all registered users and their conversations	Admin can view all registered users and their uses of the chatbot.	R2

Non-functional Requirements

No.	Requirements	Description	Release
1)	Security	Data of all users(email, password etc) should be secured. It should not be accessible publicly	R1
2)	Availability of AI	The AI/API used in the project should be available in the Industry.	R1
3)	Compatibility	The project/web application should be compatible with multiple browsers/systems	R2
4)	Scalability	The system will be able to adjust with an increasing number of concurrent users.	R2

Use case diagram



Here dotted lines represent alternate flow whereas arrows represent main flow

Use case description

Use Case Number:	1
Use Case Name:	Registering New User
Overview:	User will select the register option and be able to create a new profile on the website
Actors:	User
Pre condition:	Users must not have already registered with the same Email before.
Flow:	Main (success) Flow: 1) User comes to the registration/sign-up page.
	Alternate Flows: If the user is already registered, redirect him to the login page after a suitable message from the website.
Post Condition:	User is led to login page where they can login with their new credentials

Use Case Number	2
Use Case Name	Login Existing User
Overview	User will click on the login page where the user can type in their credentials and enter the chatbot interface
Actors	User
Pre-condition	User must be registered beforehand

Flow	1. User comes to the main page 2. User clicks the login button 3. User enters credentials in the login page 4. If credentials are correct, user is redirected to the chatbot page
	Alternate Flows: 1.If the credentials are incorrect then a message is displayed saying that the credentials are wrong and the login page is refreshed. 2. If some credentials are left empty then a message is displayed saying that all details are not entered and the page is refreshed.
Post Condition	User is led to the chatbot page where he can enter prompts to get their required responses.

Use Case Number	3
Use Case Name	Ask questions to the chatbot
Overview	Users can type their prompts in the designated space and click the submit button present, the prompt is processed using the API, and the response generated is displayed to the user.
Actors	User
Precondition	User must be logged in
Flow	1. Users will type their query in the designated space. 2. A POST request is sent to the API
	Alternate Flows: If the API is down, an error message is displayed.
Post Condition	The request is processed and a response is generated from the API.

Use Case Number	4
Use Case Name	View Users

Overview	The user will be able to see their answers on the screen.
Actors	User
Precondition	The user must have submitted a query.
Flow	1. The API sends a GET request to our server 2. The server receives the request and the response is printed on the screen in a typing fashion
	Alternate Flows: If the API is not able to find a suitable answer to the query, a response can be printed that tells the user about the same.
Post Condition	The user can view the response on the screen, and the user can use it for their own purposes.

Use Case Number	5
Use Case Name	Continue conversations based on past context.
Overview	The user will be able to build on current context of the conversation and carry forward the conversation with context specific questions
Actors	User
Precondition	The user has already initiated a conversation with the chatbot
Flow	1. User is present in a conversation 2. Users can type their queries which can have ambiguous questions based on previous context. 3. The chatbot will interpret the context from the previous conversations and give a specific answer to the user.
Post Condition	The user can continue the conversations.

Use Case Number	6
Use Case Name	Demonstrate robustness when Bard API is down
Overview	In case of unresponsiveness of Bard API , the chatbot goes temporarily down since it cannot generate responses
Actors	User
Precondition	The API server is down
Flow	1. The API is down and not able to generate responses 2. An error message is displayed on the screen saying that the server is down.
Post Condition	A message is displayed letting the user know about the condition.User can try again after sometime once the server is up and running.

Use Case Number	7
Use Case Name	Demonstrate robustness to invalid inputs.
Overview	In case of inputs where Bard cannot generate suitable outputs , the chatbot will accordingly take care of not displaying invalid outputs
Actors	User
Precondition	User types in a prompt and no proper output is available in the API
Flow	1. User gives the prompt and sends the POST request. 2. The API receives and finds no proper response 3. A message is displayed on the screen saying that no proper response could be found due to limitations

Post Condition	A message is displayed to the user , letting him know that there is no apt answer for their query.

Use Case Number	8
Use Case Name	Admin can view the prompts stored in the AWS S3 bucket
Overview	The chatbot admin will have an option to view all prompts stored in the S3 bucket.
Actors	User, System Admin
Precondition	Users have interacted with chatbot
Flow	The admin will have an interface to view the prompts stored and filter them based on conditions which display matching prompts.
Post Condition	The sys-admin can take necessary decisions if need be.

Use Case Number	9
Use Case Name	Delete history from S3 bucket
Overview	The admin can delete instances from S3 bucket if need be.
Actors	User, System Admin
Precondition	There should be prompts in the file.
Flow	The sys-admin selects option to delete instance by selecting instance based on primary key and it will be deleted.
Post Condition	The relevant instances will be removed.

Use Case Number	10
Use Case Name	View all registered users and their conversations
Overview	Admin can view all registered users and their uses of the chatbot.
Actors	User, System Admin
Precondition	There should be registered users in system
Flow	The sys-admin will select the option to view users and based on email user will be selected and their history displayed
Post Condition	The sys-admin can take necessary decisions based on this if need be.