UML Class Diagram

«interface» IConfigurationManager
- + getNumberOfGroup(): int
- + getVictims(): Person[]
- + getMessages(): Message[]
- + getSmtpPort(): int
- + getSmtpServerAddress(): String

«interface» ISmtpClient
- + sendMessage(Message): void

PrankGenerator
- MIN_VICTIMS: int
- LOG: Logger
- configurationManager: IConfigurationManager
- + generateGroups(Person[], int): List<Group>
- + generatePranks(): List<Prank>

Message
- cc: Email[]
- to: Email[]
- subject: String
- bcc: Email[]
- body: String
- from: Email
- + getBcc(): Email[]
- + setCc(Email[]): void
- + getTo(): Email[]
- + setBody(String): void
- + setSubject(String): void
- + getBody(): String
- + getSubject(): String
- + setFrom(Email): void
- + getCc(): Email[]
- + setBcc(Email[]): void
- + getFrom(): Email
- + setTo(Email[]): void

Person
- firstName: String
- lastName: String
- address: Email
- + getLastName(): String
- + getFirstName(): String
- + getAddress(): Email

Email
- email: String
- + toString(): String
- + validateEmail(String): boolean

ConfigurationManager
- instance: ConfigurationManager
- victims: Person[]
- messages: Message[]
- smtpPort: int
- numberOfGroups: int
- LOG: Logger
- smtpServerAddress: String
- parseEmail(String): void
- + getVictims(): Person[]
- parseMessages(String): void
- + getSmtpServerAddress(): String
- + getMessages(): Message[]
- + getInstance(): ConfigurationManager
- + getSmtpPort(): int
- + getNumberOfGroup(): int

SmtpClient
- smtpServerAddress: String
- LOG: Logger
- writer: PrintWriter
- ~ config: IConfigurationManager
- smtpServerPort: int
- reader: BufferedReader
- socket: Socket
- + sendMessage(Message): void

Prank
- victimSender: Person
- victimRecipients: List<Person>
- message: Message
- + getVictimSender(): Person
- + getVictimRecipients(): List<Person>
- + generateMailMessage(): Message
- + setMessage(Message): void
- + setVictimSender(Person): void
- + getMessage(): Message
- + addVictimRecipients(List<Person>): void

Group
- members: List<Person>
- + getMembers(): List<Person>
- + addMember(Person): void

Relationships:
- generate v (PrankGenerator 1 — * Prank)
- generate > (Prank 1 — 1 Message)
- < validate (Person 1..* — 1 Email)
- form v (Person 3..* — 1 Group)