<u>The progress on Siemens IIoT App Interface and the future goals and things to do:</u>

- Things we have:
    - PLC to AWS IoT Core:
        - MQTT connection
        - Read and write analog and bool signals.
    - AWS IoT Core to APP (MQTT communication)
        - Cross Platform app: Frontend - Flutter App (Cross Platform App – both iOS and Android)
            - The UI gets updated automatically when new values are received from the PLC. (flutter state management using flutter Provider and sync timer)
            - Authentication: Using email and password using AWS Cognito pools and AWS IAM permissions and roles (allowed social (google) login in iOS only).UI using Amplify Authenticator. (flutter: amplify_flutter, amplify_auth_cognito and amplify_authenticator)
            - User Management: Admin Users have the option to add new users to access site data and remove users.
            - Stations Management: digital, analog and SP control: can have multiple stations and tags for each station.
            - Communication with the AWS IoT Core using MQTT protocol to publish messages and subscribe to topic, secure connection for user using Fleet provisioning by Claim. (flutter: mqtt_client)
            - Getting the data from the backend AWS GraphQL API (Amplify Data Store –AWS DynamoDB) and building the Widgets on the Data. (flutter: amplify_datastore and amplify_api)
            - Data Logging: Alarm Log (with Active Alarms), Pumps Log, Pit Level Log, Effluent Flow Log and Set Points Logs (Line Chart View with DateTime for Trends and Text View). (flutter: fl_chart)
            - Storage: Storing data on the backend AWS S3. (flutter: aws_storage_s3 and aws_common)

        - Syncing data using AWS AppSync to AWS DynamoDB tables.
        - Local Notifications (using flutter_local_notifications) – can be customized, user preferences saved using flutter: shared_preferences to retain on app restart and notify when app is running in the foreground and in background.
        - Push Notifications (using AWS SNS (platform applications and endpoints using device tokens - Firebase Console FCM): IoT message from PLC triggers AWS Lambda functions that sends a notification to the device.(flutter: firebase_core and firebase_messaging)
        - Email Notifications: AWS SES (used for sign up experience), for alarm notifications (user verified again when email notifications enable using lambda function - email sent and user needs to click on the link), notifications from IoT/+/+/+/AlarmWord and ~/Pump/Status topic.
        - SMS Notifications: AWS SNS (Text messages - user phone Number verified sandbox number - verification code sent to user) and receive notifications from IoT/+/+/+/AlarmWord and ~/Pump/Status topic.
    - Backend
        - AWS Amplify studio: manage users, manage content (add PLC data to the table accordingly to the fields).
        - AWS SNS or AWS Pinpoint to send push notifications (flutter: amplify_push_notifications_pinpoint) via FCM (Firebase Console)
        - Analytics of Endpoints. (flutter: amplify_analytics_pinpoint)

- Things to do in future for the IOT Applications:
    - App Side Changes:
        - Test the Entire App flow: Admin Management, Communication with PLC, Real time Updating values in the new UI, Logs and Trends - Filter the Charts with timestamp selection and updating the backend Databases and Storage, Testing Local and Push

Notifications, Email and SMS Notifications for Alarms and Pump Alarms.

- Optional:
    - Local Database to have constant logs and store only on app terminates or once a day.
    - A monthly billing system (maybe on the app)
    - Cost Analysis per user.
    - Being able to add app information on the Indus automation website (probably add an APK along with a demo functionality).
    - Determining the cost of the sending the data from PLC to AWS, APP TO AWS and the cost of the storage of the logs in AWS S3 or Data Log CloudWatch: setting a limit of the number of data to be logged or stored and keep that customizable (if user want to increase).
    - Web Application.
    - Application Release.