

Project Development Plan



Drugpedia

— Encyclopedia of drugs —

Prompt engineering for drug entity extraction for a
comprehensive drug directory

Team Members

Ahtasham Akram	40218017
Mian Basit Mahmood	40185188
Rohith Kodyath Padinhare Veed	40233953
Fatemeh Mokhtari	40215870

Date: 22/07/2023

1. Introduction

Access to accurate and up-to-date information about pharmaceuticals is crucial to assure patients' health in the ever-changing healthcare environment. The Comprehensive Medication Directory Service is an ambitious and innovative initiative that aims to revolutionize how healthcare professionals, pharmacists, and end consumers access and use medication information. In today's swiftly evolving medical landscape, the need for a centralized, trustworthy source of accurate, up-to-date pharmaceutical data has never been greater. This project aims to address this need by creating an innovative application that will serve as an exhaustive repository of pharmacological information, allowing for seamless user-system interactions.

1.1. Objectives of the project

The primary objective of this project is to develop a state-of-the-art Comprehensive Drug Directory Service that provides end users, such as healthcare professionals and pharmacists, with the most accurate and up-to-date information about a broad variety of medications through the application of prompt engineering techniques. With a focus on precision, accessibility, and user-friendliness, our mission is to empower medical practitioners to make informed decisions when prescribing medications, while also providing a valuable resource for end users to obtain comprehensive drug-related information.

1.2. Scope statement

Our initiative aims to create a user-friendly and exhaustive drug directory that compiles the most current and relevant information on pharmaceutical products by using prompt engineering techniques. By providing a streamlined and user-friendly interface, the application will enable healthcare professionals to make informed drug-prescribing decisions. This service will also benefit pharmacists, as they will be able to ensure accurate medication dispensing and provide patients with additional information when necessary.

1.3. Stakeholder profiles

Name	Position	Category (External / Internal)	Email
Dr. Chun Wang	Project Manager	Internal	chun.wang@concordia.ca
Ahtasham Akram	QA Engineer	Internal	atashamabbaxi@gmail.com
Mian Basit Mahmood	Tester	Internal	mianbasit1994@gmail.com
Fatemeh Mokhtari	Tester	Internal	fateme89am@gmail.com
Rohith Kodiyath Padinhare Vee	Team Lead	Internal	rohithkodiyath@gmail.com

2. Product and deliverables

2.1 Product Description

The end users, such as doctors and pharmacists, can utilize the associate web interface to search for drugs using their names or associated symptoms. This application is capable of finding alternative medicines by comparing their chemical compositions, which can be helpful in case a specific drug is not available.

2.2 Product Features

- **Extraction of drug properties from the text:** This involves gathering and extracting of information about a drug, including its description, uses, symptoms, and potential side effects with the help of the Chat GPT 3 language model. This information is then stored in a database for future reference.
- **Recommendations of the drug based on symptoms:** The users can search for drugs that treat multiple symptoms. The data will be sorted based on the drug's uses, with drugs that treat the most number of symptoms appearing first.
- **Extraction of drug details:** Users can fetch details of the drug with its name.

- ***Find a drug with a similar chemical composition:*** The alternative drug suggestion feature will list all drugs with the same chemical composition as the user's input.

2.3 Expected duration of the project

- **Starting date of project:** 12 July 2023
- It is expected that the project will complete within one month from its starting date.
- **Expected completion date:** 8 August 2023

3. Project interfaces

Software interfaces: This project has only software interfaces.

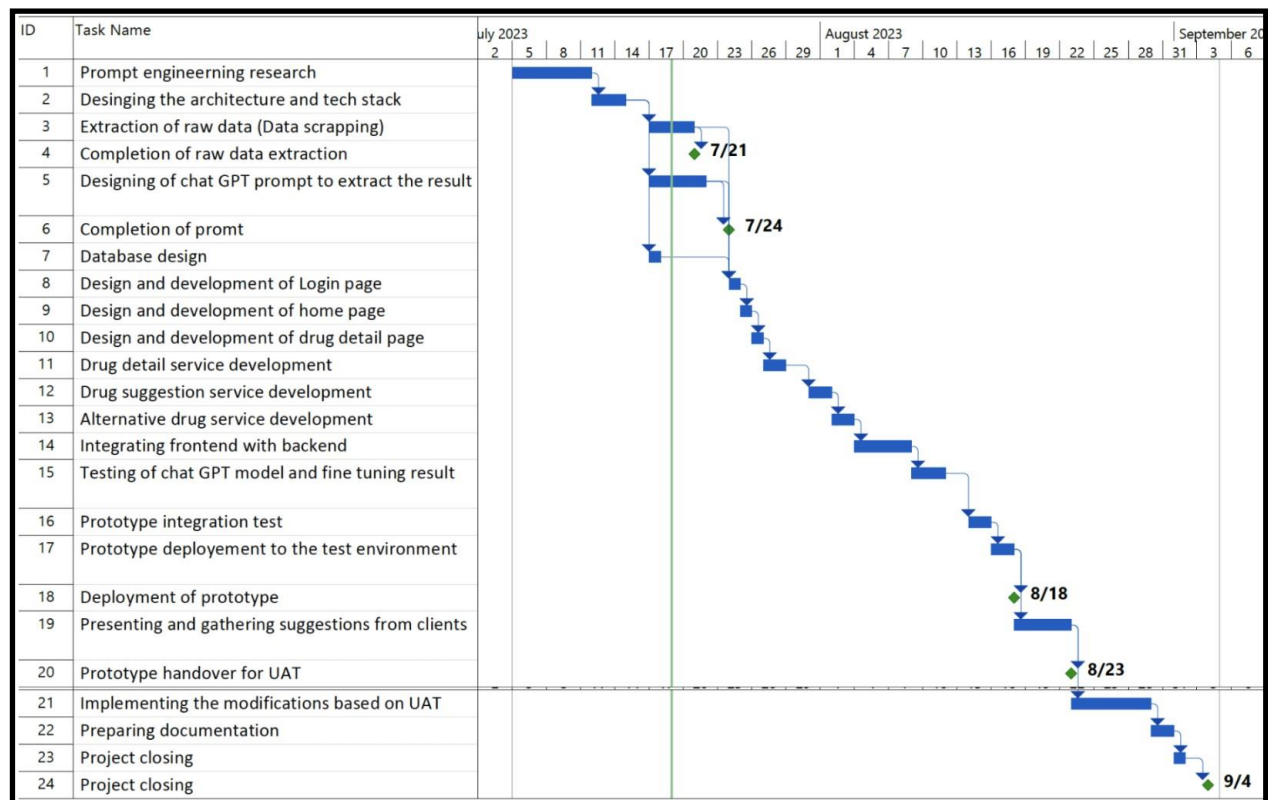
- The interface for the scheduler to run the data-scraping tasks.
- Another interaction is between the data scrapper and entity extraction service. This can be achieved with the help of a broker or queue.
- Web interface interacts with the system via secure API endpoints. Detailed API documentation is required for the front-end developers.

4. Software development process

For our project, we opted for an iterative (incremental) development model. Its stages are logical and necessary. The project manager defines the project, the business analyst elicits the requirements, the designer transforms them into software requirement specifications, the developer performs coding, and then the tester verifies the application.

5. Project process activities and schedule

The project tasks have been identified and their estimated duration has been determined. After sequencing these tasks based on their interdependencies, a Gantt chart has been prepared.



6. Project milestones

Project milestones are significant points in a project timeline that signify the completion of specific tasks or goals. These milestones help to measure progress and ensure that the project is on track to meet its objectives.

The primary goal of the drug directory project is to design a prompt for the language model that can extract drugs from the row format efficiently. The prompt's quality plays a crucial role in how well the system can find drugs for the user. All the project's identified milestones are explained below.

No	Project Milestone	Completion date	Deliverable
1.	Extraction of raw data	07/21/2023	Text file containing the raw data scrapped from the source website
2.	Engineering of prompt	07/24/2023	A prompt the extract entitles from the row data
3.	Deployment of prototype	08/23/2023	URL of the website for the
4.	Project closing	09/04/2023	API ends points, Video walkthrough links, Project development plan, SRS, SQAP, Final Report, FAQ, User Manual

7. Project team organization

Team Member	Designation	Roles and Responsibilities
Dr. Chun Wang	Project Manager	Monitoring and controlling the developers' and testers' preparation, execution, testing, and delivery of the web application.
		Defining the project's scope, timeline, and testing and development methodologies with the team's developers and testers.
Rohith Kodyath	Team Lead/ Website Developer	Transforming client or project specifications into user-friendly web applications.
		Consult with the project manager about the status of the project and any new proposals. Discussing testing, validation, and maintenance limitations and procedures with testers.
		Offer assistance and feedback into the workflow of the project management process.

Ahtasham Akram	SQA Engineer	<p>Create and execute test plans, test cases, and test procedures to assure software product quality and functionality.</p> <p>Work together with the programmers to fix any bugs you find and make sure the final product is of the highest quality.</p> <p>Establish and maintain testing documentation, such as test plans, test reports, and defect monitoring, to provide detailed records of testing activities.</p> <p>Provide constructive input to improve software development quality and efficiency.</p>
Mian Basit Mahmood / Fatemeh Mokhtari	SQA Testers/ Maintenance Specialists	Carry out comprehensive tests, both manually and automatically, in order to check the software's functioning and locate any flaws or discrepancies.
		Discussing quality assurance throughout the enterprise, reviewing SRS, SQAP, Project Development Report, and final SQA Report to understand testing scope and constraints.
		Verify the software's functionality, dependability, and usability by running test cases and scenarios, and recording and reporting any problems found.
		Improve the project development lifecycle's quality assurance procedures on a consistent basis by updating and expanding software testing techniques and methodologies.

8. Development facilities

To develop a drug pedia application, you will need a set of essential development facilities, tools, and technologies. Below is a list of key components required to create and maintain a drug pedia:

1. **Development Environment:**

- Integrated Development Environment (IDE): Choose a suitable IDE like Visual Studio Code, IntelliJ IDEA, or Eclipse, which offers features to enhance coding productivity and debugging.

2. **Programming Languages:**

- Backend: Select a server-side programming language, such as Python, Node.js (JavaScript), Ruby, or Java, to handle data processing and server-side operations.
- Frontend: Utilize HTML, CSS, and JavaScript (with frameworks like React, Angular, or Vue.js) for the client-side user interface.

3. **Database Management System:**

- Choose a relational database management system (DBMS) like MySQL, PostgreSQL, or SQLite to store drug information and other related data.
- Alternatively, consider a NoSQL database like MongoDB if a more flexible schema is required.

4. **APIs and Data Sources:**

- Identify reliable drug databases, medical APIs, or relevant APIs to fetch accurate and up-to-date drug information for your application.
- Consider partnering with medical organizations to access reputable data sources.

5. **Server Hosting and Deployment:**

- Choose a hosting provider to deploy your application, such as AWS (Amazon Web Services), Azure, DigitalOcean, or Heroku.
- Set up a deployment pipeline using tools like Docker and Kubernetes to automate the deployment process.

6. **Version Control:**

- Use a version control system like Git to track changes in your codebase and collaborate with a team efficiently.

7. **API Documentation Tools:**

- Utilize tools like Swagger/OpenAPI or Postman to document the APIs used in your application, making it easier for developers to integrate and understand them.

8. **Security Tools:**

- Implement security tools and practices to safeguard user data and protect against potential security threats, such as using HTTPS, SSL certificates, and input validation.

9. **User Interface Design Tools:**

- Design the application's user interface using tools like Adobe XD, Figma, or Sketch to create mockups and prototypes.
- 10. Error Tracking and Logging:**
 - Implement error tracking and logging mechanisms using services like Sentry or Loggly to monitor application errors and debug issues effectively.
 - 11. Automated Testing Frameworks:**
 - Set up automated testing using frameworks like Jest, Pytest, or Selenium to ensure the application functions correctly and remains stable across different scenarios.
 - 12. Performance Monitoring:**
 - Use performance monitoring tools to identify bottlenecks and optimize application performance. Tools like New Relic or Google Analytics can help gather insights.
 - 13. Documentation Generation Tools:**
 - Use tools like JSDoc, Sphinx, or Doxygen to automatically generate documentation for the codebase.
 - 14. Backup and Disaster Recovery:**
 - Set up automated and regular backups to prevent data loss in case of unexpected events or server failures.
 - 15. User Feedback and Analytics:**
 - Integrate tools like Google Analytics or Mixpanel to gather user behavior data and feedback for continuous improvement.

9. Development risks

ID	Risk Description	Risk Category	Likelihood of occurrence	Impact
R1	Extraction of unwanted data	Tool Risk	Medium	High
R2	Mismatch of entity attributes	Tool Risk	Low	High
R3	Source Websites blocking the scrapping of data	Technology Risk	Low	High
R4	Unresponsive ChatGPT model	Tool Risk	Medium	High
R5	Database Crash	Tool Risk	Low	High
R6	<i>Lack of</i>	People Risk	Low	Medium

	<i>communication among stakeholders</i>			
R7	The authentication of the user is not reliable	Security Risk	Low	High
R8	Webpage designs are not userfriendly	People Risk	Low	Medium
R9				

10. Control methods

Risk Name	Preventive Action	Corrective action
Extraction of unwanted data	Unwanted data extraction happens when the language model is not properly prompted to analyze information or when the data is unclear. To ensure the quality of the prompt, it is recommended to thoroughly test it with all possible raw data extracted.	Remove the entries as soon as possible from the database. To improve the quality and mitigate future errors give more examples of the language model.
Mismatch of entity attributes	At times, the language model may not recognize an entity's characteristics, leading to an entity mismatch. To prevent this, it's essential to test the model with all available raw data thoroughly. When such scenarios are identified, format the raw data to make it more informative for the language model to process.	Through vigilant data monitoring, anomalies can be swiftly identified. In the event of any erroneous data, promptly eliminate the entry and analyze the cause of the error.
Source Websites blocking the	Before scraping any data, it's	It's important for the system

scrapping of data	crucial to ensure that the website you are sourcing data from allows it. Additionally, it's essential for the system to keep a detailed record of any errors that occur during the process and immediately notify the developers.	to have a backup source of information in case the primary source fails. This switch should occur automatically and notify the maintenance team of the change. Relying solely on one source of information is not ideal.
Unresponsive ChatGPT model	Make sure not to use the free version of the Chat GPT model in the production release of the software.	During periods of high demand, Chat GPT has been known to be unresponsive. If this occurs, the system should be given several hours to retry. If the issue persists despite multiple attempts, it is recommended to notify the developers.
Database Crash	To reduce the number of database reads and prevent excessive load and connections, it is recommended to implement a caching layer within the service layer of the application. Additionally, enabling transaction support and read replicas can further optimize performance.	In the production environment, it is important to schedule periodic backups of the database and snapshots. Enabling transaction log backup, if supported by the database, ensures that no information will be lost in the event of a crash.
Lack of communication among stakeholders	It is important to regularly meet with stakeholders to assess the system and provide updates on progress. In the event of any issues or changes to requirements, be sure to obtain approval before proceeding.	Schedule a meeting with the stakeholders immediately to review the use case and SRS document. The project manager must be held accountable for ensuring effective client communication. Furthermore, arrange developer-client sessions as necessary to clarify and fully comprehend the requirements. This is essential to the success of the project and cannot be delayed any longer.
The authentication of the user is	For added security, it is recommended to enable SSL	To ensure safety and security, it is recommended

not reliable	encryption to prevent man-in-the-middle attacks. In cases of suspicious login attempts, it is advised to block the user. The security level of the application can be further increased by enabling OTP.	to block the user and prompt them to reset their password. Additionally, it may be useful to compile a list of suspicious IP addresses for future reference.
Webpage designs are not user-friendly	It is important to use visual representations and mockup webpages to effectively communicate the user interface to stakeholders. It is vital that the colors and fonts used are appealing to the user. By reducing the number of clicks required, user interaction can be improved.	It is important to pay close attention to the comments and suggestions provided by the user. When making changes, it is best to opt for modifications that do not significantly alter the user interaction. Drastic alterations to the user interface can cause confusion and reluctance to use the system.