

ISWE-Projekt: Verbrauchstransparenz

Svaran Chandla (193922), Bernd-Albrecht Heizmann (192915),
Christian Ortlepp (191835), Elisabeth Ostryanin (191674),
Moritz Seppelt (194557), Dennis Untiet (192151)

12. Februar 2023

Inhaltsverzeichnis

1	Einleitung	3
1.1	Zweck	3
1.2	Richtlinien des Dokuments	3
1.3	Zielgruppe und Leseempfehlung	3
1.3.1	Entwickler	3
1.3.2	Kunde	3
1.3.3	Tester	3
1.4	Umfang	4
1.5	Verweise auf sonstige Ressourcen oder Quellen	4
2	Allgemeine Beschreibung	4
2.1	Produktperspektive	4
2.2	Produktfunktionen	4
2.2.1	Eintragen von Verbräuchen	4
2.2.2	Statistiken	5
2.3	Benutzermerkmale	5
2.4	Entwicklungsumgebung	6
2.4.1	Server	6
2.4.2	Client	6
2.5	Design- und Implementierungsbeschränkungen	6
2.6	Benutzerhandbuch/Nutzerdokumentationen	6
2.7	Annahmen und Abhängigkeiten	6
2.7.1	Statistiken	6
2.7.2	Externes Gerät	6
2.7.3	Zahlungsanbieter	6
3	Anforderungen an externe Schnittstellen	7
3.1	Benutzeroberflächen	7
3.1.1	Benutzeroberfläche	7
3.1.2	Supportoberfläche	7

3.1.3	Adminoberfläche	7
3.2	Hardware-Schnittstellen	7
3.3	Software-Schnittstellen	7
3.3.1	Stadtwerke	7
3.3.2	Zahlungsdienstleister	7
3.4	Kommunikationsschnittstellen	7
4	Systemfunktionen	7
4.1	Systemfunktionen für alle Nutzergruppen	8
4.1.1	Anmeldung	8
4.2	Benutzer	8
4.2.1	Accounterstellung	8
4.2.2	Verbräuche eintragen	9
4.2.3	Statistiken einsehen	9
4.2.4	Verbraucher verwalten	10
4.2.5	Prognosen einsehen	10
4.2.6	Kosten einsehen	11
4.2.7	Immobilien bearbeiten	11
4.2.8	Abomodel ändern	12
4.2.9	Account für Support freischalten	13
4.2.10	Account verwalten	14
4.2.11	Verbrauch vergleichen	15
4.2.12	Support Ticket erstellen und interagieren	16
4.3	Support	16
4.3.1	Support-Ticket eines Benutzers annehmen und interagieren	16
4.3.2	Account eines Nutzers verwalten	17
4.4	Admin	17
4.4.1	Abomodelle modifizieren	17
4.4.2	Support- und Adminaccount erstellen	18
4.4.3	Vergleichstags festlegen	19
4.4.4	Vergleichszeitreihe erstellen	19
5	Qualitätsziele und -anforderungen	20
5.1	Benutzerfreundlichkeit für den Benutzer	20
5.2	Verfügbarkeit	20
5.3	Wiederherstellbarkeit	21
5.4	Analysierbarkeit und Modifizierbarkeit	21
5.5	Weitere Anforderungen	21
5.6	Nicht-zu-berücksichtigende Anforderungen	22
6	Andere Anforderungen	22
6.1	Use-Case Diagramm	24
6.2	Klassendiagramm	25

1 Einleitung

1.1 Zweck

Dieses Dokument beschreibt das Softwareprojekt „Verbrauchstransparenz“ in seiner initialen Version. Das Dokument hat den Anspruch, das Projekt in seiner Gesamtheit inklusive aller Subsysteme abzubilden.

1.2 Richtlinien des Dokuments

Schlüssel für funktionale Anforderungen sind stets **fett** gedruckt.

Um die Lesbarkeit zu verbessern, verwenden wir stets das generische Maskulinum, meinen damit jedoch Personen jedes Geschlechts.

1.3 Zielgruppe und Leseempfehlung

1.3.1 Entwickler

Entwickler können dieses Dokument nutzen, um Informationen über die Anforderungen an die von ihnen hergestellte Software zu bekommen.

Grundlegend sind für sie alle Kapitel relevant (je nach Abstraktionsebene), es bietet sich an diese in Reihenfolge zu lesen, um von generellen Aspekten zu spezifischen Details überzugehen. Für die Implementierung ist Kapitel 4 von besonderem Interesse, da es die einzelnen Funktionalitäten detailliert beschreibt.

1.3.2 Kunde

Der Kunde kann aus diesem Dokument den aktuellen Stand der Projektplanung verfolgen, außerdem dient es ihm um zu überprüfen, ob seine Anforderungen korrekt verstanden wurden und die Prioritäten des Entwicklungsteams mit seinen eigenen übereinstimmen.

Auch für den Kunden sind grundsätzlich alle Sektionen relevant, jedoch sollte er besonderes Augenmerk auf die Kapitel mit hohem Abstraktionsgrad legen, wie zum Beispiel die allgemeine Beschreibung und die System- und Qualitätsanforderungen.

1.3.3 Tester

Tester können dieses Dokument nutzen, um die konkreten Anforderungen an das Produkt und somit auch das zu testende Verhalten besser zu verstehen.

Für sie ist Kapitel 4 und 5 von besonderer Bedeutung, um einen Gesamtüberblick zu bekommen kann es aber auch sinnvoll sein die anderen Teile des Dokuments zu lesen.

1.4 Umfang

Die Software gibt Benutzern die Möglichkeiten ihre Verbräuche, primär Strom, Gas und Wasser, aber auch individuelle Verbrauchswerte im Blick zu haben und diese mit Anderen zu vergleichen.

Dabei stehen dem Benutzer 3 verschiedene Abomodelle (FREE, STANDARD, PROFESSIONAL), unter denen er wählen kann, zur Verfügung. Diese bestimmen die Möglichkeiten der Datenerhebung, den Grad der Auswertung dieser Daten und deren Speichermöglichkeiten.

Für die Nutzer entsteht ein Mehrwert zum einen durch das Erfüllen seines Informationsinteresses am eigenen Verbrauch und zum anderen durch das Erkennen von Einsparpotentialen im Haushalt.

Darüber hinaus soll die Nutzung des Produkts auch zum Energiesparen im Sinne des Umweltschutzes beitragen.

1.5 Verweise auf sonstige Ressourcen oder Quellen

Dieses SRS-Dokument erhebt den Anspruch der Vollständigkeit. Aus diesem Grund sind keine weiteren Quellen oder sonstige zusätzliche Ressourcen vonnöten.

2 Allgemeine Beschreibung

2.1 Produktperspektive

Beim entwickelten Produkt handelt es sich um eine komplette Neuentwicklung. Architekturell handelt es sich um eine Client-Server-Anwendung mit einer Browserapplikation, welche mit einem Server kommuniziert, der wiederum alle benötigten Daten aufbereitet und persistiert.

2.2 Produktfunktionen

2.2.1 Eintragen von Verbräuchen

Der primäre Anwendungsfall der Applikation ist das Eintragen und Verwalten von verschiedenartigen Verbrauchswerten. Hierzu zählen unter anderem, aber nicht ausschließlich, Gas, Strom und Wasser.

Manuelles Eintragen Verbrauchswerte können über das Webinterface eingetragen werden.

Verbraucher Verbraucher, bei denen der Verbrauch konstant ist, können ebenfalls über das Webinterface eingetragen werden.

Automatisiertes Eintragen Verbrauchswerte können zudem über Externe Geräte von Drittanbietern direkt am Zähler ausgelesen werden. Die Werte werden dann vom Gerät über eine von der Applikation bereitzustellende API übertragen.

2.2.2 Statistiken

Alle gesammelten Daten können im Webinterface auf verschiedene Arten visualisiert und mit anderen Daten verglichen werden. Der Vergleich soll sowohl anhand des eigentlichen Verbrauchs (also in beispielsweise kWh) als auch umgerechnet in Kosten in der jeweiligen Landeswährung oder Umweltbelastung in beispielsweise Tonnen CO_2 möglich sein.

Vergleich mit eigener Historie Der Benutzer kann seinen aktuellen Verbrauch mit (von ihm ausgewählten) Vergleichszeitreihen aus seiner eigenen Verbrauchshistorie vergleichen.

Vergleich mit anderen Nutzern der Plattform Darüber hinaus kann der Nutzer seinen Momentanverbrauch auch mit gemittelten Werten aus Vergleichszeitreihen vergleichen, die aus diversen Nutzergruppen mit bestimmten Vergleichstags generiert werden. Die Vergleichstags sind variabel und müssen durch den Systemadministrator einstellbar sein. Vorstellbare Vergleichstags sind beispielsweise eine Aufteilung der Nutzer nach Wohnsitz oder Größe der Immobilie.

Vergleich mit plattformunabhängigen Metriken Die letzte mögliche Vergleichsmetrik besteht in Standard Vergleichswerten, welche nicht von Nutzern dieser Applikation, sondern von externen Forschungsinstituten bereitgestellt werden. Diese Daten werden aus dem Format des externen Instituts zur internen Verwendung in eine Vergleichszeitreihe übertragen, um unabhängig von der Auslieferungsart der Daten des Instituts zu sein.

2.3 Benutzermerkmale

Der Nutzerkreis wird auf Privatpersonen eingeschränkt. Zur Nutzung der App muss eines von drei Abomodellen (FREE, STANDARD, PROFESSIONAL) ausgewählt werden. Die Aufteilung folgt einem „Freemium“ Schema, sprich es gibt eine kostenlose Klasse und zwei Bezahlmodelle. Die Abgrenzung zwischen den Klassen bleibt zu entscheiden. Vorstellbar ist hier eine Begrenzung der Anzahl von Immobilien im Free-Modell, oder etwa, dass Features, wie das automatisierte Eintragen von Verbrauchswerten (siehe 2.2.1), nur für bezahlende Nutzer zur Verfügung stehen.

Neben dem Benutzer soll es auch weitere Benutzertypen geben - dem Support- und Admin-Benutzer. Diese können zur Bedienung des Produkts geschult werden.

2.4 Entwicklungsumgebung

2.4.1 Server

Für die Laufzeitumgebung des Servers gibt es keine Einschränkungen.

2.4.2 Client

Die Client-Anwendung wird als Browseranwendung zur Verfügung gestellt. Sie muss alle gängigen Browser (Google-Chrome und Chromium basierte Browser, Firefox, Safari) auf allen gängigen Endgeräten (Smartphone, Tablet, PC) mit den jeweiligen Betriebssystemen (Android / iOS, Windows / MacOS / Linux) unterstützen.

2.5 Design- und Implementierungsbeschränkungen

Wie bereits in 2.4.2 beschrieben, ist die Client-Anwendung als Browserapplikation zu implementieren. Darüber hinaus muss die Nutzeranwendung einfach und intuitiv bedienbar sein da kein Fachpublikum angesprochen wird.

2.6 Benutzerhandbuch/Nutzerdokumentationen

Da die Client-Applikation intuitiv bedienbar sein muss, sollte eine Dokumentation für den Großteil der Nutzer nicht notwendig sein. Da jedoch auch technisch weniger versierte Nutzer zur Zielgruppe gehören und um spezielle Anwendungsfälle abzudecken wird eine Dokumentation in Form eines Online-Wikis direkt auf der Website der Anwendung bereitgestellt. Darüber hinaus können Fragen auch vom Kundensupport beantwortet werden.

2.7 Annahmen und Abhängigkeiten

2.7.1 Statistiken

Plattformunabhängige Statistiken (siehe 2.2.2) werden von Drittanbietern bereitgestellt. Das Format, die Aktualität der Daten und die Kontinuität der Bereitstellung kann nicht sichergestellt werden.

2.7.2 Externes Gerät

Die Implementierung von kompatiblen externen Geräten (siehe 2.2.1) ist abhängig von Drittanbietern.

2.7.3 Zahlungsanbieter

Um Nutzern das Bezahlen so komfortabel wie möglich zu machen, greifen wir auf externe Zahlungsdienstleister, wie beispielsweise PayPal, Klarna, Visa usw., zurück. Die Schnittstellen dieser Anbieter können sich ändern, sowie deren Geschäftsbedingungen, Kosten und Anforderungen.

3 Anforderungen an externe Schnittstellen

3.1 Benutzeroberflächen

3.1.1 Benutzeroberfläche

Die Software bietet ein User-Interface an, welches Zugriff auf alle Standard Funktionen, wie Verbrauch eintragen, Verbrauch vergleichen oder Statistiken anzeigen, bereitstellt.

3.1.2 Supportoberfläche

Über ein Support-Interface soll erweiterter Zugriff auf User-Accounts nach deren Freigabe möglich sein.

3.1.3 Adminoberfläche

Es wird eine Admin-Oberfläche bereitgestellt, welche vollkommene Kontrolle über alle Funktionalitäten hat. Insbesondere soll es möglich sein, Vergleichszeitreihen anzulegen und die Funktionalitäten, die den Abo-Modellen zugeordnet sind, anzupassen.

3.2 Hardware-Schnittstellen

Die Software bedient sich keiner Hardware-Interfaces.

3.3 Software-Schnittstellen

3.3.1 Stadtwerke

Die Software bezieht, sofern vorhanden, Daten aus den örtlichen Stadtwerken um Statistiken bereitzustellen, mit denen der eigene Verbrauch verglichen werden kann.

3.3.2 Zahlungsdienstleister

Ein Austausch mit den Zahlungsdienstleistern ist erforderlich, um Zahlungen für kostenpflichtige Abomodelle gewährleisten zu können.

3.4 Kommunikationsschnittstellen

Die Kommunikation der einzelnen Interfaces erfolgt über HTTPS.

4 Systemfunktionen

Wir haben die Systemfunktionen nach Nutzergruppen aufgeteilt.

4.1 Systemfunktionen für alle Nutzergruppen

4.1.1 Anmeldung

Beschreibung und Priorität Bevor ein dem System bereits bekannter Nutzer die Funktionen seiner Nutzergruppe ausführen kann, muss er sich zunächst authentifizieren. Die Funktion hat eine sehr hohe Priorität, da sonst Nutzer und Nutzergruppen nicht voneinander unterschieden werden können.

Sequenzen von Benutzeraktionen und Systemantworten Der Nutzer gibt in einer Anmelde-Maske seinen Benutzernamen oder seine E-Mail-Adresse und sein Passwort ein und bestätigt dies. Ist ANM4 erfüllt so ist der Nutzer nun eingeloggt und wird zum Home-Bildschirm weitergeleitet. War dies nicht der Fall, so wird eine Fehlermeldung mit den möglichen Ursachen angezeigt.

Funktionale Anforderungen

- **ANM1** Darstellen der Anmelde-Maske.
- **ANM2** Validieren der Eingabe also Überprüfung von ANM4 durch ggf. Abgleich mit der Datenbank.
- **ANM3** Auf nicht erfüllte Anforderungen (ANM4) in der Anmelde-Maske aufmerksam machen, falls diese nicht erfüllt sind.
- **ANM4** Der Benutzername oder die E-Mail-Adresse muss im System vorhanden sein und das Passwort bzw. dessen Hash müssen übereinstimmen.

4.2 Benutzer

4.2.1 Accounterstellung

Beschreibung und Priorität Diese Funktion ermöglicht es einer Person, die an dem Produkt interessiert ist, einen Account anzulegen und anzufangen das Produkt zu nutzen. Die Priorität dieser Funktion ist sehr hoch, da ohne sie das Produkt keine neuen Nutzer gewinnen kann.

Sequenzen von Benutzeraktionen und Systemantworten Der Interessierte gibt in der Anmelde-Maske seinen frei gewählten Benutzernamen, seine E-Mail-Adresse und schließlich ein Passwort ein. Nach einem Knopfdruck wird, wenn die Eingaben die funktionalen Anforderungen REG4, REG5, REG6 erfüllen, der Benutzer nun auf die Startseite weitergeleitet. Wenn nicht, wird der Benutzer über die nicht erfüllten Anforderungen in der Anmelde-Maske informiert und kann daraufhin seine Eingaben anpassen.

Funktionale Anforderungen

- **REG1** Darstellen der Anmelde-Maske
- **REG2** Validieren der Eingabe also Überprüfung von REG4, REG5, REG6 durch ggf. Abgleich mit der Datenbank
- **REG3** Auf nicht erfüllte Anforderungen (REG4, REG5, REG6) in der Anmelde-Maske aufmerksam machen, falls diese nicht erfüllt sind.
- **REG4** Keine 2 Benutzer dürfen denselben Benutzernamen haben.
- **REG5** Das Passwort muss dem folgenden Standard genügen:
 - Mindestlänge: 8 Zeichen
 - Enthält min. einen deutschen Großbuchstaben (A-Z, Ä, Ö, Ü)
 - Enthält min. einen deutschen Kleinbuchstaben (a-z, ä, ö, ü, ß)
 - Enthält min. eine arabische Ziffer (0-9)
 - Enthält min. ein Sonderzeichen (!, \$, #, %, ...)
- **REG6** Die eingegebene E-Mail-Adresse muss im richtigen Format sein.
- **REG7** Abspeichern der neuen Accountdaten.

4.2.2 Verbräuche eintragen

Beschreibung und Priorität Die Funktion ermöglicht dem Benutzer seine Verbrauchsdaten in die Datenbank einzutragen, welche später für weitere Features vonnöten sind. Neben dem manuellen Eintragen gibt es auch die Möglichkeit der automatischen Erfassung der Daten mittels eines externen Geräts, welches der Benutzer bei sich Zuhause installiert. Sobald dies passiert ist, übermittelt das Gerät jeden Tag die aktuellen Daten.

Sequenzen von Benutzeraktionen und Systemantworten Beim manuellen Eintragen trägt der Benutzer in einer Webmaske seine einzelnen Verbräuche ein und kann mit einem Klick auf den Button „hochladen“ seine Daten der Datenbank hinzufügen. Bevor dies geschieht, wird aber überprüft, ob die Daten neu und sinnig sind.

Die konkreten Anforderungen an die Schnittstelle für das externe Gerät zum automatischen Eintragen von Verbräuchen sind noch zu diskutieren.

Funktionale Anforderungen Noch zu diskutieren.

4.2.3 Statistiken einsehen

Beschreibung und Priorität Die Funktion ermöglicht es dem Benutzer seine eingetragenen Verbräuche (siehe 4.2.2) zu visualisieren. Da diese Funktion den größten Nutzen für die Benutzer bringt, hat auch diese Funktion eine hohe Priorität, sie ist aber nicht essenziell für das Funktionieren des Systems.

Sequenzen von Benutzeraktionen und Systemantworten Der Nutzer geht auf den Menüpunkt Statistiken und wählt den gewünschten Verbrauchszeitraum und die Verbrauchsart aus. Außerdem kann er sich für eine Art der Visualisierung (z.B. Säulendiagramm, Liniendiagramm, oder Tabelle) entscheiden. Die gewünschten Daten werden daraufhin dargestellt.

Funktionale Anforderungen Noch zu diskutieren.

4.2.4 Verbraucher verwalten

Beschreibung und Priorität Es soll möglich sein, dass neben dem manuellen Eintragen der Verbräuche und automatischen Eintragen durch ein externes Gerät (siehe 4.2.2), die Verbräuche durch einen Verbraucher zu generieren. Das Verwalten von Verbraucher hat eine eher niedrige Priorität, da die Kernfunktionalität der Verbräuche schon mit manuellen Eintragen abgedeckt ist.

Sequenzen von Benutzeraktionen und Systemantworten Wenn der Nutzer einen bestehenden Verbrauch bearbeiten will, soll dieser aus einer Liste aller erstellten Verbräucher auswählbar sein. Dabei ist jeder Verbraucher einer Immobilie zugeordnet, was in der Liste auch kommuniziert sein soll. Es kann natürlich auch ein Verbraucher aus der Liste gelöscht werden.

Falls ein Verbraucher bearbeitet oder erstellt werden soll, kann man nun den Namen, den täglichen Verbrauch (in Bezug auf eine Verbrauchszeitreihe) und die Immobilie eintragen oder bearbeiten. Schließlich können die Änderungen gespeichert werden.

Funktionale Anforderungen

- **VERV1** Liste aller Verbraucher anzeigen. Zusätzlich signalisieren, zu welcher Immobilie der jeweilige Verbraucher assoziiert ist.
- **VERV2** Name, assoziierte Immobilie und täglichen Verbrauch anpassen.
- **VERV3** Tägliches Erstellen eines Verbrauchs mit dem gespeicherten täglichen Verbrauch.

4.2.5 Prognosen einsehen

Beschreibung und Priorität Die Funktion Prognosen einsehen dient dem Nutzer dazu, Prognosen für zukünftige Verbräuche einzusehen. Die Prognosen werden auf Basis von vorherigen Verbräuchen und eingetragenen Verbrauchern gebildet. Wenn der Verbrauch eines Nutzers im letzten Jahr ungefähr gleich geblieben ist, wird dies auch für das nächste Jahr prognostiziert.

Sequenzen von Benutzeraktionen und Systemantworten Wenn der Nutzer sich Prognosen für seine Verbräuche anzeigen lassen möchte, muss er zunächst einen vorhandenen Verbrauchstyp auswählen. Zusätzlich dazu muss der Nutzer einen bestimmten Zeitraum wählen, für die er die Prognose möchte. Falls dieser Zeitraum zu weit in der Zukunft liegt, oder der Verbrauchstyp kaum oder gar keine Verbräuche besitzt, können keine Prognosen ermittelt werden. Wenn Prognosen ermittelt werden können, werden diese dem Nutzer angezeigt. Ob diese Prognosen in Form eines Gesamtverbrauchs (z.B. 1600 kWh für nächstes Jahr), oder in Form einer Verbrauchszeitreihe (Diagramm) angezeigt werden, ist noch offen.

Funktionale Anforderungen Noch zu diskutieren

4.2.6 Kosten einsehen

Beschreibung und Priorität Die Funktion Kosten einsehen ermöglicht dem Nutzer die Kosten zu seinen Verbräuchen einzusehen. Bevor dies geschehen kann, muss der Benutzer einen Preis pro angegebener Einheit angeben. Zudem kann er sich bei eigenen Verbrauchstypen auch eigene Kosten eintragen. Beispiel: Der Nutzer hat den Verbrauchstyp Anzahl an Kaffee pro Tag. Dazu kann der Nutzer nun einen Preis eintragen: Ein Kaffee kostet einen Euro.

Sequenzen von Benutzeraktionen und Systemantworten Der Nutzer muss einen Verbrauchstyp auswählen und dann zu diesem Typ einen Zeitraum und Kosten pro Einheit wählen. Danach werden ihm die Kosten für den Verbrauch über den Zeitraum angezeigt. Wenn der Nutzer seinen Verbrauch vergleicht, ist neben dem Vergleich des Verbrauchs auch ein Vergleich der Kosten sichtbar.

Funktionale Anforderungen Noch zu diskutieren

4.2.7 Immobilien bearbeiten

Beschreibung und Priorität Der Benutzer kann Immobilien hinzufügen, bearbeiten und entfernen. Diese Funktion hat eine mittlere Priorität da Verbräuche an eine Immobilie gebunden sind.

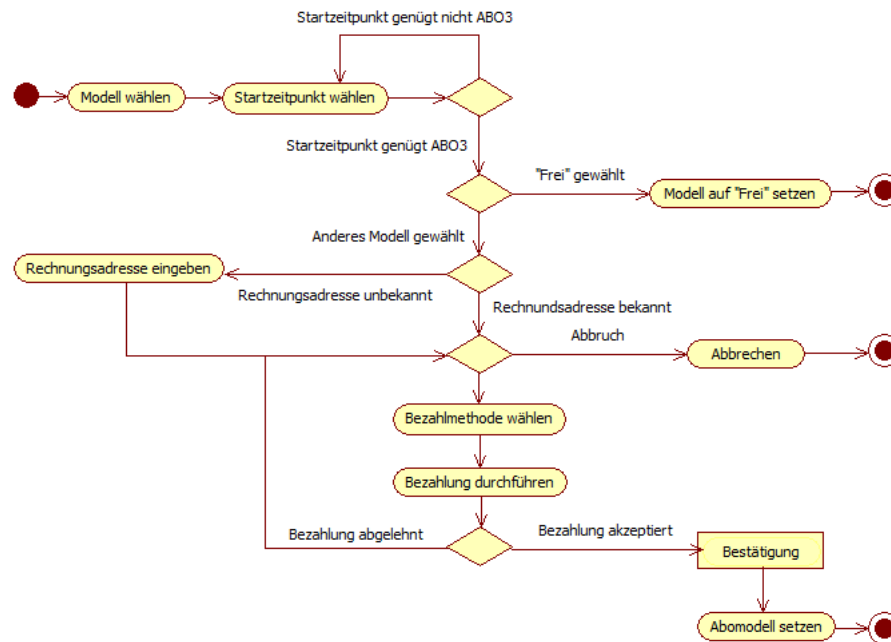
Sequenzen von Benutzeraktionen und Systemantworten Der Benutzer wählt in seinem Profil aus, dass er seine Immobilien bearbeiten möchte. Dabei bekommt er die Option, eine neue Immobilie anzulegen, oder von vorhandenen Immobilien eine zu bearbeiten. Wenn er das Erste aussucht, kann man eine Bezeichnung, Flächeninhalt und Haushaltsgröße eintragen. Die Bezeichnung ist zwingend notwendig, die anderen Angaben nicht. Der Nutzer kann sich dazu entscheiden, den Vorgang abzubrechen. Wenn der Nutzer eine vorhandene Immobilie bearbeiten will, dann werden ihm ebenfalls angeboten, Bezeichnung, Flächeninhalt und Haushaltsgröße zu ändern. Zusätzlich hat er die Option, die

ganze Immobilie zu löschen. Wenn der Nutzer etwas bearbeitet hat, muss er seine Änderungen bestätigen. Sobald er das getan hat, ist der Bearbeitungsvorgang abgeschlossen.

Wenn der Nutzer sich für das Löschen entscheidet, wird er final gewarnt, dass das Löschen einer Immobilie alle damit verbundenen Daten löscht und der Vorgang nicht widerrufbar ist. Wenn der Nutzer zugestimmt hat, dass ihm das bewusst ist und trotzdem fortfahren will, dann wird die Immobilie und alle Daten, die damit verbunden waren, aus der Datenbank entfernt.

Funktionale Anforderungen Noch zu diskutieren.

4.2.8 Abomodell ändern



Beschreibung und Priorität Der Benutzer kann hiermit zwischen Abomodellen wechseln. Geplant sind die Abomodelle FREE, STANDARD und PROFESSIONAL, wovon Ersteres kostenlos ist und standardmäßig ausgewählt sein sollte. Diese Funktion enthält neben dem Auswählen des Modells die Zahlung und Überprüfung vertraglicher Kündigungsfristen. Die Funktion gehört nicht zu den Kernanforderungen jedoch kann ohne sie kein Umsatz erzielt werden und hat daher trotzdem eine hohe Priorität.

Sequenzen von Benutzeraktionen und Systemantworten Nach der Auswahl des gewünschten Abomodells muss man, falls sich der Nutzer für ein kos-

tenpflichtiges Modell entscheidet, einen Startzeitpunkt (nach Richtlinie ABO3) entscheiden. Nach der Eingabe von Rechnungsadresse, die, falls sie für den Benutzer schon bekannt ist, auch schon voreingetragen sein sollte, kann der Benutzer eine präferierte Zahlungsmethode anwählen, wodurch er zu einem externen Zahlungsdienstleister weitergeleitet wird. Schließlich kann der Prozess abgeschlossen werden und das neue Abomodell wird zum angegebenen Startzeitpunkt gesetzt. Eine Zahlungsbestätigung wird versendet. Der Vorgang kann zu jedem Zeitpunkt abgebrochen werden.

Funktionale Anforderungen

- **ABO1** Darstellen der Maske zum Auswählen des Abomodells und Eintragen von Startzeitpunkt und Rechnungsadresse.
- **ABO2** Validieren der Eingabe des Startzeitpunkts (nach ABO3) und Reflektieren möglicher Fehler in der Maske.
- **ABO3** Der Startzeitpunkt muss folgenden Richtlinien genügen:
 - Er muss in der Zukunft liegen oder am derzeitigen Tag sein
 - Falls der Kündigungszeitraum eines laufenden Abos noch nicht abgelaufen ist, muss der Startzeitpunkt nach Ablauf dieses Zeitraums liegen
- **ABO4** Grundlegende Format-Überprüfungen der Rechnungsadresse (z.B. Postleitzahl besteht aus Zahlen etc.). Nichteinhaltung sollen in der Maske kommuniziert werden.
- **ABO5** Eine Verbindung zu den externen Zahlungsanbietern muss aufgebaut werden.
- **ABO6** Schaffen einer Möglichkeit in jedem Schritt den Prozess abzubrechen.
- **ABO7** Versenden einer Zahlungsbestätigung.
- **ABO8** Setzen des neuen Abomodells in den Benutzerdaten zum angegebenen Startzeitpunkt.

4.2.9 Account für Support freischalten

Beschreibung und Priorität Diese Funktion ermöglicht es dem Benutzer bei Problemen seinen Account für den Support freizuschalten, sodass der Support Änderungen vornehmen kann. Dies ermöglicht einen Support, der zugleich datenschutz- und nutzerfreundlich ist.

Sequenzen von Benutzeraktionen und Systemantworten Der Benutzer hat mit der Nutzung des Systems ein Problem und benötigt Hilfe. Nachdem er die FAQ zur Handhabung des Systems gelesen hat und er dort keine Lösung für sein Problem finden konnte, nutzt er die Supportfunktion, wo er mit einem Mitarbeiter in Kontakt kommt. Zuerst versucht der Support dem Nutzer ohne weitere Daten weiterzuhelfen. Sollte dies das Problem immer noch nicht lösen, muss der Nutzer seinen Account für den Support freischalten. Dem Nutzer wird ein Code generiert, den er dem Support durchgibt und den der Support benötigt, um sich berechtigten Zugang zum Nutzeraccount zu verschaffen. Nun kann der Support dem Nutzer optimal weiterhelfen, indem er Zugriff auf seinen Account hat.

Funktionale Anforderungen

- **REG1** Laufende Internet- und/oder Telefonverbindung.
- **REG2** Der generierte Code für den Nutzer besteht aus Zahlen und Buchstaben.
- **REG3** Der generierte Code von Nutzer muss unbedingt mit dem benötigten Code des Supports übereinstimmen.

4.2.10 Account verwalten

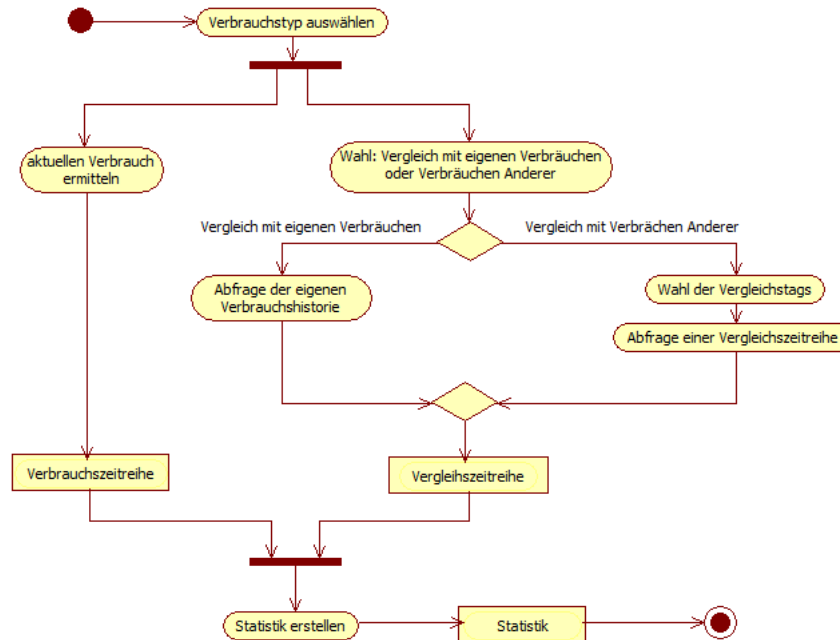
Beschreibung und Priorität Das Feature erlaubt es dem Nutzer seine Accountdaten einzusehen, zu ändern und gegebenenfalls zu löschen und es hat eine sehr hohe Priorität.

Sequenzen von Benutzeraktionen und Systemantworten Der Nutzer wählt den Menüpunkt „Account Verwalten“ aus. Dort sind alle Optionen AVW1 - AVW6 einsehbar und änderbar.

Funktionale Anforderungen

- **AVW1** Name.
- **AVW2** Adresse.
- **AVW3** Zahlungsmethode.
- **AVW4** Nutzernamen.
- **AVW5** Email.
- **AVW6** Passwort. Hier sollte das aktuelle Passwort jedoch nicht einsehbar sein (da es nicht gespeichert werden soll).

4.2.11 Verbrauch vergleichen



Beschreibung und Priorität Die Funktion Verbrauch vergleichen ermöglicht dem Nutzer seinen Verbrauch mit seiner eigenen Verbrauchshistorie, oder mit den Verbräuchen anderer zu vergleichen. Falls man sich mit den Verbräuchen von anderen vergleichen möchte, kann man Vergleichstags auswählen (siehe 4.4.3). Dann wird eine Vergleichszeitreihe neben der eigenen Verbrauchszeitreihe visualisiert.

Sequenzen von Benutzeraktionen und Systemantworten Zunächst muss der Nutzer einen Verbrauchstyp auswählen. Danach wählt der Nutzer aus, ob er sich mit seinen eigenen Verbräuchen, oder mit den Verbräuchen anderer vergleichen möchte. Wenn er ersteres auswählt, sucht sich der Nutzer zwei Zeiträume aus, die er gegeneinander vergleichen möchte. Daraus werden zwei Zeitreihen generiert, die dann in einer Statistik visualisiert werden. Möchte der Nutzer sich mit Anderen vergleichen, wählt er einen oder mehrere Vergleichstags (siehe 4.4.3) aus. Beispiel: Der Nutzer möchte sich mit anderen Haushalten aus seiner Region vergleichen. Der Nutzer kann den Namen seiner Region und die Wohnfläche des Hauses angeben. Auch hier muss der Nutzer einen Zeitraum wählen, in dem der Nutzer sich vergleichen will. Dann wird eine Vergleichszeitreihe mit der Vergleichsgruppe in dem gegebenen Zeitraum erstellt und visualisiert.

Funktionale Anforderungen Noch zu diskutieren

4.2.12 Support Ticket erstellen und interagieren

Beschreibung und Priorität Dem Benutzer wird die Möglichkeit gegeben, Support Tickets zu erstellen und somit Kontakt zu einem Supporter aufzunehmen der ihm gegebenenfalls bei einem Problem bezüglich der Software helfen kann. Wenn ein Problem gelöst wurde, kann das Support-Ticket auch geschlossen werden. Diese Funktion ist relevant um die Benutzerfreundlichkeit weiter zu erhöhen.

Sequenzen von Benutzeraktionen und Systemantworten Der Benutzer wählt den Knopf „Support kontaktieren“ um mit dem Support in Kontakt zu treten. Hier hat er schon die Möglichkeit, sein Problem konkreter zu schildern. Nach anschließender Bestätigung durch einen weiteren Knopfdruck wird das Ticket erstellt und der Benutzer erhält eine Benachrichtigung, dass sich zeitnah jemand um seine Angelegenheit kümmern wird. Sobald ein Supporter Zeit hat, sich mit dem Problem zu beschäftigen, kann der Supporter das Ticket annehmen (siehe 4.3.1) und ein Live-Chat zwischen beiden wird eingerichtet. Wenn sich das Problem erledigt hat, bekommt der Support die Möglichkeit den Live-Chat zu schließen. Es soll auch einen Knopf geben, um das Support-Ticket zu schließen.

Funktionale Anforderungen

- **TICK1** Bereitstellung eines Interfaces um das Ticket zu erstellen.
- **TICK2** Möglichkeit für den Benutzer den Live-Chat zu schließen.

4.3 Support

4.3.1 Support-Ticket eines Benutzers annehmen und interagieren

Beschreibung und Priorität Es ist wichtig, dass der Support nicht unkontrolliert auf alle Accounts der Nutzer zugreifen kann. Daher müssen vor einer Interaktion mit den Daten der Nutzer immer ein Ticket vom Benutzer erstellt werden (siehe 4.2.12). Nachdem das gelungen ist und das Ticket vom Support angenommen wurde, hat dieser die Berechtigung den Account zu verwalten (siehe 4.3.2) und dem Benutzer optimal zu helfen.

Sequenzen von Benutzeraktionen und Systemantworten Nachdem der Benutzer ein Support-Ticket geöffnet hat (siehe 4.2.12), muss der Support nur noch das Ticket annehmen, durch das Klicken auf einen Button. Danach öffnet sich ein Live-Chat zwischen beiden und der Support hat Zugriff auf die Daten des Benutzers. Im Fall, dass es nützlich sein sollte, kann der Supporter das Bearbeiten des Benutzer-Accounts anfragen (siehe 4.3.2). Nachdem das Problem gelöst wurde, kann der Supporter den Live-Chat und somit auch das Ticket

schließen, durch das Klicken auf einen weiteren Button und der Support verliert den Zugriff auf die Daten des Nutzers.

Funktionale Anforderungen

- **SUPA1** Möglichkeit für den Supporter den Live-Chat zu schließen.
- **SUPA2** Möglichkeit für den Supporter das Support Ticket zu schließen.

4.3.2 Account eines Nutzers verwalten

Beschreibung und Priorität Ein Support muss in der Lage sein, etwaige Benutzerdaten auf Anfrage des jeweiligen Benutzers zu ändern. Dafür muss allerdings das Einsehen der Benutzerdaten für den Support freigeschaltet werden (siehe 4.2.9). Diese Funktion ist nicht Teil der Kernanforderungen, jedoch muss es unbedingt Teil der ersten Version sein, da die Kundenzufriedenheit davon abhängig ist. Daher ergibt sich eine mittlere Priorität.

Sequenzen von Benutzeraktionen und Systemantworten Der Support kann den Benutzer auswählen, dessen Stammdaten bearbeitet werden sollen. Dabei werden dem Support nur diejenigen Benutzerdaten angezeigt, die die Bearbeitung freigeschaltet haben. Danach wird eine Oberfläche angezeigt, die der Oberfläche zum Bearbeiten der eigenen Stammdaten eines Benutzers ähnelt. Hier können nun die Änderungen vorgenommen und gespeichert werden.

Funktionale Anforderungen

- **SMG1** Zeige alle Benutzer, die die Bearbeitung nach 4.2.9 freigeschaltet haben.
- **SMG2** Bearbeiten aller Stammdaten eines Benutzers in der Oberfläche ermöglichen.
- **SMG3** Speichern der veränderten Daten.

4.4 Admin

4.4.1 Abomodelle modifizieren

Beschreibung und Priorität Der Admin kann hiermit alle drei verfügbaren Abomodelle (FREE, STANDARD und PROFESSIONAL) grundlegend ändern. Alle Abomodelle haben verschiedene Funktionen, welche der Admin anpassen kann. Der Admin hat die Möglichkeit, die Kosten der verschiedenen Modelle anzupassen, Funktionen freischalten und Funktionen einzuschränken oder sogar ganz zu entfernen. Falls der Admin die Kosten eines Modells ändert, oder Funktionen hinzufügt oder entfernt, müssen gesetzliche Regelungen für Vertragsänderungen eingehalten werden. Das heißt Nutzer dieses Abomodells müssen benachrichtigt werden und es muss Ihnen eine Möglichkeit gegeben werden, das Abomodell zu

kündigen. Das passiert unabhängig davon, ob der Admin die Kosten erhöht oder verringert, oder Funktionen hinzufügt oder entfernt.

Sequenzen von Aktionen Admin und Systemantworten Falls der Admin Abomodelle modifizieren will, muss der Admin gemäß ABOM3 einen Startzeitpunkt wählen, ab dem die Änderungen gültig werden. Dem Admin soll bekannt gemacht werden, ob der Zeitpunkt zu kurzfristig gewählt ist. Danach kann der Admin aus folgenden Funktionen wählen:

- Vorhandene Funktion bearbeiten oder entfernen (Funktion muss in dem Abomodelle momentan verfügbar sein).
- Neue Funktion hinzufügen.
- Den Preis des Abomodells ändern.

Wenn der Admin sich für eine der Funktionen entschieden hat, kann er diese ausführen und bestätigen, sofern alles den Anforderungen entspricht. Der gesamte Vorgang kann jederzeit auch abgebrochen werden. Fährt der Admin fort, speichert das System die Änderungen und wendet diese zu dem gegebenen Zeitpunkt an.

Funktionale Anforderungen

- **ABOM1** Darstellen der Maske zum Auswählen des Abomodells und Eintragen von Startzeitpunkt.
- **ABOM2** Funktion entsprechend dem gewählten Abomodelle muss änderbar und löschar sein.
- **ABOM3** Funktion soll hinzufügar sein.
- **ABOM4** Ein neuer Preis für das Abomodelle soll wählbar sein.
- **ABOM5** Der Startzeitpunkt muss in der Zukunft liegen und muss den gesetzlichen Regelungen entsprechen.
- **ABOM6** Validieren der Eingabe des Startzeitpunkts (nach ABO5) und Reflektieren möglicher Fehler in der Maske.

4.4.2 Support- und Adminaccount erstellen

Beschreibung und Priorität Ein Admin soll in der Lage sein Support- und Adminaccounts zu erstellen. Der Supportaccount kann von dem Support-Team verwendet werden um Zugang zu dem System zu erhalten. Über diesen Account kann der Support die Systemfunktionen des Supports nutzen. Ein Adminaccount hätte dieselben Rechte, wie der erstellende Adminbenutzer.

Sequenzen von Adminaktionen und Systemantworten Um einen Support- oder Adminaccount zu erstellen soll eine Maske erstellt werden. In der Maske gibt es die Felder: Name, Passwort und E-Mail sowie die Wahl, ob es sich um ein Support- oder Adminaccount handelt. Wenn der Admin alle Felder ausgefüllt hat, kann er auf der Maske einen Bestätigungsknopf drücken und der Account wird dadurch angelegt. Zusätzlich soll es auf der Maske einen „Schließen“ Knopf geben, falls der Admin die Aktion abbrechen will.

Funktionale Anforderungen Noch zu diskutieren.

4.4.3 Vergleichstags festlegen

Beschreibung und Priorität Der Admin soll in der Lage sein, Vergleichstags festzulegen. Beispiele dafür wären regionale Vergleiche, Vergleiche in Haushaltsgröße, Vergleiche mit ähnlicher Immobiliengröße, Vergleiche mit Immobilienalter und Vergleiche mit dem Alter des Nutzers. Später können dann Vergleichszeitreihen (siehe 4.4.4) mit dem neuen Tag erstellt werden.

Sequenzen von Adminaktionen und Systemantworten Der Admin wählt in einer Maske eine neuen Vergleichstag, den er hinzufügen möchte. Es muss gewährleistet sein, dass nicht schon ein Vergleichstag unter demselben Namen existiert. Die Maske kann der Admin schlussendlich bestätigen oder abbrechen. Bei Bestätigung wird der Vergleichstag erstellt.

Funktionale Anforderungen Noch zu diskutieren.

4.4.4 Vergleichszeitreihe erstellen

Beschreibung und Priorität Der Admin kann Vergleichszeitreihen anlegen. Dafür lässt er sich die Daten aller Nutzer zu interessanten und relevanten Kennzahlen berechnen, welche bei ihm gespeichert werden, damit er, wenn ein Nutzer sich vergleichen möchte mit anderen Nutzern, nur noch die Daten vom Admin geladen werden, statt jedes Mal auf die gesamte Datenbank zuzugreifen. Alternativ kann der Admin auch aus externen Ressourcen Vergleichswerte eintragen.

Sequenzen von Adminaktionen und Systemantworten Der Admin klickt im Menü auf die Schaltfläche „Vergleichszeitreihe erstellen“. Dann sieht man auf der Webseite die Möglichkeit „aus eigenen Nutzerdaten erstellen“ oder „manuell eintragen“. Wenn man das Erste auswählt, werden die Daten der Nutzer ausgewertet und mit den zugehörigen Tags verknüpft. Das kann einige Sekunden dauern also gibt es während der Ladezeit eine Ladegrafik, welche dem Admin mitteilt, dass die Seite mit Rechnen beschäftigt ist. Der Admin kann dann in allen Tags die berechneten Werte einsehen. Wenn er auf „Vergleichszeitreihe aktualisieren“ klickt, wird er danach gefragt, ob er sich sicher ist, dass er die Zeitreihe ändern möchte. Wenn der Admin die Anfrage bestätigt, werden diese

Werte als neue Vergleichswerte definiert. Das heißt, wenn der Nutzer seinen Verbrauch mit dem des Durchschnitts vergleichen möchte, werden ab sofort diese neu ermittelten Werte verwendet, welche beim Admin gespeichert sind.

Wenn der Admin sich dafür entscheidet, manuell eine Verbrauchszeitreihe anzulegen, bekommt er die Möglichkeit Tags einzutragen, denen der Eintrag zugeordnet werden soll. Außerdem kann er die Werte eintragen und Einheiten auswählen, die der Eintrag haben soll. Wenn er dann auf „bestätigen“ klickt, werden die neuen Werte als neue Vergleichswerte definiert in den Tags die eingetragen wurden.

Funktionale Anforderungen Es muss gewährleistet werden, dass beim manuellen Eintragen die Tags tatsächlich existieren. Die eingetragenen Werte werden auf Plausibilität untersucht.

5 Qualitätsziele und -anforderungen

5.1 Benutzerfreundlichkeit für den Benutzer

Da unsere Zielgruppe sehr breit gefächert ist und wir mit Nutzern rechnen müssen, welche nicht technikaffin sind, hat die Beachtung der Benutzerfreundlichkeit eine sehr hohe Priorität. Dies gilt allerdings in erster Linie nicht für Support oder Admin-Oberflächen, da diese geschult werden können (siehe 2.3).

Die angezeigten Daten sollten möglichst viele Kundenbedürfnisse befriedigen, übersichtlich und leicht verständlich sein. Um das zu gewährleisten, sollte man vor Fertigstellung des Produkts Prototypen an mögliche zukünftige Kunden verteilen und Usability-Tests durchgeführt werden. In diesen Tests sollten dann die wichtigsten Funktionen, wie Registrierung, Verbräuche eintragen oder Statistiken einsehen unter die Probe gestellt werden. Hier kann auch ggf. die Zeit gemessen werden, die ein Benutzer benötigt, um bestimmte Aktionen durchzuführen. Außerdem kann man den Nutzer stichprobenartig bitten, eine Bewertung oder Kritik zu hinterlegen zur Benutzerfreundlichkeit.

Weiterhin kann analysiert werden, ob bei dem Support Fragen zum Ausführen oder Finden einer konkreten Funktionalität besonders häufig vorkommt. Gegebenenfalls wäre eine FAQ-Seite dann hilfreich.

5.2 Verfügbarkeit

Die abzurufenden Daten sollten jederzeit verfügbar sein, damit der Benutzer immer die Möglichkeit hat, seine Verbräuche einzusehen und vergleichen zu können. Zusätzlich sollen auch die externen Verbrauchs-Einlese-Geräte jederzeit in der Lage sein, neue Daten ins System einzupflegen. Unser Ziel ist, dass im Normalbetrieb das System innerhalb eines Monats maximal für eine Stunde nicht erreichbar ist.

Um das zu gewährleisten, greift man jeden Tag an mehreren Uhrzeiten auf das System zu und dokumentiert, ob man vom System eine Antwort bekommt

und wie lange das dauert. Daraus wird analysiert, ob es bestimmte Uhrzeiten, Wochentage oder Monate gibt, an denen mit einem höheren Betrieb zu rechnen ist, als an den anderen und es kann dementsprechend das System auf diese Zeiten angepasst werden, damit die Verfügbarkeit trotzdem gewährleistet werden kann.

Es sollte auch regelmäßig getestet werden, ob das System im Hochlastbetrieb erreichbar ist, indem man auf dem System eine besonders hohe Menge an Anfragen simuliert und währenddessen probiert, ob das System immer noch verfügbar ist.

Im Fall eines Ausfalls des Systems sollte es nach spätestens sechs Stunden wieder erreichbar sein. Um dieses Ziel zu testen, messen wir bei einem Totalausfall die Zeit, die das System lahmgelegt ist. Idealerweise erhalten wir den Anfangszeitpunkt durch etwaige Logs oder ähnliche Maßnahmen, die in der Umsetzung einer grundsätzlichen Wartbarkeit (siehe 5.5) bzw. der Analysierbarkeit (siehe 5.4) entstehen.

5.3 Wiederherstellbarkeit

Beim Versagen einer Komponente oder des ganzen Systems soll nach zwei Stunden bemerkt werden, dass ein Problem aufgetreten ist und nach spätestens sechs Stunden muss das System wieder verfügbar sein. Zusätzlich sollte stets sichergestellt werden, dass Daten, die älter als 5 Minuten sind, bei einem Ausfall des Systems rekonstruierbar sind. Etwaige Zahlungsinformation sollten niemals verloren gehen.

Dazu sollten Trainingsszenarios geplant werden, bei denen ein Versagen des Systems simuliert wird. Es wird gemessen, wie lange es dauert, das System wieder verfügbar zu machen, und es kann beobachtet werden, wie aktuell die Daten sind, welche vor dem Ausfall gespeichert wurden.

5.4 Analysierbarkeit und Modifizierbarkeit

Da ein neues Produkt geschaffen werden soll, muss unbedingt die Analysierbarkeit und Modifizierbarkeit gewährleistet sein, um viele anfängliche Fehler möglichst schnell auszuräumen. Dazu sollen Fehlermeldungen des Systems innerhalb von 2h wahrgenommen und innerhalb von weiteren 3h lokalisiert werden. Weiterhin muss gut dokumentiert werden, was das System wann macht, bzw. welche Fehler auftreten, z.B. in einer Logdatei.

Ein ganz einfaches Testszenario wäre, die Zeit zu messen beginnend von dem Auftreten des Fehlers, bis der Fehler wahrgenommen, dann lokalisiert und zuletzt behoben wurde.

5.5 Weitere Anforderungen

Hier sind weitere Anforderungen beziehungsweise Erwartungen an das System formuliert, die eingehalten werden sollen, aber nicht dieselbe Priorität, wie die oben genannten Qualitätsanforderungen, genießen.

Wartbarkeit Grundsätzliche Wartbarkeit ist natürlich Grundlage eines langlebigen Produkts. Dazu zählt eine fundamentale Code-Dokumentation bzw. -Kommentierung, Einhaltung von Code-Guidelines und, wie beim Punkt Analysierbarkeit (siehe 5.4) schon angesprochen, mögliche Logdateien.

Sicherheitsanforderungen Um die Nutzerdaten ausreichend zu schützen, muss auch ein Augenmerk auf Sicherheit gelegt werden. Wir gehen davon aus, dass die Verbindung zum Webserver stets verschlüsselt ist. Außerdem sollte die Möglichkeit von brute-Force-Attacks zum Beispiel bei der Anmeldung eingeschränkt werden.

Im Falle, dass ein Angreifer in unser System eindringt, muss in der Datenablegung Maßnahmen gegen Datenklau getroffen werden, besonders in Bezug auf sensible Daten (Passwort, Wohnort, persönliche Verbräuche). Möglich wäre das Salzen und Hashen von Passwörtern und das Speichern der Daten an verschiedenen Orten.

Datenschutzanforderungen Es müssen nationale und internationale Datenschutzgesetze eingehalten werden. Nach DSGVO muss insbesondere jeder Benutzer in der Lage sein, all seine gespeicherten Daten einzusehen, herunterzuladen oder zu löschen.

5.6 Nicht-zu-berücksichtigende Anforderungen

Weil das Produkt auf einem Webserver laufen soll, ist die Übertragbarkeit des Systems kein Ziel. Außerdem ist extreme Genauigkeit der Messdaten beziehungsweise der Prognosen nicht erforderlich, da nur der generelle Trend interessant ist.

6 Andere Anforderungen

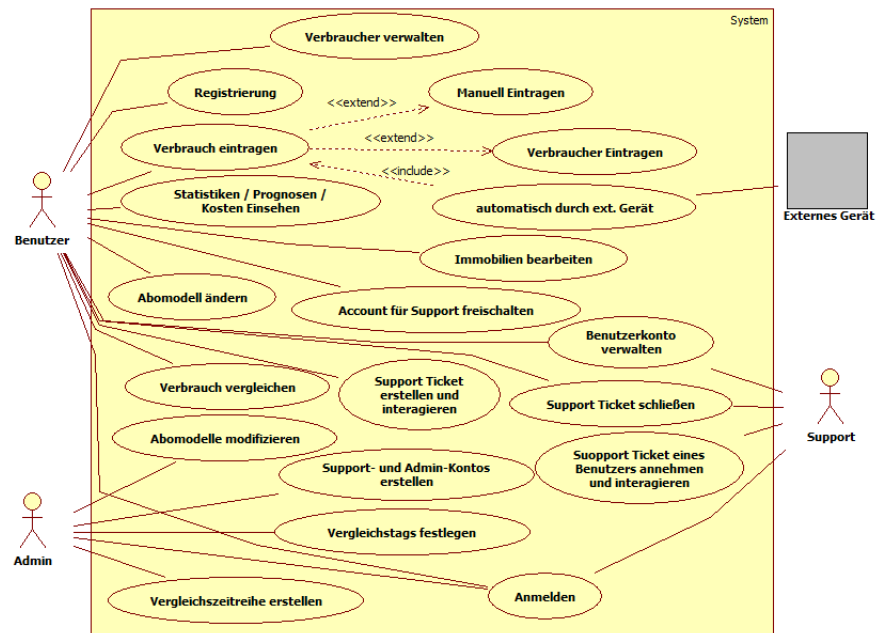
Anhang A: Glossar

- Nutzer/Benutzer: Als Benutzer beschreiben wir einen Konsumenten des Softwareprodukts.
- Support/Supporter: Als Supporter beschreiben wir eine angestellte Hilfsperson, mit eingeschränkten Berechtigungen, die dem Benutzer bei Problemen helfen kann.
- Support Ticket: Ein Support Ticket ist ein Objekt welches die Kommunikation zwischen Supporter und Benutzer zu einem bestimmten Thema bündelt.
- Admin: Der Admin ist eine angestellte Person, die Vollzugriff auf das gesamte System hat.

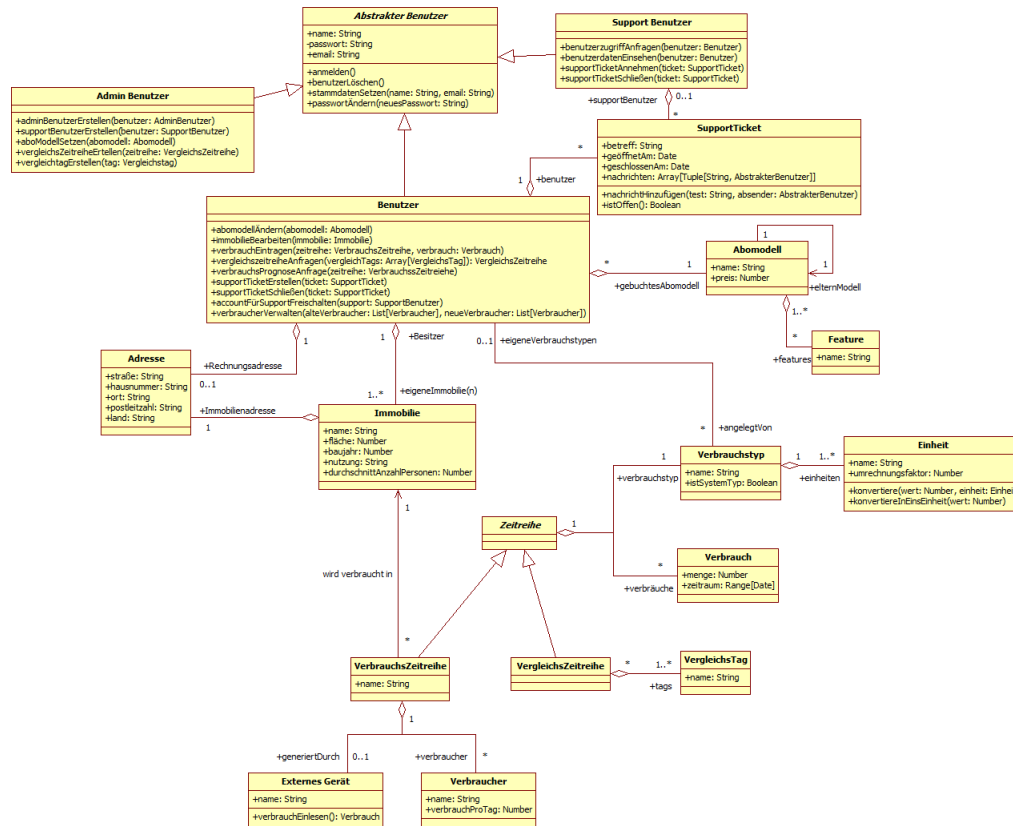
- Externes Gerät: Das Externe Gerät, ist in der Lage automatisch Strom vom Stromzähler eines Benutzers einzulesen.
- Immobilie: Immobilien sind in unserer Beschreibung Immobilien in denen der Benutzer seine Verbräuche erfassen möchte.
- Verbrauchstyp: Der Verbrauchstyp ist die Art des Verbrauchs, zum Beispiel: Strom.
- Einheit: Die Einheit beschreibt eine tatsächliche Einheit eines Verbrauchstyps, zum Beispiel: Kilowattstunde
- Einseinheit: Die Einseinheit ist eine Einheit mit dem Umrechnungsfaktor 1. Diese muss für jeden Verbrauchstyp definiert sein.
- Verbrauch: Der Verbrauch modelliert den tatsächlichen Wert des Verbrauchs in der Einseinheit über einen gewissen Zeitraum.
- Zeitreihe: Eine Zeitreihe modelliert einen Verbrauch mit einem Verbrauchstyp.
- Vergleichszeitreihe: Als Vergleichszeitreihen beschreiben wir Zeitreihen, die wir aus Nutzerdaten generieren oder aus externen Quellen erstellen. Diese dienen lediglich zum Vergleichen des Verbrauchs eines Benutzers.
- Verbraucher: Ein Verbraucher ist ein Gerät einer Immobilie welches einen konstanten Verbrauch hat und somit automatisch täglich in eine Verbrauchszeitreihe eingetragen werden kann.

Anhang B: Modelle

6.1 Use-Case Diagramm



6.2 Klassendiagramm



Anhang C: Offene Punkte

- Die Zuteilung der Features zu den Abomodellen.
- Die konkrete Wahl der unterstützten Zahlungsanbietern.
- Alle funktionalen Anforderungen die mit „Noch zu diskutieren“ beschrieben sind.
- Die konkreten Anforderungen an die Schnittstelle für externe Geräte.