

R&D Projekt

Testplan

Version 1.1

**Bezeichnung**: Erstellung einer JRZ Demodatenbank (DemoDB)

**Projektschlüssel:** RD16-03

**Betreuer**: DI Eduard Hirsch, DI Fabian Knirsch, BSc

**Kurzbeschreibung**: Konvertierung verschiedener Smartmeter Messdaten und Ablage in einer gemeinsamen Datenbank mit rollenbasiertem Zugriff.

**Beteiligte** **Firma**: Salzburg AG

**Studenten**: Isidor Reimar Klammer, BSc.

Maximilian Unterrainer, BSc.

Christopher Wieland, BSc.

Puch/Salzburg, 25. Mai 2017

Inhalt

[Historie 3](#_Toc483596827)

[Einführung 4](#_Toc483596828)

[Zweck 4](#_Toc483596829)

[Umfang 4](#_Toc483596830)

[Beziehungen zu anderen Dokumenten 4](#_Toc483596831)

[Testbezeichner 4](#_Toc483596832)

[Referenzen 5](#_Toc483596833)

[Systemüberblick 5](#_Toc483596834)

[Module und deren Funktionen 6](#_Toc483596835)

[Merkmale 6](#_Toc483596836)

[Zu testende Merkmale 6](#_Toc483596837)

[Funktionale Anforderungen 6](#_Toc483596838)

[Nicht zu testende Merkmale 6](#_Toc483596839)

[Abnahme- und Testendekriterien 6](#_Toc483596840)

[Vorgehen 7](#_Toc483596841)

[Unit-Tests 7](#_Toc483596842)

[Komponenten- und Integrationstest 7](#_Toc483596843)

[Funktionstest 7](#_Toc483596844)

[Anforderungen an Hardware und Software (Server/Client) 8](#_Toc483596845)

[Server - Hardware 8](#_Toc483596846)

[Server - Software 8](#_Toc483596847)

[Server - Konfiguration 8](#_Toc483596848)

[Client 8](#_Toc483596849)

[Testfälle 8](#_Toc483596850)

[Komponententest 8](#_Toc483596851)

[Integrationstest 8](#_Toc483596852)

[Funktionstest 9](#_Toc483596853)

[Leistungstest 9](#_Toc483596854)

[Härtetest 9](#_Toc483596855)

[Volumentest 9](#_Toc483596856)

[Sicherheitstest 9](#_Toc483596857)

# Historie

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Änderung** | **Änderer** |
| 1.0 | 15.4.2017 | Ersterstellung | M |
| 1.1 | 20.5.2017 | Ergänzung weiterer Testfälle | M |

# Einführung

Dieses Dokument beschreibt den Testplan für das Research & Development Projekt „SmartValAPI“, welches im Zuge des ITSM Studiums an der FH Salzburg von I. Klammer, M. Unterrainer und C. Wieland umgesetzt wird.

## Zweck

Dieser Testplan gibt dem Auftraggeber und den Entwicklern einen Überblick, welche Schritte wie abgeschlossen werden müssen, um das Projekt erfolgreich abzuschließen und die Software produktiv einzusetzen.

## Umfang

Dieser Testplan basiert auf den Vorgaben des Standards IEEE 829-2008 [1] und wurde an jenen Stellen, die auf Grund des Projektes nicht anwendbar sind, wie zum Beispiel genaues Ausspezifizieren des Testteams oder Einflüsse parallel umgesetzter Projekte entsprechend gekürzt.

## Beziehungen zu anderen Dokumenten

Als Basis für die Umsetzung diesen der Auftrag des JRZ, die Protokolle der Besprechungen und Feedbackrunden und vor allem das Pflichtenheft. Aus diesen Inputs und der Erfahrung der Entwickler ergeben sich die Testfälle.

## Testbezeichner

Tests werden mit eindeutigen Bezeichnern in der Form   
X-NN-V  
identifiziert. Die Bedeutung der Kürzel ist in Tabelle 1 erklärt.

|  |  |  |
| --- | --- | --- |
| Kürzel | Ausprägungen | Bedeutung |
| X | F, I, K , L | I – Integrationstest K – Komponententest F – Funktionstest L – Lasttest |
| NN | 00 – 99 | Laufende Nummer innerhalb von X |
| V | A – Z | Variante eines Tests, bezüglich des Ergebnisses und der Umgebungsbedingungen |

Tabelle 1Testbezeicher

## Referenzen

IEEE Std. 829-2008: Software Engineering Technical Committee of the IEEE Computer Society. IEEE standard for software test documentation, USA, 2008.

# Systemüberblick

Abbildung 1(Vorsicht, die Grafik ist noch nicht richtig!) zeigt die Komponenten von SmartValAPI. In der Folge werden die Funktionen beschrieben.

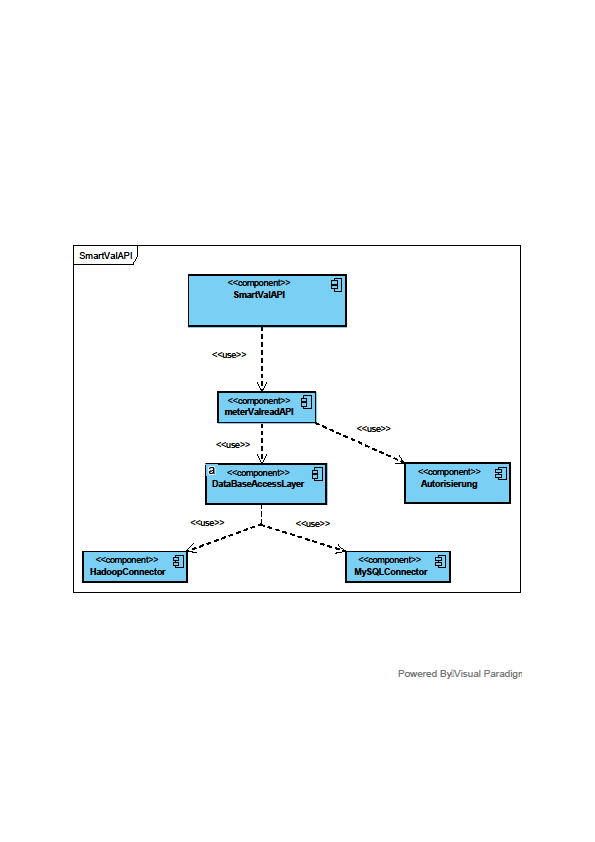


Abbildung 1Komponenten von SmartValAPI

## Module und deren Funktionen

Administrationsschnittstelle: Funktionen ergänzen

Benutzerschnittstelle: Funktionen ergänzen

Datenbankanbindung (MySQL und MongoDB): Funktionen ergänzen

Zugriffsverwaltung: Funktionen ergänzen

# Merkmale

## Zu testende Merkmale

## 

## Funktionale Anforderungen

## Nicht zu testende Merkmale

Nicht explizit getestet werden die grundsätzlichen Funktionen der eingesetzten Fremdkomponenten, wie zum Beispiel MySQL, die JDBC Anbindung und Hibernate, da davon ausgegangen werden kann, dass diese Systeme vielfach im Einsatz sind und von den jeweiligen Herstellern bereits ausgiebig getestet wurden. Hingegen werden die Ergebnisse, welche von diesen Komponenten geliefert werden, und vor allem das Zusammenspiel mit SmatValAPI sehr wohl getestet.

# Abnahme- und Testendekriterien

Auftretende Fehler werden entsprechend der Einteilung in Tabelle 2 kategorisiert. Das Ende des Teststadiums kann erst erreicht werden, wenn alle Fehler der Kategorien 1 und 2 behoben wurden. Ebenso kann eine Abnahme nur Die Abnahme kann erst dann erfolgen, beziehungsweise das Ende des Teststadiums ist erst erreicht, wenn alle Fehler der Priorität 1 und 2 behoben wurden. Die Zuordnung zu einer Klasse unterliegt dem Konsens des Entwicklerteams und des Auftraggebers.

|  |  |
| --- | --- |
| Klasse | Beschreibung |
| Fatal (Priorität 1) | Absturz oder Dienstunterbrechung, es kann nicht weitergearbeitet werden, ein Eingriff am Server ist notwendig und Änderungen in der Software die Folge. |
| Kritisch (Priorität 2) | Fehler, die die Verwendbarkeit der gelieferten Daten zerstören. |
| Gering (Priorität 3) | Fehler, welche die Leistung geringfügig einschränken. |
| Unwesentlich (Priorität 4) | Fehler, die zwar auftreten, aber keine erkennbaren Schäden verursachen, werden mit niedriger Priorität behoben. |

Tabelle 2 - Fehlerkategorien

# Vorgehen

## Unit-Tests

Auf Klassen- und Methodenebene werden die Tests mittels JUnit-Tests getestet. Diese Tests werden während, spätestens aber zum Ende der Implementierung geschrieben und decken neben den Standardfällen („Happy Path“) Situationen in denen Ausnahmen ausgelöst werden und bewusstes Fehlverhalten provoziert wird ab. Besonderes Augenmerk wird auf Grenzsituationen gelegt, zum Beispiel die Berechnung des Durchschnitts einer leeren Menge.

## Komponenten- und Integrationstest

Die Komponenten- und Integrationstests werden von der Basis zur Gesamtkomponente hin durchgeführt („bottom-up“). Das Mapping der Datenbankinhalte auf POJOs ist hier der zuunterst liegende Teil, in der Folge wird durch Schnittstellentests (lokale Installation) die Funktionsfähigkeit getestet. Parallel dazu erfolgen die LDAP-Tests gegen den Deployment-Server (landsteiner.fh-salzburg.ac.at). Anschließend erfolgen die Tests gegen die REST Schnittstelle, hier liegt der Schwerpunkt auf dem Zusammenspiel der einzelnen Komponenten und den Ergebnissen der Schnittstelle.

Hier einen Baum der Komponenten rein, evtl auch mit Methodennamen.

## Funktionstest

Die Musterauswertungen werden, wie sie im Pflichtenheft angeführt sind, getestet. Die Mindestanforderung ist, dass jede der drei Auswertungen für einen konkreten Test korrekte Ergebnisse liefert. Voraussetzung dafür ist die Beschickung der Datenbankinstanz mit jenen Daten, die in den Installationsskripts (Namen der beiden .sql Dateien rein) enthalten sind.

# Anforderungen an Hardware und Software (Server/Client)

## Server - Hardware

Für die Tests und in der Folge die Ausrollung der Software steht eine virtuelle Maschine am Blade-Server des JRZ zur Verfügung. Ausgestattet ist dieser Rechner mit 2 Intel Xeon E5-2620 Prozessoren mit 2GHz, 4 GB Hauptspeicher und 80 GB Plattenplatz.

## Server - Software

Zusätzlich zur eigentlichen Software wird am Server als Betriebssystem Windows Server 2012 R2 installiert. Als RDBMS wird ein MySql Server 5.7.17 eingesetzt, zur einfacheren Überprüfung von Daten am Server wird die MySQL Workbench 6.9 verwendet. Für die optionale NoSQL-Verwaltung der Meterdaten wird MongoDB in der Version 3.4 installiert.

## Server - Konfiguration

Dieser Rechner ist über das Internet unter der IP-Adresse 193.170.119.66 oder dem Hostnamen landsteiner.fh-salzburg.ac.at ansprechbar, erreichbar mit dem Remote Desktop Protokoll. Die REST Schnittstelle horcht am Port 8080.

## Client

SmartValAPI stellt eine REST-Schnittstelle zur Verfügung, die Anforderung an den Client ist daher gering. Zugriff über das Internet auf den Port des Servers ist zu gewährleisten. Jeder beliebige REST-Client kann eingesetzt werden, getestet wird mit dem in IntelliJ IDEA integrierten REST-Client und dem frei verfügbaren SOAPUI in der Version 5.3.0. Örtlich gibt es keine Einschränkungen.

# Testfälle

Die folgende Übersicht führt alle zu erfüllenden Testfälle an.

## Komponententest

## Integrationstest

|  |  |
| --- | --- |
| Testbezeichner | Testbeschreibung |
| I-00-A | REST Abfrage mit Benutzer auf eigene Daten |
| I-00-B | REST Abfrage mit Benutzer auf fremde Daten |
| I-00-C | REST Aufruf der Administrations-Schnittstelle mit einem Benutzer ohne Administratorberechtigung |
| I-01-A | REST Abfrage ohne laufende Datenbank |
| I-01-B | REST Abfrage ohne laufenden Applikationsserver |
| I-01-C | REST Abfrage ohne erreichbaren LDAP-Server |
| I-02-A | REST Abfrage auf eigene Daten in Zeitraum ohne Messdaten |
| I-03-A | REST Abfrage auf Zeitraum mit „ZeitraumVon“ > „ZeitraumBis“ |
| I-03-B | REST Abfrage ohne Benutzerkontext |
| I-04-A | REST Aufruf mit einer weiteren Abfrage die nicht funktioniert |
| I-05-A | REST Aufruf mit falschem Pfad im Parameter |
| F-00-A | Administratorschnittstelle Abfrage bestehender Kundendaten |
| F-00-B | Administratorschnittstelle Abfrage nicht bestehender Kundendaten |
| F-01-A | Administratorschnittstelle Update Kundendaten (ist das ein Teil der Admin oder der Benutzerschnittstelle) |
| F-02-A | Administratorschnittstelle Abfrage bestehender Kundendaten |
| F-02-B | Administratorschnittstelle Abfrage nicht bestehender Kundendaten |
| F-03-A | Administratorschnittstelle Update Kundendaten (ist das ein Teil der Admin oder der Benutzerschnittstelle) |
|  |  |
|  |  |
|  |  |
|  | … alle anderen der Adminschnittstelle eintragen |
|  |  |
|  |  |

Funktionstests – DAO

|  |  |
| --- | --- |
| Testbezeichner | Testbeschreibung |
| F-00-A | MySQL Datenbank Zugriff Mapping vorhandener Kundendaten |
| F-00-B | MySQL Datenbank Zugriff auf nicht vorhandener Kundendaten |
| F-01-A | MySQL Datenbank Zugriff Mapping vorhandener Meterstammdaten |
| F-01-B | MySQL Datenbank Zugriff auf nicht vorhandener Meterstammdaten |
| F-02-A | MySQL Datenbank Zugriff Mapping vorhandener Meterstammdaten |
| F-02-B | MySQL Datenbank Zugriff auf nicht vorhandener Meterstammdaten |
| F-03-A | MySQL Benutzerschnittstelle Zählernummern zur Kundennummer |
| F-04-A | MySQL Datenbank Zugriff Mapping vorhandener meter\_data |
| F-04-B | MySQL Datenbank Zugriff auf nicht vorhandene meter\_data |
| F-04-C | MySQL Datenbank Zugriff Mapping Liste von vorhandenen meter\_data auf Zeitbereich eingeschränkt |
| F-04-D | MySQL Datenbank Zugriff Mapping Liste von nicht vorhandenen meter\_data auf Zeitbereich eingeschränkt |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Funktionstests - LDAP

|  |  |
| --- | --- |
| Testbezeichner | Testbeschreibung |
| F-00-A | Identifizierung mit bestehendem Benutzer und richtigem Passwort |
| F-00-B | Anmeldung mit Benutzer und falschem Passwort |
| F-00-C | Anmeldung mit gesperrtem Benutzer |
| F-00-D | Anmeldung mit Administrator Benutzer und richtigem Passwort |
| F-01-A | Anmeldung mit unerreichbarem LDAP-Server |
| F-03-A | Abfrage der zugeteilten Smart-Meter zu einer Benutzerkennung |
| F-04-A |  |

# Leistungstest

|  |  |
| --- | --- |
| Testbezeichner | Testbeschreibung |
| L-00-1 |  |

## Härtetest

## Volumentest

Die Volumentests werden mit SoapUI mit der Funktion „Load Test“ von mehreren Rechnern aus parallel gegen den Testserver (landsteiner.fh-salzburg.ac.at durchgeführt.

## Sicherheitstest

Hier die LDAP Tests gesammelt einsortieren.

Offen: zum Schluss alles konsistent durchnummerieren.