**Authentication Subsystem**

**Class**: Auth

* **Attributes**:
  + user\_id: integer, private - Unique identifier for the user.
  + session\_token: string, private - Token that maintains the session state across interactions.
  + email\_host: string, private - SMTP server host, derived from config (EMAIL\_HOST).
  + email\_port: integer, private - SMTP server port, derived from config (EMAIL\_PORT).
  + email\_user: string, private - SMTP server username, derived from config (EMAIL\_USER).
  + email\_password: string, private - SMTP server password, derived from config (EMAIL\_PASSWORD).
* **Operations**:
  + verify\_credentials(username, password): boolean, public - Validates user credentials against stored data.
  + send\_verification\_email(email\_address): void, public - Sends an email to verify the user's email address using SMTP settings.

**Monitoring Subsystem**

**Class: PriceMonitor**

* **Attributes**:
  + url: string, private - URL to monitor for price changes.
  + current\_price: float, private - Most recent fetched price.
* **Operations**:
  + check\_price(): float, public - Fetches and returns the current price from the URL.
  + start\_monitoring(frequency): void, public - Begins monitoring the price at specified intervals.
  + stop\_monitoring(): void, public - Stops monitoring the price.

**Class: AvailabilityMonitor**

* **Attributes**:
  + url: string, private - URL to monitor for availability changes.
  + availability\_status: string, private - Latest availability status fetched.
* **Operations**:
  + check\_availability(): string, public - Checks and returns the current availability from the URL.
  + start\_monitoring\_availability(frequency): void, public - Starts monitoring availability at specified intervals.
  + stop\_monitoring\_availability(): void, public - Stops monitoring availability.

**Browser Operation Subsystem**

**Class**: BrowserController

* **Attributes**:
  + browser\_instance: Object, private - Instance of the browser controlled by the bot.
* **Operations**:
  + open\_browser(url): void, public - Opens a browser window and navigates to the specified URL.
  + close\_browser(): void, public - Closes the currently open browser window.
  + login(url, username, password): boolean, public - Logs into a website with provided credentials.

**Data Handling Subsystem**

**Class**: DataHandler

* **Attributes**:
  + database\_connection: Object, private - Connection object for the database.
* **Operations**:
  + save\_data(data): void, public - Saves specified data to the database.
  + retrieve\_data(query): Object, public - Retrieves data from the database based on the query.
  + export\_data\_to\_excel(data): void, public - Exports data to an Excel file.
  + export\_data\_to\_html(data): void, public - Converts and saves data in HTML format.

**Communication Subsystem**

This subsystem facilitates all user interactions and communications, both inbound and outbound.

**Class: MessagingSubsystem**

* **Attributes**:
  + command\_parser: CommandParser, private - An instance of the CommandParser class to parse user commands.
  + response\_templates: dict, private - A dictionary mapping commands to response templates for generating user messages.
* **Operations**:
  + receive\_message(user\_input): string, public - Receives input from the user, parses it to determine the command, and retrieves the appropriate response.
  + send\_message(channel\_id, message): void, public - Sends a message to a specific Discord channel, using the channel ID.
  + format\_response(command, data): string, public - Formats the response based on the command executed and the data retrieved from the system.

**Class: NotificationSubsystem**

* **Attributes**:
  + email\_config: EmailConfig, private - Configuration settings for email, containing host, port, user, and password.
* **Operations**:
  + send\_email(subject, body, recipient): void, public - Sends an email to the specified recipient with a subject and body.
  + notify\_user(channel\_id, message): void, public - Sends a notification message to a user via a specific Discord channel.
  + generate\_email\_body(template\_id, data): string, public - Generates the body of the email based on a template and data, useful for price changes, availability updates, etc.