AndroAI Project Module Reports

1. Web Form Module:

Description:
The Web Form Module provides a user-friendly interface designed with React. Users can fill out necessary information for creating an Android application. The form collects various details such as the application's name, the number of activities, the connections between activities, button placements, and other specific requirements.

Purpose:
To collect and validate user inputs necessary for generating an Android application. This ensures that all required information is gathered before proceeding to the next steps.

Functionality:

- **Form Design:** Interactive forms designed using React components.
- **Input Validation:** Ensures that user inputs meet the required criteria before submission.
- **Data Submission:** Sends the collected data to the Prompt Creation Module.

Technologies Used:

- React for front-end development.
- Formik and Yup for form handling and validation. (optional for now)

Connections with Other Modules:

- **Prompt Creation Module:** Receives user data to create an appropriate prompt.
- **Debugging Module:** Processes feedback during the debugging phase.

---

2. Prompt Creation Module:

Description:
The Prompt Creation Module processes the user data received from the Web Form Module and converts it into a suitable prompt format for the LLM (Large Language Model) API. It ensures that the data is structured in a way that the LLM can understand and generate the required code.

Purpose:
To transform user inputs into a structured prompt that can be sent to the LLM API for code generation.

Functionality:

- **Data Processing:** Analyzes and structures the user data.
- **Prompt Generation:** Creates a detailed prompt based on user requirements.

- **Error Handling:** Ensures that the prompt is free of errors and complete.

Technologies Used:

- Python for backend processing.
- JSON for data formatting.

Connections with Other Modules:

- **Web Form Module:** Receives user data.
- **LLM-Model Module:** Sends the generated prompt for code generation.

---

3. LLM-Model Module (API):

Description:
The LLM-Model Module acts as an interface between the system and the Large Language Model (LLM). It sends the prompt generated by the Prompt Creation Module to the LLM and receives the generated Android application code.

Purpose:
To utilize the LLM for generating Android application code based on the structured prompt.

Functionality:

- **API Communication:** Handles API requests and responses.
- **Code Generation:** Receives the generated code from the LLM.
- **Error Handling:** Manages errors during API communication.

Technologies Used:

- RESTful API design.

Connections with Other Modules:

- **Prompt Creation Module:** Receives the structured prompt.
- **File Creation Module:** Sends the generated code for file structuring.

---

4. File Creation Module:

Description:
The File Creation Module takes the generated code from the LLM-Model Module and creates the necessary file structure for an Android application. This includes creating manifest files, resource directories, and Java/Kotlin source files.

Purpose:
To organize the generated code into a standard Android project structure.

Functionality:

- **File Structuring:** Creates directories and files as per Android project standards.
- **Code Placement:** Places the generated code in the appropriate files and directories.
- **Manifest Generation:** Creates and populates the AndroidManifest.xml file.

Technologies Used:

- Python (or C) for file system operations.
- Standard Android project structure guidelines.

Connections with Other Modules:

- **LLM-Model Module:** Receives the generated code.
- **Compilation Module:** Provides the structured files for compilation.

---

5. Compilation Module:

Description:
The Compilation Module takes the structured files from the File Creation Module and compiles them into an APK file. It uses tools like Gradle or Makefile to perform the build process.

Purpose:
To compile the generated code into a working APK file that can be installed on Android devices.

Functionality:

- **Build Automation:** Uses Gradle (gradle is only an option for now because it is cumbersome) or Makefile for automating the build process.
- **Error Handling:** Captures and reports errors during the build process.
- **APK Generation:** Produces the APK file.

Technologies Used:

- Gradle for build automation.
- Java/Kotlin compilers (javac).

Connections with Other Modules:

- **File Creation Module:** Receives the structured files.
- **APK Delivery Module:** Sends the generated APK file to the user.
- **Debugging Module:** Reports errors encountered during the build process.

---

6. Debugging Module:

Description:
The Debugging Module handles errors that occur during the APK generation process. It analyzes the errors, communicates with the LLM-Model Module to generate fixes, and reprocesses the code until a successful build is achieved.

Purpose:
To ensure the generated APK is error-free and meets the user's requirements.

Functionality:

- **Error Analysis:** Identifies and analyzes errors during the build process.
- **Automated Fixes:** Communicates with the LLM to generate fixes for the identified errors.
- **Rebuilding:** Repeats the build process with the fixed code until a successful APK is generated.

Technologies Used:

- Makefile that prints error output to file
- Gradle or Makefile for rebuilds.

Connections with Other Modules:

- **Compilation Module:** Receives errors for analysis.
- **LLM-Model Module:** Communicates to generate fixes.
- **APK Delivery Module:** Ensures the final APK is error-free before delivery.

---

7. APK Delivery Module:

Description:
The APK Delivery Module is responsible for delivering the generated APK file to the end-user and collecting feedback for future improvements.

Purpose:
To provide the user with the final APK file and gather feedback to improve the system.

Functionality:

- **APK Delivery:** Sends the generated APK file to the user.
- **Feedback Collection:** Collects feedback from the user regarding the APK and overall process.

Technologies Used:

- Direct download links for delivery.
- Feedback forms or surveys for feedback collection.

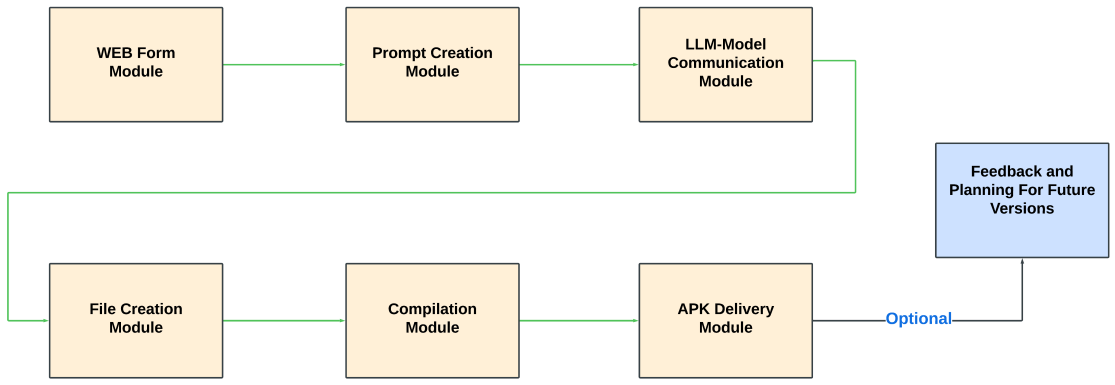
Connections with Other Modules:

- **Compilation Module:** Receives the generated APK file.
- **Web Form Module:** May process feedback for future versions.
- **Debugging Module:** Feedback may initiate debugging and improvements.

Module Connection Diagram

The following diagram illustrates the connections and interactions between the different modules:

SCENARIO 1:



SCENARIO 2:

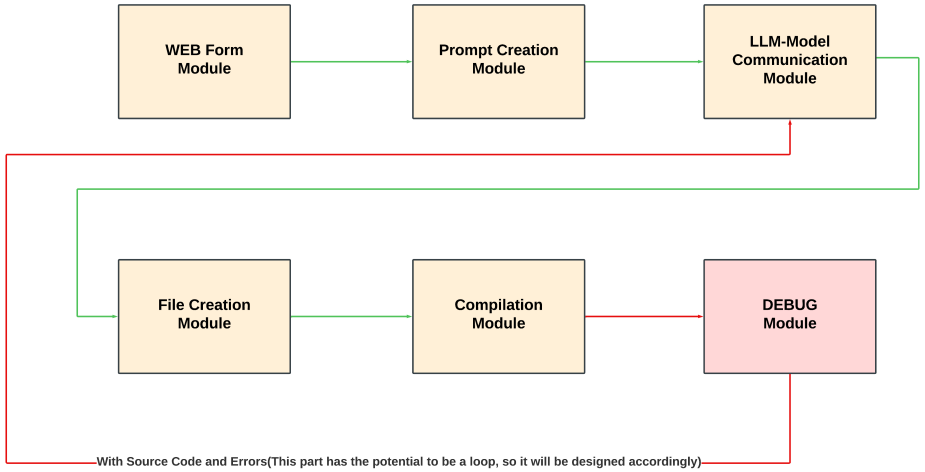


Diagram Explanation:

- **Web Form Module** -> **Prompt Creation Module** -> **LLM-Model Module (API)** -> **File Creation Module** -> **Compilation Module** -> **APK Delivery Module**.
- **Debugging Module:** Interacts with the **Compilation Module** and communicates with the **LLM-Model Module** as needed.
- **Feedback Loop:** After receiving the APK, user feedback is analyzed, and necessary planning is made for future improvements.

Feedback and Planning for Future Versions

- **Feedback Collection:** User feedback is collected after the APK delivery.
- **Analysis and Planning:** The feedback is analyzed, and necessary planning is done to incorporate improvements in future versions.

This report outlines the detailed functionality, purpose, technologies, and interactions of each module within the AndroAI project. It serves guide for understanding the project's structure and workflow.