



Lernsituation NetSpy

Szenario

Die Firma NetFusion GmbH ist ein weltweit tätiges Unternehmen, das Filialen in den USA, in Deutschland, England und in Südafrika unterhält. Aufgrund eines aktuellen Vorfalls der Geschäftsspionage durch einen Mitarbeiter überlegt die Firmenleitung, die elektronische Kommunikation aller Mitarbeiter zu überwachen.

Da die Firma einen eigenen und zentralen Email-Server unterhält, können alle ankommenden und abgehenden Emails während der Weiterleitung zwischengespeichert werden. Die Geschäftsleitung wünscht sich eine Java Applikation, die die zwischengespeicherten Emails auf vorher festgelegte Schlüsselwörter untersucht und feststellt, ob diese verdächtig sind oder nicht. Erweist sich eine Email als verdächtig, soll der Absender und der Empfänger der Email in eine Datei gespeichert werden.

Die Firma NetFusion GmbH beauftragt das IT-Unternehmen HHBKProComp GmbH, eine Machbarkeitsstudie für dieses Vorhaben zu erstellen. Erster Meilenstein der Machbarkeitsstudie ist die Fertigstellung eines UML Klassendiagramm. Nach Abnahme des Klassendiagramms können Sie einen ersten Prototypen in Java erstellen.

Hier eine Beschreibung der gewünschten Funktionalitäten.

Basis- Funktionalitäten von NetSpy

- ✓ Alle Emails in einem bestimmten Verzeichnis nach Suchwörtern durchsuchen
- ✓ Suchwörter aus einer Blacklist lesen
- ✓ Einstufung „verdächtig“ nach bestimmter Anzahl von Treffern innerhalb einer Email
- ✓ Quarantäneordner für „verdächtige“ Emails einrichten und „verdächtige“ Emails dahin verschieben
- ✓ Für jeden Suchlauf eine Logdatei mit Infos über „verdächtige“ Emails schreiben. Die Logdatei sollte pro Email folgende Informationen enthalten: (Absender, Empfänger, Sendedatum, Betreff, gefundene Suchwörter mit Angabe der Trefferzahl pro Suchwort)
- ✓ Groß- und Kleinschreibung ignorieren
- ✓ ASCII-Code ersetzen (Umlaute / Sonderzeichen)
- ✓ Suchwörter in zusammengesetzten Wörtern erkennen (Teilstrings)



Optionale Funktionalitäten

- ✓ Zeilenumbruch, Leerstellen ignorieren
- ✓ Suchbegriffen unterschiedliche Prioritäten zuordnen
- ✓ Berücksichtigung von Jokerzeichen im Suchwort
- ✓ Automatisch nach ähnlichen Wörtern suchen (regular expressions)
- ✓ Blacklist dynamisch aus der Anwendung heraus verwalten (löschen, ändern, hinzufügen)
- ✓ Whitelist für Email-Adressen (Absender, Empfänger) verwenden
- ✓ Whitelist für Email-Adressen dynamisch aus der Anwendung heraus verwalten
- ✓ Option für Suche als ganzes Wort oder als Teilstring zur Verfügung stellen
- ✓ HTML-Teil bei der Suche ignorieren
- ✓

Anforderungen an das UML-Klassendiagramm

Das UML-Klassendiagramm soll alle verwendeten Klassen und die Beziehungen der Klassen untereinander enthalten. Die Beziehungen müssen mit den Mitteln der UML verständlich dokumentiert sein. Sie dürfen dazu auch durchaus Beziehungs- und Rollennamen verwenden.

Es folgt eine Liste von Klassen, die auf jeden Fall in Ihrem Klassendiagramm auftauchen müssen. Die Beschreibung der Klassen enthält nur zentrale Aspekte und hat keinen Anspruch auf Vollständigkeit. Natürlich dürfen Sie die Liste der Klassen auch erweitern.

- **Klasse NetSpy**

Die Klasse NetSpy enthält die Klassenmethode main() und startet in dieser Methode die NetSpy-Anwendung. Dazu gehört u.a., die Blacklist-Datei einzulesen.

- **Klasse EmailContainer**

Die Klasse EmailContainer stellt ein Attribut zur Speicherung aller Emailobjekte zur Verfügung. Natürlich verfügt sie auch über Eigenschaften zum Hinzufügen und Löschen von Emailobjekten.



- **Klasse Email**

Die Klasse Email speichert alle relevanten Eigenschaften einer Email (Absender, Sender, Betreff, Inhalt, etc.). Der Zugriff auf diese Eigenschaften erfolgt über öffentliche setter- und getter-Methoden.

- **Klasse EmailHandler**

Die Klasse EmailHandler erzeugt Emailobjekte, holt aus dem Inhalt einer Emaildatei die relevanten Eigenschaftswerte eines Emailobjektes, setzt die Eigenschaftswerte eines Emailobjektes und stellt eine Methode zum Scannen des Email-Inhaltes zur Verfügung.

- **Klasse BlackList**

Die Klasse BlackList speichert alle Suchwörter in einer ArrayList. Der Zugriff auf die ArrayList erfolgt über öffentliche setter- und getter-Methoden.

- **Klasse FileManager**

Die Klasse EmailHandler nutzt die Fähigkeiten der Klasse FileManager, um auf alle Dateinamen eines Email-Verzeichnisses zugreifen zu können. Auch das sequentielle Lesen einer Emaildatei oder einer Blacklistdatei ist Aufgabe des FileManagers.

- **Klasse LogManager**

Die Klasse LogManager bietet eine Methode zum Schreiben eines Eintrags in die Logdatei. Dabei setzt sie hauptsächlich den auszugebenden Texteintrag für die Logdatei zusammen. Das sequentielle Schreiben in die physikalische Logdatei überlässt sie der Klasse FileManager.