**Technical Specification - Grok Insights Bot**

**1. Project Overview**

**1.1 Purpose**

The Grok Insights Bot is a Telegram bot designed to analyze social media posts on X by topic, leveraging multiple AI-powered APIs to generate concise reports on sentiment, keywords, and entities. The project showcases expertise in AI integration, bot development, and modular software design, serving as a portfolio piece for demonstrating full-stack development skills.

**1.2 Objectives**

* Develop a Telegram bot that integrates seven AI APIs (xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson) for text analysis.
* Provide users with an intuitive interface to select services, analyze topics, view history, and share results.
* Support Russian and English languages for accessibility.
* Store analysis results in a SQLite database for history tracking.
* Ensure modular architecture for easy extensibility.

**1.3 Target Audience**

* Marketers, startups, and researchers needing real-time social media insights.
* Developers and recruiters evaluating AI integration and bot development skills.

**2. Functional Requirements**

**2.1 User Commands**

* /start: Initiates the bot with a welcome message.
* /select\_service: Allows users to choose an analysis service (xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson).
* /analyze <topic>: Analyzes X posts by topic, returning a report with sentiment, keywords, and/or entities.
* /history: Displays the user's analysis history.
* /help: Lists available commands.

**2.2 Inline Buttons**

* **Repeat**: Repeats the last analysis.
* **Share**: Shares the report via Telegram.
* **History**: Views analysis history.

**2.3 Analysis Services**

* **xAI**: Analyzes posts using Grok API (sentiment, keywords, trends). Requires credits.
* **OpenAI**: Uses GPT-3.5-turbo for simulated post analysis. Requires credits.
* **Hugging Face**: Uses distilbert-base-uncased-finetuned-sst-2-english for sentiment analysis (~1000 requests/day free).
* **uClassify**: Uses sentiment classifier for sentiment analysis (unlimited free requests).
* **TextRazor**: Analyzes sentiment and entities (500 requests/day free).
* **MonkeyLearn**: Uses cl\_SentimentAnalyzer for sentiment analysis (requires corporate account).
* **Watson**: Uses Natural Language Understanding for sentiment and keywords (requires credits).

**2.4 Localization**

* Supports Russian and English, with messages stored in handlers.py (MESSAGES dictionary).
* Default language: Russian, configurable per user.

**2.5 Database**

* SQLite database (insights.db) stores:
  + User settings (selected service, language).
  + Analysis history (user ID, topic, result, timestamp).

**3. Non-Functional Requirements**

**3.1 Performance**

* Response time: <5 seconds per analysis request (dependent on API latency).
* Handle up to 100 concurrent users.

**3.2 Scalability**

* Modular service architecture (services/ directory) allows adding new APIs without core code changes.
* SQLite database supports small-scale usage; scalable to PostgreSQL if needed.

**3.3 Security**

* API keys stored in config.py (placeholders for public repository).
* No sensitive data stored in the database.

**3.4 Reliability**

* Error handling for API failures (e.g., "insufficient credits", "invalid key").
* Logging of API responses and errors for debugging.

**4. Technical Architecture**

**4.1 Technologies**

* **Python 3.8**: Core language.
* **aiogram 2.21**: Telegram Bot API framework.
* **APIs**: xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson.
* **SQLite**: Database for user settings and history.
* **aiohttp**: Asynchronous HTTP requests to APIs.
* **Git**: Version control.

**4.2 Project Structure**

* main.py: Bot initialization and startup.
* config.py: API keys and service configuration.
* utils.py: API interaction and database utilities.
* handlers.py: Command and button handlers.
* create\_db.py: SQLite database initialization.
* services/:
  + xai.py, openai.py, huggingface.py, uclassify.py, textrazor.py, monkeylearn.py, watson.py: API-specific logic.

**4.3 API Integration**

* **Endpoints**:
  + xAI: https://api.x.ai/v1/chat/completions
  + OpenAI: https://api.openai.com/v1/chat/completions
  + Hugging Face: https://api-inference.huggingface.co/models/...
  + uClassify: https://api.uclassify.com/v1/uclassify/sentiment/classify
  + TextRazor: https://api.textrazor.com/
  + MonkeyLearn: https://api.monkeylearn.com/v3/classifiers/...
  + Watson: https://api.us-south.natural-language-understanding.watson.cloud.ibm.com/...
* **Authentication**:
  + Bearer tokens (xAI, OpenAI, Hugging Face, TextRazor).
  + Token-based (uClassify, MonkeyLearn).
  + API key via BasicAuth (Watson).
* **Error Handling**: Captures "insufficient credits", "invalid key", and network errors, returning user-friendly messages.

**5. Deliverables**

* Source code in Rostislav62/grok\_insights\_bot repository.
* README.md with project overview, usage, and structure.
* SQLite database schema (insights.db).
* Modular API service modules in services/.

**6. Constraints**

* **API Limits**:
  + xAI, OpenAI, Watson: Require active credits.
  + Hugging Face, uClassify, TextRazor: Free tiers (~500-1000 requests/day).
  + MonkeyLearn: Limited to corporate accounts.
* **Text Input**: Short texts may yield limited sentiment analysis (e.g., uClassify: 0.00% confidence).

**7. Assumptions**

* Users have Telegram installed and can interact with the bot.
* API keys are obtained from respective providers (x.ai/api, platform.openai.com, huggingface.co, uclassify.com, textrazor.com, monkeylearn.com, cloud.ibm.com).
* SQLite is sufficient for initial user load.

**8. Success Criteria**

* Bot successfully analyzes topics via all seven services (or returns expected errors like "insufficient credits").
* Users can switch services, view history, and share results.
* Code is modular, documented, and publicly available on GitHub.
* Project demonstrates AI integration, bot development, and database skills.

**9. Author**

Rostislav — Full-stack developer specializing in AI-powered applications and bot development.