

Fallstudie: **Weather Data Analytics (WDA)**

Dozenten: Roland Gisler & Jordan Šućur

Version 0.9, April 2022

Inhalt

1. Ausgangslage	3
2. Schnittstelle	3
3. Funktionale Anforderungen	3
3.1. Abfragen	4
3.2. Verwaltung von Stammdaten	4
4. Nichtfunktionale Anforderungen	5
5. Ablauf	5
5.1. Phase 1	5
5.2. Phase 2	5
6. Dokumentation	6
7. Abgaben	6
7.1. Phase 1	6
7.2. Phase 2	6
8. Präsentation (Vorführung)	6
9. Organisatorisches	7
9.1. Projektaufwand	7
9.2. Relevanter Datenbestand	7

1. Ausgangslage

Ein junges Unternehmen namens WDA GmbH braucht eine Applikation, welche die Wetterdaten für diverse Ortschaften in der Schweiz bewirtschaften kann. Am Anfang soll die Anzahl Ortschaften klein gehalten werden, jedoch soll die Applikation danach in der Lage sein, eine beliebige Anzahl von Ortschaften zu berücksichtigen.

Die Wetterdaten werden von einem Provider bezogen, der sie über einen Web-Service (REST) und in einem proprietären Format zur Verfügung stellt. Dazu kommt noch, dass solche Daten nur für eine begrenzte Zeitperiode zur Verfügung stehen¹ (z.B. letzte 30 Tage). Die WDA GmbH möchte diese Daten so aufbereiten, dass sie anschliessend nach diversen Kriterien und für eine beliebige Zeitperiode gefiltert und an interessierte Kunden verkauft werden können. Ausserdem sollen die aktuellen Wetterdaten der letzten 24 Stunden allen interessierten über einen entsprechenden Web-Service kostenlos zur Verfügung gestellt werden.

2. Schnittstelle

Über die vom erwähnten Provider zur Verfügung gestellte REST-Schnittstelle sind folgende Abfragen möglich:

- alle Ortschaften, für welche die Wetterdaten verfügbar sind,
- die zuletzt gemessene Wetterdaten für eine konkrete Ortschaft und
- alle verfügbaren Wetterdaten für die angegebenen Ortschaft und Jahr.

Genauere Angaben zu einzelnen Abfragen können direkt auf der folgenden Adresse angesehen werden:

<http://swde.el.eee.intern:8080/weatherdata-provider>

Hinweis:

Der Zugriff ist nur innerhalb des HSLU-Netzwerkes bzw. über eine VPN-Verbindung möglich.

3. Funktionale Anforderungen

Die zu erstellende Applikation besteht (grob gesagt) aus den folgenden Teilen:

- Ein Reader-Modul, welches die Wetterdaten beim Provider über den REST Web Service bezieht und für weitere Verarbeitung zur Verfügung stellt.
- Ein Business-Modul, welches die
 - von dem Reader-Modul gelesenen Daten übernimmt und in einer «normalisierten» Form dauerhaft speichert,
 - verfügbaren Daten nach diversen Kriterien filtert und in unterschiedlichen Formaten (*csv, xml, json, pdf etc.*) für den Verkauf an interessierte Kunden zur Verfügung stellt.
- Ein UI Modul, das von Mitarbeitern der WDA GmbH für die Bewirtschaftung der Wetterdaten(-datenbank) verwendet wird.
- **Fakultativ:**
Ein Web Service, welcher die Wetterdaten für eine angegebene Ortschaft für letzten 24 Stunden jedem unentgeltlich zur Verfügung stellt, der an diesen Daten in irgendeiner Art und Weise interessiert ist².

¹ Es muss damit gerechnet werden, dass die "älteren" Daten laufend entfernt werden.

² Diese Daten stehen sowohl den Kunden als auch den «nicht Kunden» von WDA GmbH unentgeltlich zur Verfügung.

3.1. Abfragen

Die zu erstellende Applikation soll folgende Abfragen durchführen können:

ID	Abfrage
A01	Für welche Ortschaften stehen die Wetterdaten zur Verfügung?
A02	Wie sehen die einzelnen, konkreten Werte für Temperatur, Luftdruck und Feuchtigkeit für eine angegebene Ortschaft während einer angegebenen Zeitperiode ³ aus?
A03	Wie sehen die Durchschnittswerte für Temperatur, Luftdruck und Feuchtigkeit für eine angegebene Ortschaft in einer angegebenen Zeitperiode ⁴ aus?
A04	Wie sahen die Maximal- und Minimalwerte für Temperatur, Luftdruck und Feuchtigkeit für eine angegebene Ortschaft während einer angegebenen Zeitperiode ⁵ aus?
A05	In welchen Ortschaften herrschte zu einem bestimmten Zeitpunkt die höchste bzw. die tiefste Temperatur (gleiche Abfrage auch für Luftdruck und die Feuchtigkeit) ⁶ .

Tabelle 1 - Die zu realisierenden Abfragen

3.2. Verwaltung von Stammdaten

Neben den zuvor aufgeführten Abfragen müssen auch die folgenden Funktionalitäten verfügbar sein:

ID	Funktionalität
A06	Verwaltung von Daten der eigenen Benutzer: CRUD (C reate, R ead, U ppdate, D eleate).
A07	Export von bestellten Daten (gemäss den Abfragen in der obigen Tabelle) in einem Format Ihrer Wahl (z.B. xml, csv, json, pdf etc.). Natürlich dürfen auch mehrere Formate unterstützt werden, um auf die Wünsche von Kunden möglichst optimal eingehen zu können.
A08	Der Zugriff auf die Applikation muss mit einem Login ⁷ geschützt werden.
A09	Fakultativ: Minimale Verwaltung von Kundendaten (CRUD).

Tabelle 2 - Weitere funktionale Anforderungen

³ z.B. während den letzten 48 Stunden

⁴ z.B. während einem Monat wie Februar, März oder ähnlich

⁵ z.B. während einem bestimmten Monat wie Januar, Februar usw.

⁶ Sollte der maximaler bzw. der minimale Wert bei mehreren Ortschaften vorkommen, müssen alle angegeben werden.

⁷ Die Verschlüsselung von Zugangsdaten wird nicht verlangt.

4. Nichtfunktionale Anforderungen

Die zu erstellende Applikation muss (erst in der zweiten Phase) als verteilte Applikation realisiert werden. Dabei muss es möglich sein, dass der Client⁸ mehrfach und verteilt ausgeführt wird, während das Backend **nur einmal** installiert und ausgeführt wird. Die Kommunikation zwischen Client- und Backend soll mit RMI realisiert werden.

Für die dauerhafte Ablage von Daten muss ein DBMS verwendet werden. Bei der Implementierung des Clients gibt es keine Vorgaben. Beachten Sie dabei: Es stehen die **fachliche Funktion** und Architektur im Vordergrund, es geht nicht um ein «schönes» UI!

5. Ablauf

Die Applikation wird in zwei Phasen entwickelt.

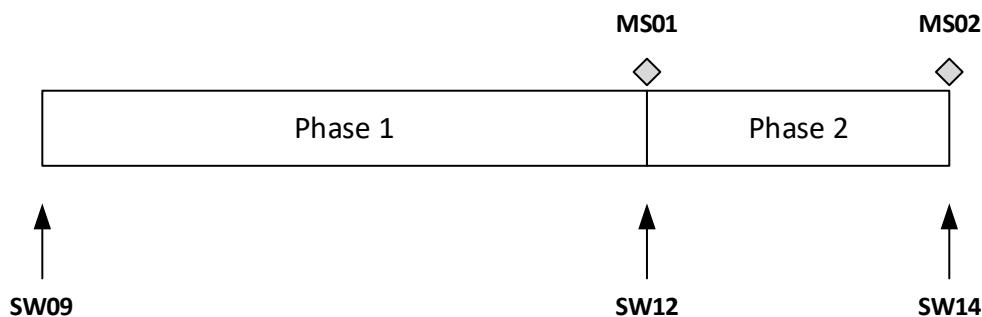


Abbildung 1 - Projektphasen und Meilensteine⁹

Die Dokumentation wird laufend geführt. Es wird auch verlangt, dass die Applikation mit Hilfe von automatisierten Unit- und Integrationstests laufend getestet wird. Am Schluss sind manuelle Systemtests für die Verteilung durchzuführen. Die Planung und Koordination des gesamten Projektes erfolgt online über das Issue-/Tasksystem von GitLab.

5.1. Phase 1

Diese Phase konzentriert sich auf die rein funktionalen Anforderungen, die Applikation wird in dieser Phase als eine "nicht verteilte" Applikation realisiert.

5.2. Phase 2

Diese Phase konzentriert sich primär auf technische Anforderungen wie Verteilung, Performance etc.

⁸ Die UI Komponente, die intern (im Intranet) von den Mitarbeitern des WDA GmbH verwendet wird.

⁹ Je nach Semester und Gruppe können die Semesterwochen leicht von den auf diesem Bild angegebenen Semesterwochen abweichen (z.B. infolge von Feiertagen oder Anzahl Wochen im Semester)

6. Dokumentation

Die während der Entwicklung erstellte Dokumentation muss einen Einblick in das Design und die Architektur der Applikation und den Entwicklungsprozess geben. Dazu wird Ihnen eine Vorlage zur Verfügung gestellt, die das Minimum an Inhalt sicherstellt, aber jederzeit erweitert werden kann. Dazu gehört insbesondere, dass Sie die individuellen, lösungsspezifischen Konzepte ebenfalls dokumentieren. (z.B. Fehlerhandling, Logging, Performance, UI etc.)

7. Abgaben

Am Ende jeder der beiden Phasen muss der aktuelle Stand der Dokumentation als PDF auf ILIAS abgegeben werden. Die Benennung der Datei muss zwingend nach dem folgenden Muster erfolgen: gxx_wda.pdf (Beispiel: g01_wda.pdf, g02_wda.pdf usw.).

Hinweis:

Die erste Abgabe muss bis zum Ende des Unterrichtsblocks erfolgen, in dem die Zwischenabgabe geplant ist. Die zweite Abgabe muss spätestens vor der Präsentation der erarbeiteten Lösung (letzter Unterrichtsblock) erfolgen.

7.1. Phase 1

- Funktionale Anforderungen: Die Abfragen A01 bis A05 sind fehlerfrei implementiert.
- Soweit aktueller Stand der Dokumentation.

Das UI soll minimal gehalten werden.

7.2. Phase 2

- Sämtliche in den funktionalen Anforderungen definierten Abfragen implementiert.
- Alle nicht funktionalen Anforderungen sind erfüllt.
- Finale Version der Dokumentation.

Hinweis:

Der Quellcode der gesamten Applikation muss laufend im pro Gruppe zur Verfügung gestellten Git[Lab]-Projekt verwaltet werden (muss somit nicht explizit abgegeben werden). Es wird ein einfacher CI-Prozess durch GitLab gewährleistet.

8. Präsentation (Vorführung)

Im Rahmen der finalen Abgabe wird die erstellte Applikation vorgeführt. Bei der Vorführung muss gezeigt werden, dass

- die erstellte Applikation als verteilte Applikation betrieben wird,
- die Wetterdaten aus der eigenen Datenbank geholt werden¹⁰ und
- die verlangten Funktionalitäten implementiert wurden.

Bei der Vorführung muss die Ausführung jeweils aus einer Shell gestartet werden, die Nutzung der Entwicklungsumgebung ist nicht erlaubt.

¹⁰ Heisst: Der Zugriff auf den Web-Service ist nur zwecks Aktualisierung vom Datenbestand erlaubt, während in allen anderen Fällen die Wetterdaten aus der eigenen Datenbank geholt werden müssen.

9. Organisatorisches

9.1. Projektaufwand

Sie arbeiten in 4er-Gruppen. Für die Umsetzung des Projekts müssen Sie neben den zur Verfügung gestellten Unterrichtsblöcken (primär für Fragen und Coaching) zusätzliche Zeit im Rahmen des autonomen Selbststudiums einplanen.

9.2. Relevanter Datenbestand

Obwohl die Wetterdaten seit September 2019 gesammelt werden, müssen für die Präsentation nur die Wetterdaten des aktuellen Kalenderjahres in der eigenen Datenbank zur Verfügung stehen¹¹.

¹¹ Natürlich dürfen Sie auch alle Wetterdaten in Ihre Datenbank importieren, jedoch müssen Sie in diesem Fall damit rechnen, dass das Importieren etwas mehr Zeit in Anspruch nehmen kann (abhängig davon, wie effizient Ihr Code bzw. Algorithmus arbeitet).