Project Name: Period Prediction Chatbot

Number	Full Name	Student ID	Contribution	Role
1	Dương Thị Quỳnh Như	ITCSIU22202	25%	Team Leader
2	Phạm Gia Phúc	ITCSIU22178	25%	Team Member
3	Nguyễn Hoàng Thảo Trinh	ITCSIU22288	25%	Team Member
4	Nguyễn Vũ Phương Nghi	ITCSIU22282	25%	Team Member

I. Problem Statement

Menstrual cycle tracking, or period tracking, is a process of monitoring and recording individual information about person' menstrual cycle which is the recurring cycle of hormone changes for pregnancy. In our current busy world, women might find obstacles in remembering their period days which have an inconsistent cycle that leads to confusion and inconvenience in their life. Our system is created in order to solve the problem aforementioned, giving women an effective tool to track their period cycle and alert on abnormal cycles, giving users understanding about patterns and biological behaviours that are unique to them.

II. About Development Team

- **System Designer** - *Nguyễn Hoàng Thảo Trinh, Phạm Gia Phúc, Dương Thị Quỳnh Như*

Designed the program's structure and interaction flows between key components: User Interface, ANTLR processing, AST generation, controller, backend database, ensuring efficient workflow from user input to backend processing.

- **ANTLR Designer** - *Dương Thị Quỳnh Như, Nguyễn Vũ Phương Nghi*

Designed and implemented grammar rules using ANTLR to parse and interpret user inputs accurately and build the Abstract Syntax Tree file to call function.Conducted comprehensive testing of the chatbot's functionalities, ensuring accurate responses and robust performance.

- **Frontend Developer -** *Nguyễn Hoàng Thảo Trinh, Phạm Gia Phúc*

Designed and implemented a responsive, user-friendly interface using customtkinter, a Python-based GUI library. Focused on creating a clean, intuitive layout that allows users to communicate naturally with the chatbot.

- **Backend Developer** - *Phạm Gia Phúc, Dương Thị Quỳnh Như, Nguyễn Vũ Phương Nghi*

Developed the backend logic within controller directories to manage user interactions and data processing using Python. Integrated MongoDB for data storage, also efficiently managing user's information and chatbot response.

III. Framework Description

As can be seen from this Luna chatbot workflow, our model inherits the MVC (model-view-controller) architecture and consist of totally 9 components, and here is the breakdown of the detail description of framework workflow:

- First, the Frontend, which is built based on the customtkinter library of Python, will receive user input and send it back to the controller in which it will be delivered to the ANTLR processing in order to check whether user input matches with the allowed sentences. In case that the user input is in the wrong format, a default response will be sent back to the user.
- Otherwise, an Abstract Syntax hierarchical Tree (AST), which is produced from Visitor generated by ANTLR, is utilized to call appropriate functions for each input.
- The functions of each input will be stored in the model folder which will then be operated with the database to retrieve useful information. Here, we utilize cache for short and immediate calling of information to save time and MongoDB for long term information.
- Lastly, the response will be formatted and sent back to the user via Frontend display.

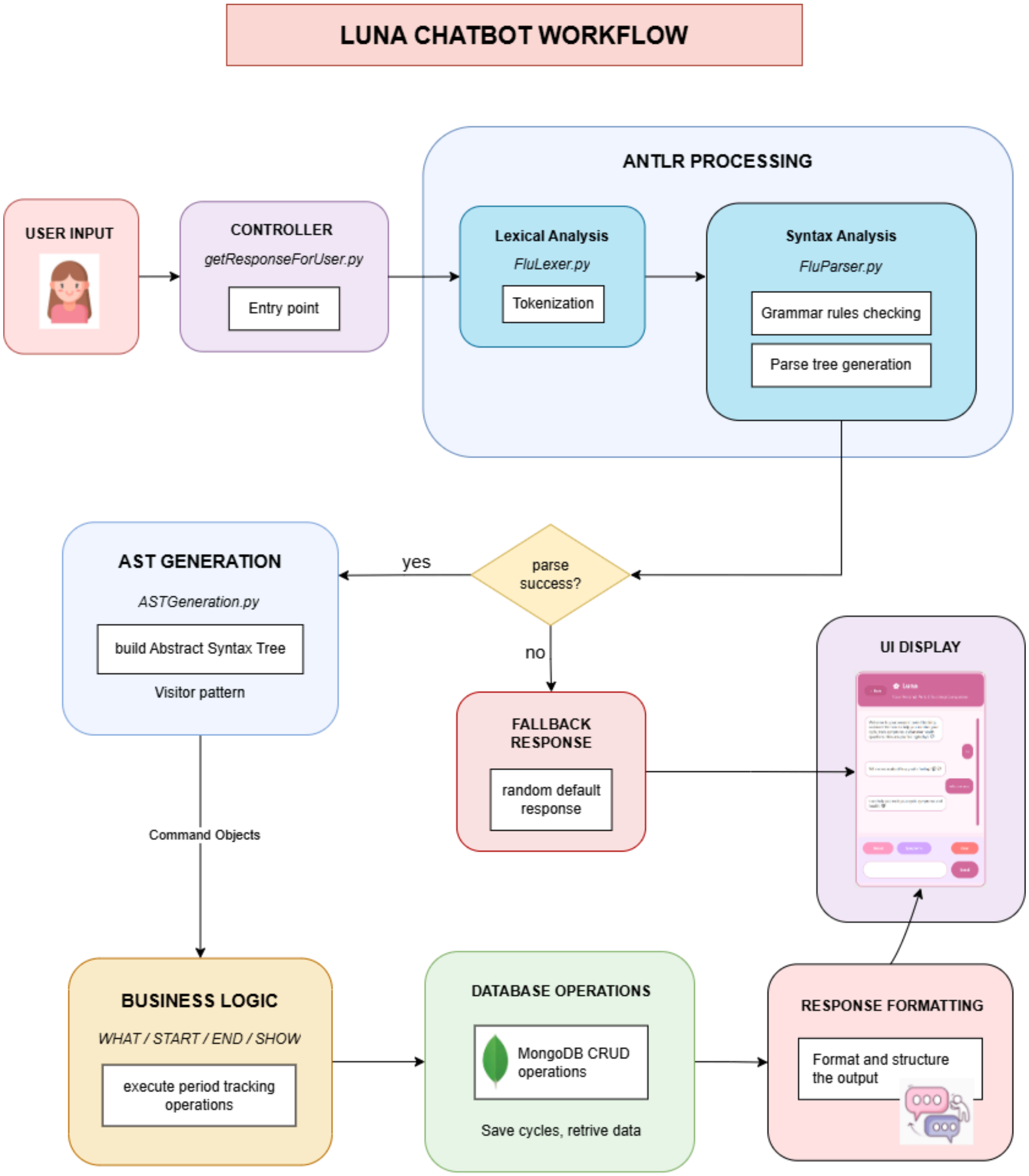
The request accepted by our system include:

Request	**Expected return**	**Operations included**
Check cycle statistic	The cycle length (from the beginning of one period to the beginning of next period) and period length (from the beginning till the end of one period). Check if these two lengths are in normal range.	Take the mean of cycle length and period length of 4 closest periods. Compared with the normal threshold (proved by the researcher of the field), 21-35 days for cycle length and 2-7 days for period length.
today i feel symptom, symptom,...	Whether the symptom is related to the period or due to outsider reasons.	Check if the current day is in period and the symptom is normal for the period. Else, suggest visiting a doctor.
show period cycle	The 3 closest period that have past	Retrieve period information from MongoDB and return.
start date (date can be a *specific date* such as 31-5-2025 or *date in word* like today, tomorrow, yesterday or *N*	Show that the start date has been successfully log into the database	Save start date in database

days before/ after)		
end date (date can be a *specific date* such as 31-5-2025 or *date in word* like today, tomorrow, yesterday or *N days before/ after*)	Show that the end date has been successfully log into the database	Save end date in database
what is fertile days on month? (month can be this/previous/next month or N month later)	Show the fertile range of the mentioned month	Take the average of cycle length to predict the period of the mentioned month. The start of a fertile day is 19 days earlier than the period and it will last in 5 days (according to researchers).
what is ovulation days on month? (month can be this/previous/next month or N month later)	Show the ovulation day of the mentioned month	Take the average of cycle length to predict the period of the mentioned month. The ovulation day is 14 days earlier than the period (according to researchers).
what is non-fertile days on month? (month can be this/previous/next month or N month later)	Show the non-fertile ranges of the mentioned month	Take the average of cycle length to predict the period of the mentioned month which are used to calculate the fertile window. The non-fertile ranges are from the days of the period till the start day of the fertile window and from the end of the fertile window to the start of the next period.
what is period days on month? (month can be this/previous/next month or N month later)	Show the period ranges of the mentioned month	Take the average of cycle length and period length to predict the period ranges of the mentioned month.
what is cycle status on date? (date can be a *specific date* such as	Show the cycle status (period, ovulation, fertile, non-fertile) on the mentioned date.	Take the average of cycle length and period length to predict the period ranges of the mentioned month.

31-5-2025 or *date in word* like today, tomorrow, yesterday or *N days before/ after*)		From that can check for fertile, non-fertile range and ovulation days.

- **System Workflow:**



IV. Principles of Programming Languages in Application

The Principle of Programming Language (PPL) knowledge is very crucial for the development of our project since several key concepts from the course are utilized to build an effective parser, which supports the user's natural language input, particularly in the parserAnalyzer module.

As shown on the table above, after clarifying the accepted user input, our group used a lexical analyzer that breaks down user input tokens. First, we apply the PPL knowledge gained in laboratory class to generate the .g4 file to create rules that describe valid sentence patterns. We divide the accepted inputs labeled as ***sentence*** into 4 main groups: require, ask, checkStats and symptom based on the similarity in the patterns of user input. The ***symptom*** starts with a date in word (today) and uses recursive rules for symptom types so that users can input multiple symptoms if they have a comma. The ***checkStats*** will be triggered by the keyword "check cycle statistics". The ***require*** express actions (such as "show", "end", and "start") for the operations with the period cycle. The ***ask*** begins with the keyword "what" and followed with the information about period state ("cycle status" or specific phrase such as "ovulation", "period", "fertile", "non-fertile") and must be ended with a question mark. Moreover, ***date*** expressions can be shown in 3 different types: ***dateInWord*** (today. yesterday, tomorrow), ***dateInNum*** (specific date such as 31-05-2025) and ***dateCompare*** ("n days before" or "after").

By applying the PPL knowledge, we allowed our system to recognize and act based on the diverse range of users' input by turning the user's raw input into tokens, making it easier for the program to understand and work with. For example, the rule "specificPharse : (OVU | PER | FER | NONF) (DAYS)? (ON|IN)? dateMonth;" allows the chatbot to recognize inputs like "non-fertile days 2 months later". By matching user input to grammar structures like ask, require, or date, the parser builds an Abstract Syntax Tree (AST). This helps the chatbot understand the logic behind the message and respond correctly based on the user's intent.

From there, we applied the hierarchical abstract syntax tree, which is represented as ASTGeneration.py in our source code. This file turns the rule into an abstract tree format that emphasizes the logical structure of the input of the user. Each visit method, visitProgram(), visitAsk() for example, corresponds to a grammar rule from Flu.g4 and goes through the tree to get what the user is entering. For instance, the method visitDateMonth() parses dates like "2 months later". Therefore, this helps our chatbot project understand user's input and use appropriate functions according to the input to response with correct information for period tracking

V. Demo Link

Youtube video: https://youtu.be/zNwewXYeHIA