A**n intelligent automation agent** that handles the entire flow you described. Let's break it into **4 parts**, and I'll explain **how to build it**, **whether it's feasible**, and **which tools/approaches** to use.

**✅ PART 1: Email Reading & Intent Detection (Requirement 1)**

**✔️ Goal:**

Automatically read incoming emails, understand user intent, and extract required details (travel request, card balance, etc.).

**🔧 How to Build:**

* Use **IMAP or Microsoft Graph API** to read emails.
* Use **LLMs (like GPT-4, OpenAI, or Azure OpenAI)** or **open-source NLP models (like spaCy + sentence transformers)** to:
  + Detect **intent** (e.g., “get card balance”)
  + Extract **entities** (e.g., user name, request date, card ID)

**✅ Tools:**

* **Email:** Python + IMAP or Microsoft Graph API
* **NLP/LLM:** LangChain + OpenAI / Hugging Face + spaCy

**✅ PART 2: Agent Decision Routing (Requirement 2)**

**✔️ Goal:**

Route the query to one of the 3 backend systems (Concur, Corporate Card, Invoice App).

**🔧 How to Build:**

* Define a **routing function** (using rules or LLM classifiers) to identify which system the request targets.
* Store the mapping of **intent → target system**.

**✅ Tools:**

* Python function with keyword or embedding-based classification
* LLM-based decision engine (e.g., OpenAI function calling)

**✅ PART 3: Agent Executes Actions in 3 Systems (Requirement 3)**

**✔️ Goal:**

The agent logs into the respective system and performs actions or fetches data.

**🔧 Option 1: If APIs are Available (Preferred & Scalable)**

If Concur, Corporate Card, and Invoice Processing systems offer APIs:

* Authenticate using OAuth2/API tokens
* Fetch or post data using HTTP requests
* Use REST clients in Python (like requests, httpx)

**🔧 Option 2: If APIs are NOT Available — Use RPA (Robotic Process Automation)**

If only a web UI is available (no APIs):

* Use tools like **UIPath**, **Microsoft Power Automate**, or **Python + Playwright/Selenium** to:
  + Log in with stored credentials or SSO
  + Navigate to required UI elements
  + Scrape information
  + Submit forms

**✅ Tools:**

* API: Python + Requests / Postman Collection / LangChain tool agent
* UI Automation: UIPath / Power Automate / Python + Playwright (headless preferred)

**✅ PART 4: Response Generation and Email Reply (Requirement 4)**

**✔️ Goal:**

Use the retrieved information to compose and send a professional email back to the employee.

**🔧 How to Build:**

* Use **LLMs** (OpenAI or local) to format a natural-sounding email from structured data.
* Send email using **SMTP** or **Microsoft Graph API**.

**✅ Tools:**

* Email: Python + SMTP / Microsoft 365 Graph
* LLM: GPT-4 / GPT-3.5 / Claude / Local LLM (e.g., Mistral, LLaMA)

**🤖 CAN THE AGENT LOG IN TO WEB UI?**

Yes, **agents can log in to web UIs** using **headless browser automation**, but with caveats:

| **Option** | **Tools** | **Pros** | **Cons** |
| --- | --- | --- | --- |
| ✅ **API** | Python + Requests / Postman | Secure, scalable | Need API access |
| ✅ **UI Automation (Browser)** | Playwright / UIPath / Selenium | Works even if no API | Fragile, slower, needs UI selectors |
| ❌ **LLM-only** | GPT with browsing tools | Experimental | Can’t log in to real web apps securely |

**⚠️ Important Note**: For real login and action, you need:

* **Service accounts or delegated access**
* Proper **authentication tokens** or **SSO integration**
* Secure storage of credentials (e.g., Azure Key Vault, AWS Secrets Manager)

**🛠 Recommended Stack to Build the Full Agent:**

| **Component** | **Tools** |
| --- | --- |
| Email Read/Send | IMAP/SMTP or Microsoft Graph API |
| Intent Detection | LangChain + OpenAI / spaCy + Transformers |
| Routing Engine | Python + Rules or LLM-based classifier |
| API Access | REST APIs (if available) |
| Web Automation | Playwright (Python/Node), UIPath, Power Automate |
| Agent Orchestration | LangChain Agents, OpenAI Function Calling, or RAG pipeline |
| Secure Credentials | Azure Key Vault / AWS Secrets Manager |
| Logging & Monitoring | ELK stack / Datadog / Custom dashboards |

**💡 Would You Like:**

* A **reference architecture diagram**?
* A **sample agent implementation in Python**?
* A **prototype using OpenAI + Playwright**?

Let me know and I’ll build it accordingly.