■ SurreyLearn Email Automation — Documentation

1. Project Overview

The SurreyLearn Email Automation project is designed to help lecturers and administrators manage incoming assignment submission notifications from SurreyLearn. It connects to Outlook, fetches unread emails from a designated subfolder, extracts key submission information, and prepares it for logging or further processing. Due to Outlook server permissions, the system is designed to safely process but not automatically move or mark emails. Instead, it processes the 10 most recent emails for manual verification and archiving.

2. System Architecture

Components: - Outlook Connector (`outlook_connector.py`): Connects to Outlook and retrieves emails from 'SurreyLearn Submissions' subfolder. - Parser (`parser.py`): Extracts structured data (student name, assignment, submission ID, course code) from email body. - Excel Updater (`excel_updater.py`): Updates Excel file with parsed submissions. - Main Script (`main.py`): Orchestrates fetching, parsing, and processing of emails. - Tests (`tests/`): Unit tests for Outlook connector, parser, and workflow.

3. Workflow

1. User runs the script: python -m src.main 2. Script fetches 10 most recent unread emails from SurreyLearn subfolder. 3. Each email is parsed for structured submission data. 4. Parsed results are displayed in the console and logged. 5. Excel is updated if configured. 6. User manually moves processed emails to an archive folder in Outlook.

4. Key Functions

main.py: - fetch_and_preview_emails(limit=10): Fetch unread emails and preview. - fetch_and_parse_emails(limit=10): Fetch emails and parse into dictionaries. - process_parsed_emails(parsed_emails): Update Excel with parsed submissions. outlook_connector.py: - get_surreylearn_subfolder(): Return 'SurreyLearn Submissions' folder. - fetch_surreylearn_emails(limit=10): Fetch most recent unread emails. parser.py: - parse_surreylearn_email(email): Parse email into structured dictionary. excel_updater.py: - update_excel_with_submission(parsed): Append a submission record to Excel.

5. Installation

1. Clone repository: git clone 2. Create venv: python -m venv venv 3. Activate venv: venv\Scripts\activate (Windows) 4. Install dependencies: pip install -r requirements.txt

6. Dependencies

- Python 3.10+ - pywin32 - pytest - openpyxl

7. Limitations

- Cannot mark emails as read or move them automatically due to server restrictions. - User must manually archive processed emails. - Processes only 10 most recent emails per run.

8. Future Work

- Enable auto-archiving when permissions allow. - Replace Excel with database (PostgreSQL/MySQL). - Add Flask API for parsed submissions. - Extend test coverage.

9. Testing

Run all tests: pytest Example: test_parser_valid_email checks parser correctness.

10. Example Run

python -m src.main Output: Found 5 unread SurreyLearn emails. Email 1 parsed: John Smith - Assignment 1 John Smith - Assignment 1 → Excel updated ■