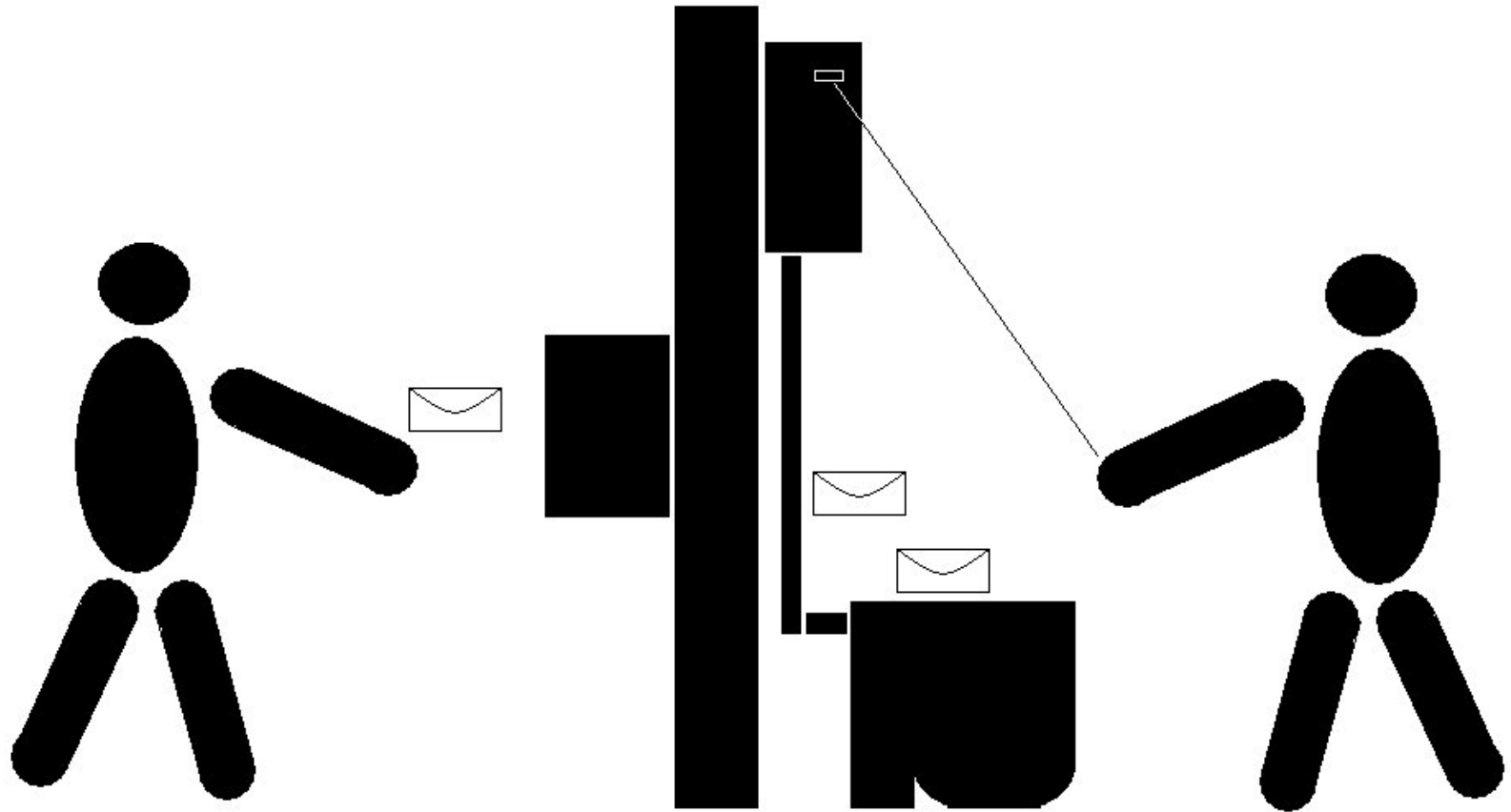


**Welcome to Spamgourmet**

# Welcome to Spamingourmet



# Spamgourmet

- **Mike Schwarz (ms111@hdm-stuttgart.de)**
- **Matthias Frenzel (mf044@hdm-stuttgart.de)**
- **Markus Zobel (mz014@hdm-stuttgart.de)**
- **Norman Pohl (np012@hdm-stuttgart.de)**

**Betreut wurde dieses Projekt von  
Prof. Dr. Martin Goik**

# Projektbeschreibung

**Der Anbieter <http://www.spamgourmet.com> bietet die Möglichkeit zur Definition temporärer Emailadressen, welche mit einer nur dem Benutzer bekannten, persönlichen Email Adresse verbunden sind.**

**Auf diese Weise kann man beispielsweise für die License Key Zusendung zur Evaluierung einer Software anstelle der eigenen Emailadresse eine solche Temporäradresse verwenden und auf diese Weise die spätere Zusendung von Werbemüll vermeiden.**

**Die Idee des Projektes besteht nun darin, an der HdM einen eigenen Service zu etablieren.**

# Anforderungen

**Anwendung zum Erstellen und Verwalten von temporären Wegwerf-Emailadressen.**

**Sortieren nach Filterregeln:**

- **Maximale Anzahl zu empfangender Emails.**
- **Gültigkeitszeitraum der temporären Adresse.**
- **Emails von bestimmten Absenderdomains akzeptieren bzw. verwerfen.**

**Weiteres Verfahren:**

- **Weiterleitung an den Emailserver der HdM**
- **Emails, die nicht weitergeleitet werden sollen, werden verworfen.**

# Realisierung der Applikation

Die Realisierung wurde in 3 Teile aufgeteilt.

- **Weboberfläche** (*Markus und Matthias*)
- **Datenbank** (*Mike*)
- **Service** (*Norman*)

# Weboberfläche:

## Anforderungen

- um persönliche Einstellungen und Adressen zu erstellen und verwalten
- Datenbankbindung um die Adressen und Einstellungen zu speichern
- Benutzerauthentifizierung über HdM LDAP Server

## Realisierung

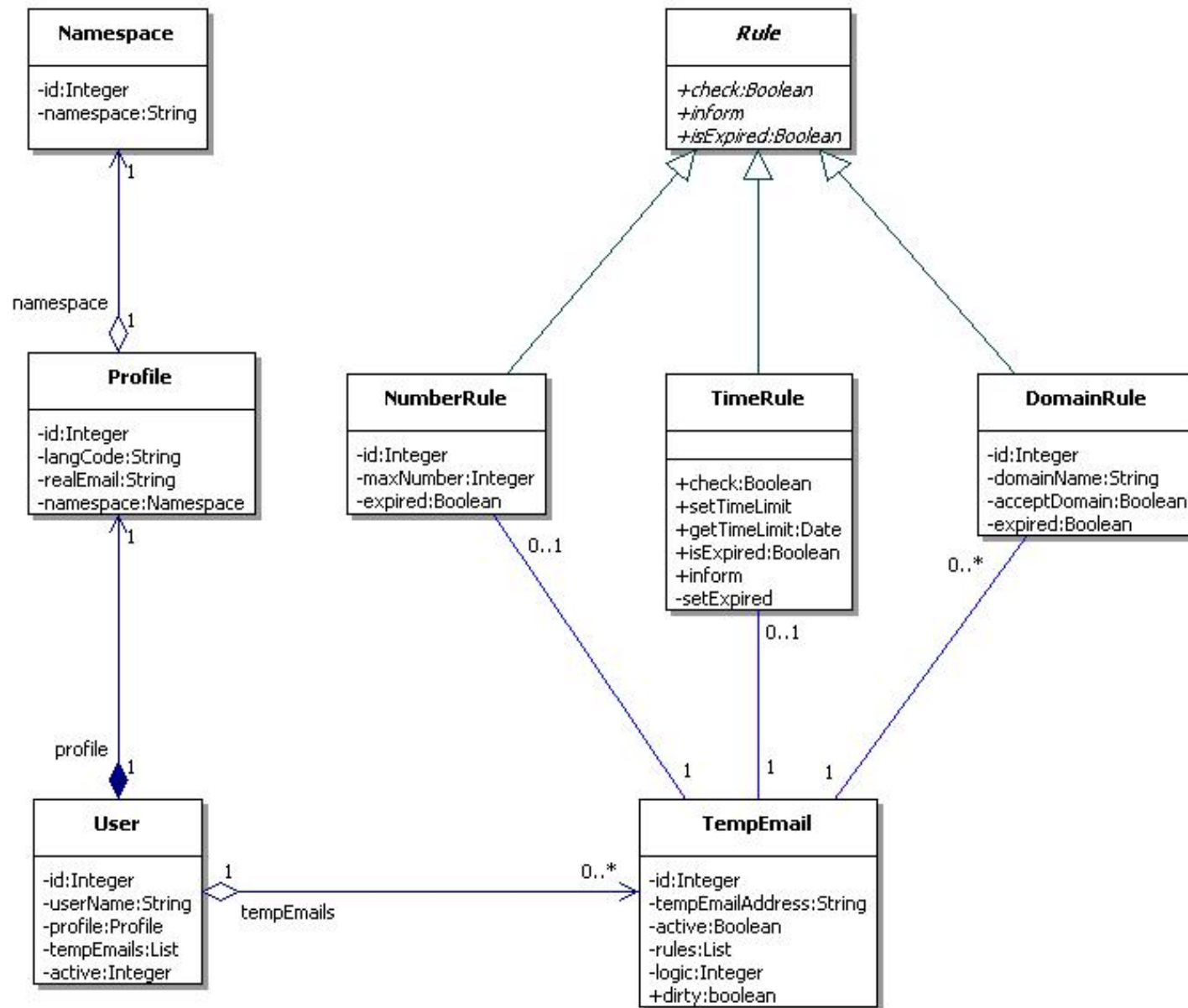
- Struts als MVC Framework
- Basic Modus, Extended Modus

# Datenbank / Persistenz

- **Die Datenbankbindung wird mit Hilfe des Hibernate Frameworks realisiert.**
- **Für die Verwaltung der Benutzerdaten sind Java Bean Klassen zuständig.**
- **Die Klassen und deren Beziehungen zueinander werden auf ein entsprechendes Datenbankschema gemappt.**



# Datenbank / Persistenz (Continued)



# Datenbank / Persistenz (Continued)

- **Über das “User Objekt” können alle Daten erreicht werden.**
- **Für die Weboberfläche und den Service existiert eine Schnittstelle, die den “User” aus der Datenbank liest und auch speichert.**
- **Der “User” kann dabei entweder über den Benutzernamen oder über seinen Namespace identifiziert werden.**

# Service

## Architektur

- **Der Service wurde mittels James, einem in Java implementierten Email-Server realisiert.**
- **Dieser kann durch sogenannte “Mailets” erweitert werden. Mailets sind kleine Plugins, die für die Email-Verarbeitung zuständig sind.**
- **Die Mailets werden in Queues organisiert, die eine eingehende Email durchläuft. Dadurch kann der Server leicht um Funktionalität erweitert werden.**

# Service (Continued)

## Funktion

- **Legt (auf Wunsch) automatisch neue temporäre Email-Adressen anhand der Default-Einstellung an, wenn eine Email an eine noch nicht vorhandene Email-Adresse eintrifft.**
- **Wertet Regeln anhand gewählter Logik aus, und prüft ob sie "abgelaufen" sind (z.B. maximale Anzahl der weiterzuleitenden Emails erreicht). Die temporäre Email-Adresse wird dann deaktiviert.**
- **Änderungen in der Datenbank speichern.**

# Verwendete Software

- **Java**
- **Datenbank**
  - **Mysql**  
**Open Source Datenbank**
  - **JDBC-Connector**  
**Verbindung zwischen MySQL und Hibernate**
  - **Hibernate**  
**Persistenzframework für OR-Mapping**
- **Webanwendung**
  - **Tomcat**  
**Servlet Container von Apache**
  - **Struts**  
**MVC Framework**

# Verwendete Software (Continued)

- **James**  
**Email-Server, der sogenannte “Mailets”**  
**unterstützt.**

# Verwendete Tools

- **Dokumentation**
  - **Javadocs**
  - **Docbook**
- **Programmierertools**
  - **Eclipse**
  - **Subversion**
  - **Checkstyle**
  - **Ant**

# Highlights

- **Die Tooltips der Spamingourmet Applikation werden direkt aus der Dokumentation über ein XSLT Stylesheet in eine separate XML Datei extrahiert und per selbst entwickelter Taglib angezeigt.**
- **Die verschiedenen Teile der Anwendung werden über die Datenbank verknüpft. Dadurch ist es möglich, das der Server im laufenden Betrieb durch den User konfiguriert werden kann, und Änderungen “on-the-fly” aktiv werden.**
- **2 verschiedene Sprachen werden unterstützt, auch bei den Datumsangaben. Weitere Sprachen können einfach hinzugefügt werden.**
- **Für uns neue Technologien: Hibernate, Struts...**



# Ausblick

**Automatisches Anlegen der Emails beim ersten Empfang werden wir eventuell wieder verwerfen. Oder eine Lösung per Tags. z.B. abc steht für NumberRule mit Count = 5.**

**Mögliche Erweiterungen:**

- **Mozilla Extension**
- **Reply mit der Tempmail Adresse**
- **SOAP Schnittstelle**
- **Handy Client**