**Professional Documentation**

**Title: LinkedIn Automation System Using Selenium and NLP**

**Objective**

To create an automated system that interacts with LinkedIn using Selenium and NLP for the following tasks:

1. Login to a LinkedIn account.
2. Search for profiles of university students.
3. Filter and identify relevant profiles.
4. Follow and send personalized messages to selected users.
5. Adhere to defined rules for message sending to avoid overuse and ensure compliance.

**Key Features**

1. **Automation Steps**:
   * Log in to LinkedIn using Selenium.
   * Perform a university-based search via the search bar.
   * Filter search results to identify valid university student profiles.
   * Automate profile following and message sending.
2. **Message Rules**:
   * Send only **2 messages per account** per day.
   * Open a maximum of **2 browser windows at a time**.
   * Send **no more than 10 messages per day** in total.
3. **NLP Integration**:
   * Generate contextually relevant messages using NLP based on profile details.
4. **Error Handling**:
   * Detect and skip irrelevant profiles.
   * Handle browser or network issues gracefully.

**Steps to Implement**

**1. Login Automation**

* Use Selenium to:
  + Navigate to LinkedIn's login page.
  + Enter valid credentials (username and password).
  + Solve potential captchas (manually or using services like 2Captcha).

**2. University Search**

* Navigate to the LinkedIn search bar.
* Input the name of a university.
* Parse search results to access relevant profiles.

**3. Profile Filtering**

* Access the "People" section from the search results.
* Apply filters to identify university student profiles. This can involve:
  + Checking profile descriptions or "About" sections.
  + Using keywords like "student" or specific study programs.

**4. Following and Messaging**

* For selected profiles:
  + Follow the user.
  + Send a personalized message using NLP.
* If a profile does not meet the criteria, skip it.

**5. Rules Enforcement**

* Limit message-sending activities to:
  + 2 messages per account per day.
  + 2 browser windows open simultaneously.
  + A maximum of 10 messages per day across all accounts.

**6. Logging and Reporting**

* Maintain logs of:
  + Profiles visited.
  + Messages sent.
  + Errors encountered.

**NLP Integration**

* Use an NLP library (e.g., Hugging Face Transformers or OpenAI GPT) to generate personalized messages.
* Example:
  + Input: Profile details ("Student at XYZ University, studying Computer Science").
  + Output: *"Hi [Name], I see you’re studying Computer Science at XYZ University. I’d love to connect and discuss opportunities in tech!"*

**Error Handling**

* **Invalid Profiles**:
  + Skip profiles that don’t meet criteria (e.g., incomplete "About" sections).
* **Rate Limits**:
  + Implement delays and randomized actions to mimic human behavior.
* **Browser/Network Issues**:
  + Retry actions with backoff strategies.

**Tools and Libraries**

1. **Selenium**: For browser automation.
2. **NLP Libraries**:
   * Hugging Face Transformers or OpenAI GPT for personalized message creation.
3. **Python Logging**: To track actions and errors.
4. **Data Storage**: CSV or database for logs and reports.

**Ethical Considerations**

* Adhere to LinkedIn’s [Terms of Service](https://www.linkedin.com/legal/user-agreement).
* Avoid spamming or sending unsolicited messages.
* Ensure data privacy and security.

**Entity-Relationship Diagram (ERD)**

**Entities and Relationships:**

1. **LinkedInAccount**:
   * Attributes: account\_id, username, password, messages\_sent
   * Relationship: Logs into and interacts with LinkedIn.
2. **Profile**:
   * Attributes: profile\_id, name, university, status, is\_student
   * Relationship: Associated with accounts through search and messaging.
3. **Message**:
   * Attributes: message\_id, content, status, timestamp
   * Relationship: Sent to profiles by accounts.
4. **SearchQuery**:
   * Attributes: query\_id, university\_name, filters
   * Relationship: Used by accounts to find profiles.