

ExecAI — Intelligent Executive Assistant

Rachelle Falcon • Zack Kohlman • Johana Huertas

Faculty Advisor: Masoud Sadjadi • Spring 2026

Agenda

- 01 Problem Definition**
- 02 System Architecture**
- 03 Implementation Highlights**
- 04 Demonstration Overview**
- 05 Testing & Evaluation**
- 06 Challenges & Lessons Learned**
- 07 Future Work**

Problem Definition

The Gap in Today's Workflow

Professionals managing calendars and email spend significant time on repetitive scheduling and communication tasks.

There is no unified, conversational tool that allows users to interact with both their calendar and email using plain language.

Current tools operate in silos:

- Calendar apps show availability but can't draft or send email

- Email clients send messages but don't know when you're free

- AI assistants are smart in conversation but don't plug into your tools

No system interprets "Reply to my latest email saying I'm available at 2pm and create the meeting" and executes all steps automatically.

System Architecture

High-Level Design

1. Intent Parser

Hybrid NLP: rule-based regex + OpenAI (primary), Groq fallback. Extracts 12+ entity types.

2. Orchestrator

Decision engine mapping 9 intents to provider-agnostic actions. Handles conflict detection, alternative slots, and atomic combined workflows.

3. Availability Engine

Queries real free/busy data via Google Calendar or Microsoft Graph. 30-min slots within 9am–5pm, ranks top 3 conflict-free.

4. AI Draft Generator

OpenAI LLM generates context-aware emails with Groq and template fallback.

5. Integrations

OAuth 2.0 with auto-refresh for Google & Microsoft. Calendar CRUD, Gmail drafts/send, Outlook via Graph API.

Key Metrics

9 Intents Understood • 2 Providers Unified • 3 Alternatives Ranked • 12+ Entities Extracted

Technology Stack:

Python • FastAPI • Streamlit • Pydantic • OpenAI • Groq • Google Calendar API • Gmail API • Microsoft Graph • OAuth 2.0

Implementation Highlights

Intent Parsing & Orchestration

Hybrid Intent Parser

Rule-based regex for common patterns with OpenAI as primary LLM parser and Groq as fallback.

Extracts 12+ entity types: title, timeframe, attendees, tone, body hint, duration.

Handles multi-intent requests in a single message (e.g., reply + create meeting).

Orchestrator Engine

Maps 9 distinct intents to provider-agnostic actions across Google and Microsoft.

Conflict-aware scheduling with real free/busy data queried via Calendar API or Graph.

Atomic combined workflows: reply+create and draft+schedule with rollback on failure.

AI Drafting & Provider Integrations

AI Draft Generator

OpenAI LLM generates context-aware email drafts with tone control (formal, casual, urgent).

For replies, injects real email thread content (subject, sender, body).

Triple fallback chain: OpenAI → Groq → template-based generation.

Provider Integrations

OAuth 2.0 with automatic token refresh for both Google and Microsoft.

Google: Calendar CRUD + Gmail draft/send via Google APIs.

Microsoft: Calendar + Mail via Microsoft Graph API.

Provider-agnostic layer: same user experience regardless of backend.

Demonstration Overview

What the Demo Covers

A separate demo video walks through ExecAI's core workflows end-to-end:

Natural Language Scheduling

User requests a meeting → ExecAI parses intent, checks availability, detects conflicts, suggests alternatives, creates event.

AI Email Drafting & Replies

User asks to draft/reply → ExecAI pulls real thread context, generates professional draft with tone control.

Combined Workflows

"Reply saying I'm available at 2pm and create the meeting" → both actions executed atomically.

Conflict Detection

When a time is busy, ExecAI shows the conflict and ranks top 3 alternative slots.

Testing & Evaluation

Verification Approach and Results

Intent Accuracy:

9 intents tested across 40+ NL variations using rule-based and LLM parsers.

Integration Testing:

E2E tests for Google & Outlook event creation, conflict detection, and email drafts.

OAuth Flows:

Token refresh, expiry, and re-auth tested for both Google and Microsoft.

Conflict Engine:

Validated against real busy blocks with correct alternative slot ranking.

Combined Workflows:

reply+create and draft+schedule tested atomically with fallback.

AI Drafts:

Contextual replies verified with real email threads; template fallback tested.

Challenges & Lessons Learned

Technical and Process Challenges

OAuth Complexity Across Providers

Google and Microsoft have different OAuth flows, token formats, and refresh behaviors. Required separate lifecycle management.

LLM Output Consistency

AI outputs varied across edge cases. Addressed with Pydantic validation, hybrid parsing, and triple fallback (OpenAI → Groq → templates).

Multi-Intent Parsing

Compound requests required decomposing into ordered actions with atomic rollback on partial failure.

Team Coordination

Parallel AI/backend/frontend development required clear API contracts and commit-by-commit testing.

Future Work

Proposed Improvements

Cloud Deployment (AWS / GCP)

Multi-user access, persistent sessions, and production-grade reliability.

Multi-Turn Dialog & Context Memory

Conversation state tracking, user preference memory across sessions.

Multi-User Auth & Team Calendars

Role-based access and cross-calendar availability for team scheduling.

Advanced NLU

Better handling of ambiguous dates, relative timeframes, and complex compounds.

Additional Providers & Proactive Reminders

Apple Calendar, Zoom, Slack integrations. Push notifications without user initiation.

Thank You

ExecAI — Intelligent Executive Assistant

Rachelle Falcon • Zack Kohlman • Johana Huertas

Florida International University • Spring 2026

References

- Google Calendar API — developers.google.com/calendar
- Gmail API — developers.google.com/gmail
- Microsoft Graph API — learn.microsoft.com/graph
- FastAPI — fastapi.tiangolo.com
- Streamlit — docs.streamlit.io
- Groq LLM API — console.groq.com/docs
- OpenAI API — platform.openai.com/docs
- OAuth 2.0 — oauth.net/2
- Pydantic — docs.pydantic.dev