

Institut für Softwaretechnologie

Abteilung Software Engineering

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Bachelorarbeit Nr. 112

Konzeption und Realisierung einer Anwendung zur Visualisierung von Maschinendaten

Matthias Milan Strljic

Studiengang: Softwaretechnik

Prüfer: Prof. Dr. rer. nat. Stefan Wagner

Betreuer: M. Sc. Asim Abdulkhaleq

begonnen am: 04.02.2014

beendet am: 29.04.2014

CR-Klassifikation: J.7

Konzeption und Realisierung einer Anwendung zur Visualisierung von Maschinendaten

Bachelorarbeit



Autor: **Matthias, Strljic**
Matrikelnummer: 2675475

Betreuung: Institut für Steuerungstechnik
der Werkzeugmaschinen und Fertigungseinrichtungen

Prof. Dr. rer. nat. Stefan Wagner

B.Eng. Michael Faller, M. Sc. Asim Abdulkhaleq

Universität Stuttgart

Studiengang: Softwaretechnik

Datum: 04.02.2014

Inhaltsverzeichnis

Abbildungsverzeichnis	7
Tabellenverzeichnis	7
Begriffsverzeichnis	8
Zusammenfassung	9
1 Einleitung	11
1.1 Motivation	11
1.2 Zielsetzung	11
1.3 Umsetzung	12
1.4 Anforderungen	13
1.4.1 Funktionale Anforderungen	13
1.4.2 Nichtfunktionale Anforderungen	14
2 Stand der Technik und Grundlagen	15
2.1 Grundlagen Fertigungsmaschinen	15
2.2 Grundlagen Evergreen	16
2.3 Grundlagen Maschinendaten	17
2.4 Stand der Technik bezüglich der Observierung von Fertigungsmaschinen	18
3 Recherche der Darstellungsformen	19
3.1 Liniendiagramm	19
3.2 Tabellen Darstellung	20
3.3 Balkendiagramm	20
3.4 Textuelle Darstellung	21
4 Recherche über Smartphone Betriebssysteme	22
4.1 IOS	22
4.2 Android	22
4.3 Windows Phone	23
4.4 Symbian	23
4.5 Blackberry OS	24
5 Analyse der Programmierplattformen für Smartphone-Anwendungen	25
5.1 Kriterien	25
5.1.1 Webservice Unterstützung	25
5.1.2 Diagrammdarstellung	25
5.1.3 Smartphone fähig	26
5.1.4 Stabile Version der Plattform ist verfügbar	26
5.1.5 Performanz	26

5.1.6	Marktabdeckung.....	26
5.2	QT 5.2 BETA.....	27
5.2.1	Webservice Unterstützung.....	27
5.2.2	Diagrammdarstellung.....	27
5.2.3	Smartphone fähig.....	27
5.2.4	Stabile Version der Plattform ist Verfügbar.....	27
5.2.5	Performanz.....	28
5.2.6	Marktabdeckung.....	28
5.3	HTML + JavaScript Anwendung.....	28
5.3.1	Webservice Unterstützung.....	28
5.3.2	Diagrammdarstellung.....	29
5.3.3	Smartphone fähig.....	29
5.3.4	Stabile Version der Plattform ist verfügbar.....	29
5.3.5	Performanz.....	29
5.3.6	Marktabdeckung.....	29
5.4	GOOGLE CHROME APP.....	30
5.4.1	Webservice Unterstützung.....	30
5.4.2	Diagrammdarstellung.....	30
5.4.3	Smartphone fähig.....	31
5.4.4	Stabile Version der Plattform ist verfügbar.....	31
5.4.5	Performanz.....	31
5.4.6	Marktabdeckung.....	31
5.5	Android App.....	31
5.5.1	Webservice Unterstützung.....	32
5.5.2	Diagrammdarstellung.....	32
5.5.3	Smartphone fähig.....	32
5.5.4	Stabile Version der Plattform ist verfügbar.....	32
5.5.5	Performanz.....	32
5.5.6	Marktabdeckung.....	33
5.6	IOS App.....	33
5.6.1	Webservice Unterstützung.....	33
5.6.2	Diagrammdarstellung.....	33
5.6.3	Smartphone fähig.....	33
5.6.4	Stabile Version der Plattform ist verfügbar.....	34
5.6.5	Performanz.....	34

5.6.6	Marktabdeckung.....	34
5.7	Blackberry App	34
5.7.1	Webservice Unterstützung	35
5.7.2	Diagrammdarstellung	35
5.7.3	Smartphone fähig.....	35
5.7.4	Stabile Version der Plattform ist verfügbar	35
5.7.5	Performanz	35
5.7.6	Marktabdeckung.....	35
5.8	Windows Phone App	36
5.8.1	Webservice Unterstützung	36
5.8.2	Diagrammdarstellung	36
5.8.3	Smartphone fähig.....	36
5.8.4	Stabile Version der Plattform ist verfügbar	36
5.8.5	Performanz	37
5.8.6	Marktabdeckung.....	37
5.9	Übersicht und Resultat	37
6	Konzeption	38
6.1	Anwendungsfälle.....	38
6.1.1	Authentifizierung.....	38
6.1.2	Konfiguration	40
6.2	Prototyp.....	43
6.2.1	Anmeldefenster.....	44
6.2.2	Serverliste editieren Dialog	45
6.2.3	Hauptfenster	46
6.2.4	Optionen.....	47
6.2.5	Tab-Fenster verwalten Dialog	48
6.2.6	Tab-Fenster editieren Dialog	49
6.2.8	Einstellungen Dialog.....	50
6.2.9	Anmeldung wiederholen Dialog	51
7	Systemmodell.....	52
7.1	Komponenten.....	52
7.1.1	Smartphone Laufzeit Umgebung	53
7.1.2	Maschine	53
7.1.3	Evergreen Server.....	53
7.1.4	Kommunikationskomponente.....	54

7.1.5	Aktualisierungskomponente.....	55
7.1.6	Datenverwaltungskomponente	56
7.1.7	Visualisierungskomponente.....	57
8	Implementierung.....	59
8.1	Entwicklungsprozess	59
8.2	Resultat.....	59
9	Zusammenfassung und Ausblick	67
10	Literaturverzeichnis	68
11	Anhang.....	70
11.1	Anwendungsfälle (Use Cases)	70
11.1.1	Akteure.....	70
11.1.2	Prioritäten.....	70
11.1.3	Anmelden.....	71
11.1.4	Abmelden.....	73
11.1.5	Diagramm entfernen	75
11.1.6	Neues Diagramm hinzufügen.....	76
11.1.7	Variable aus einem Diagramm entfernen.....	77
11.1.8	Variable in ein Diagramm hinzufügen.....	79
11.1.9	Tab-Fenster erstellen.....	81
11.1.10	Tab-Fenster entfernen	82
11.1.11	Aktives Tab-Fenster wechseln	83
11.1.12	Maschinen Optionen einsehen.....	84
11.1.13	Variable in globaler Liste manuell hinzufügen.....	85
11.1.14	Variable aus globaler Liste manuell entfernen	86
11.1.15	Externe Variablen Listen einlesen als Datei.....	87
11.1.16	Evergreen Server der Serverliste hinzufügen.....	89
11.1.17	Evergreen Server aus der Serverliste entfernen	90
11.1.18	Anmeldung wiederholen.....	91

ABBILDUNGSVERZEICHNIS

Abbildung 1 Umsetzung	12
Abbildung 2 Evergreen	16
Abbildung 3 Maschinendaten Abstraktion.....	17
Abbildung 4 Liniendiagramm Darstellungsform	19
Abbildung 5 Tabellen Darstellungsform	20
Abbildung 6 Balkendiagramm Darstellungsform.....	20
Abbildung 7 Textuelle Darstellung.....	21
Abbildung 8 Authentifizierungsanwendungsfälle.....	38
Abbildung 9 Konfigurationsanwendungsfälle	40
Abbildung 10 Anmeldefenster.....	44
Abbildung 11 Serverliste editieren Dialog	45
Abbildung 12 Hauptfenster	46
Abbildung 13 Optionen.....	47
Abbildung 14 Tab-Fenster verwalten.....	48
Abbildung 15 Tab-Fenster editieren.....	49
Abbildung 16 Einstellungen	50
Abbildung 17 Anmeldung wiederholen.....	51
Abbildung 18 Komponentendiagramm	52
Abbildung 19 Datenmodell	56
Abbildung 20 System Klassendiagramm.....	58
Abbildung 21 Anmeldefenster Resultat.....	60
Abbildung 22 Serverliste editieren Resultat.....	60
Abbildung 23 Hauptfenster Resultat	61
Abbildung 24 Optionen Resultat	62
Abbildung 25 Maschineneinstellungen Dialog Resultat	63
Abbildung 26 Tab-Fenster verwalten Dialog Resultat	64
Abbildung 27 Tab-Fenster editieren Dialog Resultat	65
Abbildung 28 Diagramm editieren Dialog Resultat.....	66

TABELLENVERZEICHNIS

Tabelle 1 Programmierplattform Analyse Übersicht.....	37
---	----

BEGRIFFSVERZEICHNIS

Paperprototype	Ein Paperprototype oder auch Papierprototyp ist ein Prototyp innerhalb einer Softwareentwurfsphase. Dieser beschreibt eine graphische Benutzeroberfläche in Form einer Abbildung oder Zeichnung. Er besitzt keine Funktionalität und simuliert keine Benutzereingaben. Eine solche Form von Prototyp eignet sich für den Entwurf einer groben Approximation an die angestrebte graphische Oberfläche und wird meist als preiswerte Diskussionsbasis für einen Anfangsentwurf genutzt.
NFC	NFC ist ein Kurzstreckenkommunikationsmechanismus zur Übertragung von Daten zwischen Geräten über kurze Distanzen, dessen Abkürzung für Near Field Communication steht.
Singleton-Pattern	Hierbei handelt es sich um ein Entwurfsmuster für die Implementierungsphase, das den Entwurf einer Klasse oder Deklaration beschreibt, wobei zur Laufzeit maximal eine Instanz existiert [1].
MVC - Pattern	Das Model – View – Controller Entwurfsmuster beschreibt die Aufteilung von Funktionsbausteinen einer graphischen Benutzeroberfläche [1].
Android-Activity	Eine Activity bildet innerhalb von Android ein eigenständiges Fenster dar, welches einen eigenen Kontext und Lebenszyklus zur Laufzeit verfügt [2].
Android-Intent	Intent's repräsentieren Nachrichten – Objekte innerhalb von Android, diese werden genutzt um über Nachrichten an das System neue Activity's aufzurufen oder unabhängigen Activity's Informationen zu übertragen [2].
Qt mobile	Qt mobile stellt eine Laufzeitkomponente von Qt (S. Pkt. 5.2) dar, welche die Entwicklung von mobilen Anwendungen ermöglicht. Diese Komponente wird von der Firma Digia entwickelt und kommerziell vertrieben.
JSON	JSON ist ein Textformat für die Speicherung von Information. Es basiert auf einer Untermenge von Javascript und ist plattformübergreifend verwendbar. Der Hauptvorteil liegt in der geringen Größe für deklarierende Elemente, somit sind gespeicherte Informationen in diesem Format geringfügig größer als die Information selbst [3].

ZUSAMMENFASSUNG

Diese Arbeit beschreibt den Verlauf einer Softwareentwicklung, dessen Resultat eine Smartphone-Anwendung für die Observierung von Fertigungsmaschine darstellt. Diese Anwendung soll als Demonstrator genutzt werden, um Smartphones innerhalb der Fertigung mittels Fertigungsmaschinen auf ihre Einsatzmöglichkeiten zu testen. Hierfür wurde mittels Recherchen und einer Analyse eine Wissensbasis für den Softwareentwurf geschaffen. Es resultierten mögliche Darstellungsformen, Smartphone-Betriebssysteme und Programmierplattformen. Für die Anwendung wurden aufgrund der geringen Komplexität Balkendiagramme, Liniendiagramme, Tabellen und eine textuelle Darstellung gewählt. Als Programmierplattform entschied man sich aufgrund der Performanz und eines großen Marktanteils für eine native Android-Anwendung, welche auf den Betriebssystemen Android und Blackberry lauffähig ist. Aufbauend auf dieser Wissensbasis wurde eine Android-Anwendung entworfen und anschließend umgesetzt.

1 EINLEITUNG

Innerhalb dieser Arbeit wird das Potential von Smartphones bezüglich der Observierung von Fertigungsmaschinen erforscht und in Form einer Demonstrator-Anwendung umgesetzt. Die Einleitung erläutert die Motivation bezüglich der Aufgabenstellung, deren Zielsetzung, die angestrebte Umsetzung und die vom Kunden erhaltenen Anforderungen an die Anwendung.

1.1 MOTIVATION

Durch das Studienprojekt Evergreen wurde als Nebenprodukt eine standardisierte Webservice-Schnittstelle entworfen, welche in Kombination mit einem Evergreen-Server, jedem Gerät, welches einen Webservice nutzen kann, die Kommunikation zu einer Fertigungsmaschine ermöglicht.

Die Motivation ist es nun, mobile Geräte aus dem Smartphone-Segment in die Fertigung zu integrieren und diese zur komfortablen Observierung von Maschinendaten zu nutzen. Dies stellt in Aussicht die gesamte Fertigung komfortabler zu gestalten und eventuelle Kosten einzusparen.

1.2 ZIELSETZUNG

Das primäre Ziel ist es, eine Smartphone-Anwendung zu entwickeln, welche die Observierung von Maschinendaten komfortabel ermöglicht. Die Kommunikation mit der Zielmaschine soll über einen Evergreen-Webservice organisiert werden. Dabei soll mittels Analysen über Smartphone-Betriebssysteme, Programmierplattformen für Smartphones und die Darstellungsformen, welche sich für Maschinendaten auf Smartphones eignen, die optimale Basis geschaffen werden. Auf dieser Basis soll die angestrebte Smartphone-Anwendung entwickelt und optimal gestaltet werden. Diese Anwendung soll als Beispielimplementierung sowie als Demonstrator dienen. Es soll zudem eine gute Usability geboten werden, um dies zu gewährleisten sollen sich eignende Usability-Patterns genutzt werden.

1.3 UMSETZUNG

Die gesamte Arbeit lässt sich in zwei Hauptteile unterteilen:

- Analysen und Recherchen:

In diesem Teil werden unterschiedliche zur Verfügung stehende Mittel auf ihre Eignung hinsichtlich der angestrebten Smartphone-Anwendung untersucht, dabei besteht dieser Teil aus „Recherche über Smartphone-Betriebssysteme“, „Analyse über Programmierplattformen für Smartphone-Anwendungen“ und „Recherche über passende Darstellungsformen“. Dieser Teil hat als Ziel eine Wissensbasis aufzubauen, um die nachfolgende Entwicklung einer Smartphone-Anwendung zu unterstützen.

- Entwicklung der Smartphone-Anwendung:

Aus dem voraus gegangenen „Analysen und Recherchen“ Teil sollen die zusammen getragenen Informationen genutzt werden, um die angestrebte Smartphone Anwendung zu entwickeln. Dieser Teil beschreibt diesen Prozess.

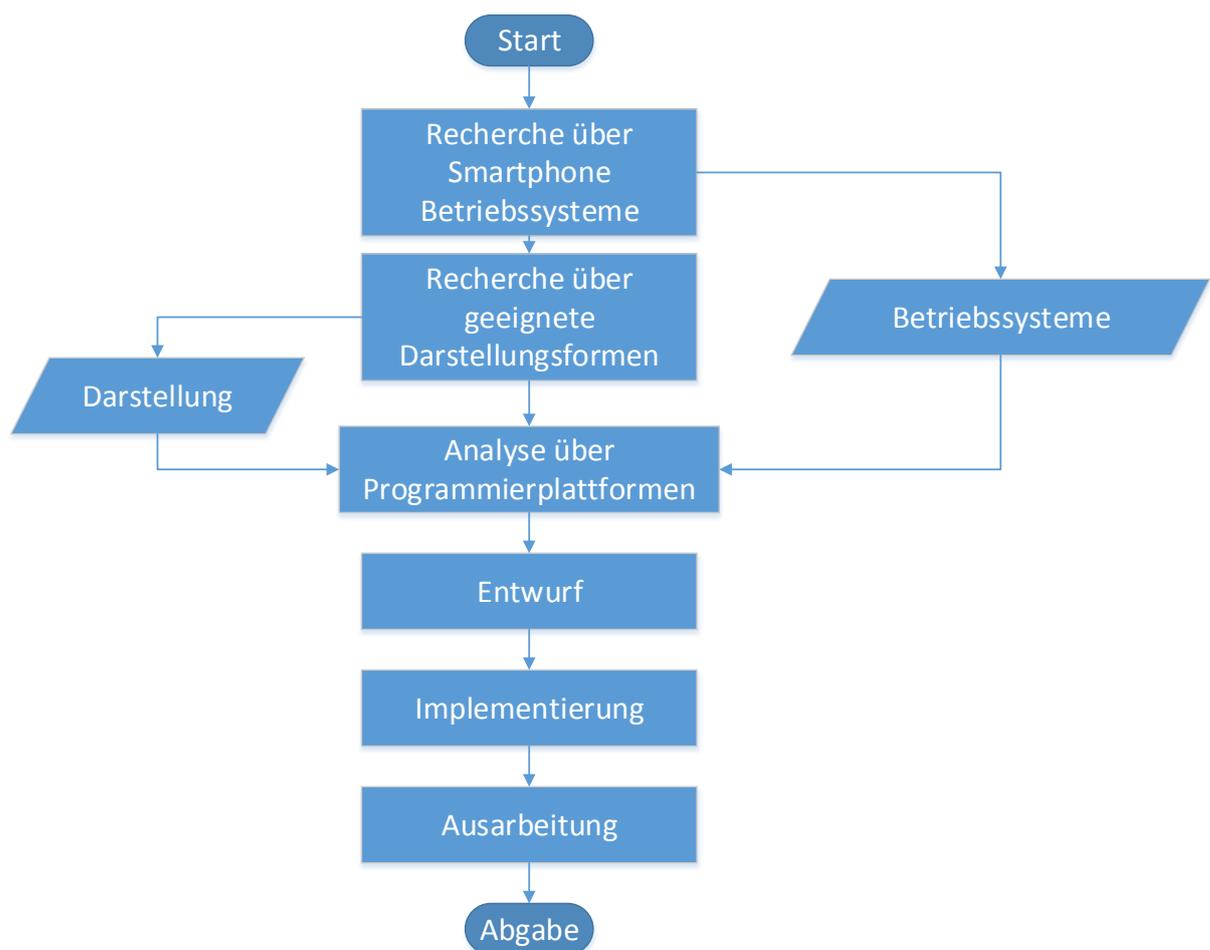


Abbildung 1 Umsetzung

1.4 ANFORDERUNGEN

Die Anforderungen umfassen die vom Kunden erfassten Anforderungen an die Anwendung, funktionale Anforderungen und nicht funktionale Anforderungen, welche wieder unterteilt werden in Musskriterien, Wunschkriterien und einer Abgrenzung.

1.4.1 FUNKTIONALE ANFORDERUNGEN

Funktionale Anforderungen beschreiben direkte Funktionen betreffende Anforderungen an die Anwendung. Mittels dieser Anforderungen wird die Funktionalität der Anwendung grob umschrieben.

1.4.1.1 Musskriterien

- Die Applikation ist in der Lage das Webservice Interface eines Evergreen Servers in der Version 1.0 [4] zu unterstützen, in den Bereichen Authentifizierung und Maschinendaten-Leseoperationen.
- Eine Aktualisierungsrate bis zu 50 Millisekunden für Variablenwerte.
- Die Darstellung von Maschinendaten in Text- und Diagrammform muss unterstützt werden.
- Es muss auf einem mobilen Gerät aus der Kategorie Smartphone ausführbar sein.
- Die Applikation muss die deutsche Sprache unterstützen.
- Die Auswahl des zu interagierenden Evergreen Servers soll zur Laufzeit möglich sein.
- Eine kontinuierliche Aktualisierung der zu observierenden Werte, welche wiederum zu einer Aktualisierung der Darstellung führt.

1.4.1.2 Wunschkriterien

- Die Applikation muss in der Lage sein, die Verwendung des Gerätes auf diese zu beschränken, um ein Gerät nur für diesen Zweck zu nutzen und mögliche Ablenkungen der Nutzer zu vermeiden.
- Eine Aktualisierungsrate bis zu 5 Millisekunden für Variablenwerte.
- Mehrere unterstützte Sprachen, darunter Englisch.
- Die Applikation lässt sich auf allen Smartphones ausführen.
- Die zur Laufzeit bereit gestellte Auswahl an Evergreen Server soll wiederum zur Laufzeit manipuliert werden können. Diese Manipulation sollte persistent sein.
- Einstellungen für die kontinuierliche Aktualisierung der zu observierenden Werte, welche zu einer Aktualisierung der Darstellung führt.

1.4.1.3 Abgrenzungskriterien

- Es handelt sich um keine portierte Evergreen-Client-Anwendung [4], das Manipulieren von Maschinendaten soll nicht unterstützt werden.
- Die Manipulation, das Auslesen von Benutzerrechten oder Benutzereinstellungen soll nicht unterstützt werden.

1.4.2 NICHTFUNKTIONALE ANFORDERUNGEN

Nichtfunktionale Anforderungen entsprechen Anforderungen, welche keiner einzelnen Funktion der Anwendung zu geordnet werden, sondern Anforderungen, welche die gesamte Anwendung betreffen und keine direkte Funktionalität beschreiben.

1.4.2.1 Musskriterien

- Das Observieren von bis zu 50 Maschinenwerten sollte möglich sein, ohne dass Performanz-Probleme die Handhabung der Applikation für einen Nutzer beeinflusst oder gar unmöglich macht.
- Eine fluide Bedienbarkeit muss gewährleistet werden.
- Es muss möglich sein sich in die Bedienung einzuarbeiten.

1.4.2.2 Wunschkriterien

- Die Bedienung sollte für eine Anwendung aus dem Smartphone Segment intuitiv sein und sich an verbreiteten Nutzungsschemata orientieren, woraus ein minimaler Einarbeitungs-Aufwand entstehen sollte.

1.4.2.3 Abgrenzungskriterien

- Keine

2 STAND DER TECHNIK UND GRUNDLAGEN

2.1 GRUNDLAGEN FERTIGUNGSMASCHINEN

Eine Fertigungsmaschine ist eine Produktionsstätte, die aus Rohmaterialien, Fertigungszwischenprodukten oder beides in Kombination, mittels eines oder mehrerer Werkzeuge, ein Produkt herstellt, dieses stellt ein Endprodukt oder ein weiteres Zwischenprodukt eines Fertigungsprozesses dar.

Der Aufbau einer solchen Maschine unterscheidet sich meist stark. Diese Variation ist der Vielzahl an unterschiedlichen Herstellern sowie Einsatzgebieten geschuldet. Jedoch bestehen sie auf einer sehr hohen Abstraktion aus denselben Komponenten:

- Zufuhr Komponente für Materialien.
- Verarbeitungskomponente, welche das Material aktiv verarbeitet meist auch Werkzeug genannt.
- Sensorkomponente, welche Maschinendaten liefert, aus denen auf den aktuellen Zustand der Maschine geschlossen werden kann.
- Steuerungskomponente, diese steuert die Verarbeitungskomponente anhand der Maschinendaten aus der Sensorkomponente.
- Kommunikationskomponente, erlaubt die Kommunikation mit der Maschine und das Auslesen oder Manipulieren von Maschinendaten, um auf die Steuerungskomponente extern einzuwirken oder sich über den aktuellen Zustand der Maschine zu informieren.
- Auswurf Komponente diese Komponenten ist für die Ausgabe des Produktes der Maschine zuständig.

2.2 GRUNDLAGEN EVERGREEN

Bei Evergreen handelt es sich um ein Studienprojekt der Universität Stuttgart, das am Institut für Steuerungstechnik und Werkzeugmaschinen (ISW) entwickelt wurde.

Das Ziel dieses Projektes war es ein Framework zu entwickeln mittels dessen ein standardisiertes Verfahren eingeführt werde, das zur erleichterten Entwicklung von Graphischen Benutzeroberflächen für Fertigungsmaschinen dienen sollte. Es teilt sich in drei Komponenten ein.

- Die zu bedienende Maschine, welche über ihre Kommunikationskomponente mit einem Evergreen-Server verbunden ist.
- Der Evergreen-Server, welcher mit der zu bedienenden Maschine verbunden ist und über einen Evergreen-Webservice [4] Funktionen zum Interagieren mit der Maschine bereitstellt. Der Server selbst besitzt noch ein Benutzer- und Rechtemanagement um kontrollierte Zugriffe zu gewährleisten.
- Der Evergreen-Client, dieser nutzt den bereit gestellten Evergreen-Webservice, um mit dem Evergreen-Server zu kommunizieren und die Maschine zu steuern.

Innerhalb dieser Arbeit wird die zu entwickelnde Smartphone Anwendung als eine Abwandlung eines Evergreen-Clients auftreten, welche eine eingeschränkte Funktionalität des Evergreen-Webservices [4] nutzt. Der genutzte Evergreen-Server sowie die zu observierende Maschine befinden sich in einem nicht modifizierten Zustand. Die Abbildung 2 Evergreen bildet die Verbindung von Evergreen-Client, Evergreen-Server und Fertigungsmaschine graphisch ab [4].

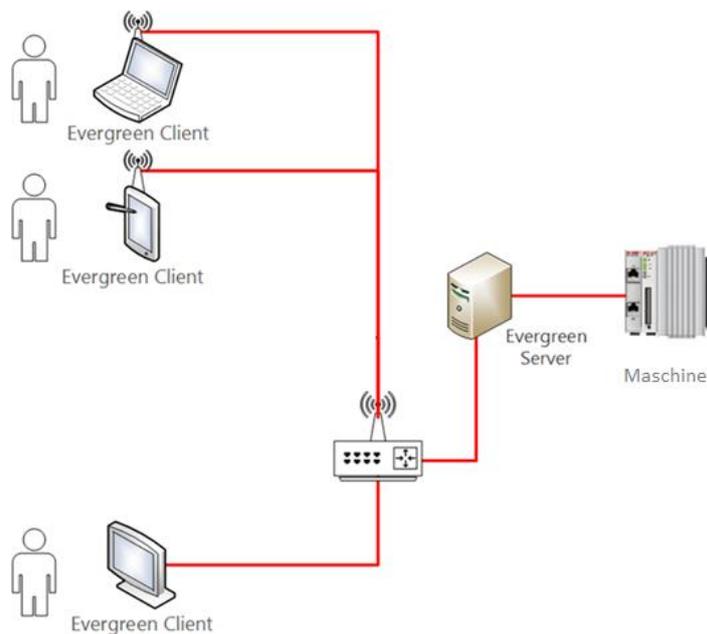


Abbildung 2 Evergreen

2.3 GRUNDLAGEN MASCHINENDATEN

Maschinendaten stellen die Informationen einer Maschine dar. Sie sind aussagekräftig über den aktuellen, geplanten oder Soll-Zustand der Maschine und ihres aktiven Prozesses. Es existieren dabei starke Parallelen zu gängigen Programmiersprachen wie C/C++ in denen Werte in Variablen gespeichert werden können. Es handelt sich hierbei um primitive, angepasste oder Listentypen, welche über die Kommunikationskomponente von der Umgebung ausgelesen oder manipuliert werden können. Durch den Evergreen-Webservice ist es möglich primitive Werte sowie Listentypen auszulesen. Die Abbildung 3 Maschinendaten Abstraktion liefert einen Zusammengefassten Überblick.

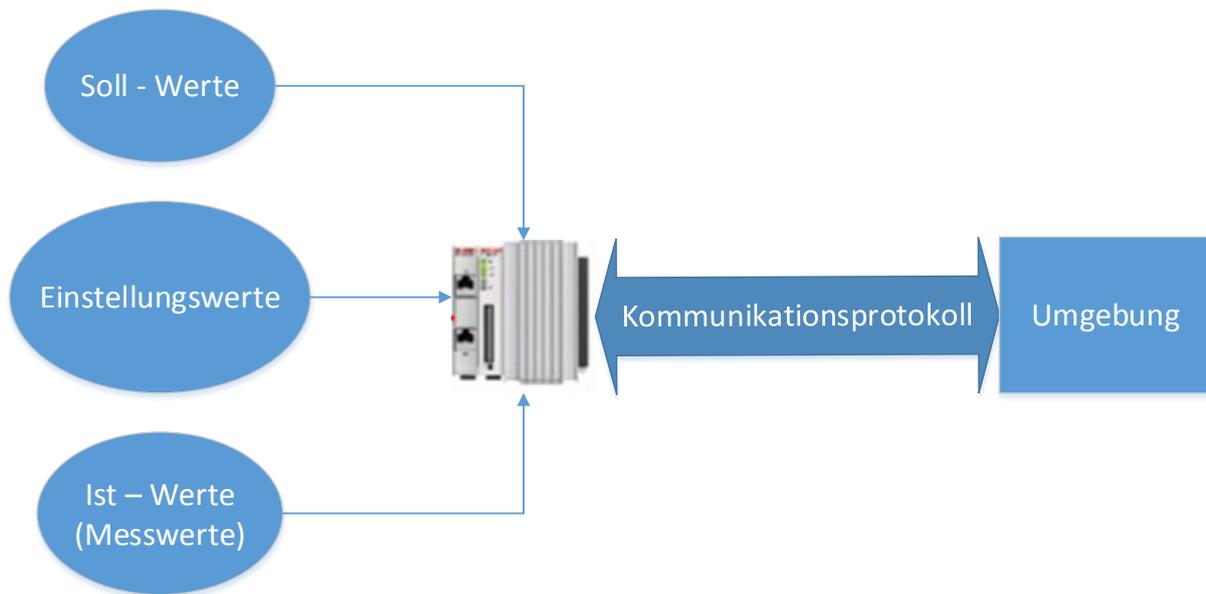


Abbildung 3 Maschinendaten Abstraktion

2.4 STAND DER TECHNIK BEZÜGLICH DER OBSERVIERUNG VON FERTIGUNGSMASCHINEN

Die Observierung von Maschinendaten zum Zeitpunkt dieser Arbeit ist simpel gehalten. Jeder Hersteller bietet zu seinen Maschinen Optionen an, um sich den Status dieser auslesen zu lassen und gegebenenfalls zu manipulieren. Dabei ist die Variation groß, da jede Anzeige und jede Schaltfläche an das Einsatzgebiet angepasst wurde. Es existieren Maschinen mit integrierten Displays, Tablets die über Kabel mit einer Maschine verbunden sind und zusammen geschaltete Zentralen, in denen Werte von mehreren Maschinen angezeigt und manipuliert werden können. Die Gemeinsamkeit aller ist die Verbindung mit Kabeln zu einander, dies beschränkt die Reichweite und die Flexibilität der Observierung einer Maschine. Der Grund hierfür ist zum einen die Sicherheit, ein Mitarbeiter muss nahe an der Maschine sein um bei Gefahr diese abzuschalten oder in den Prozess eingreifen zu können, zum anderen die Latenz zwischen Übertragungen, viele Maschinen werden über diese Anzeigen mit gesteuert und eine zeitliche Verzögerung könnte Probleme oder gar Schäden verursachen. Zudem ist jede dynamische Anpassung für einen Kunden kostenintensiv und findet seine Grenzen sobald ein Kunde mehrere Maschinen von unterschiedlichen Herstellern besitzt. Trifft so etwas ein, muss meist ein dritter Hersteller hinzugezogen werden, der proprietäre Hardware sowie Software herstellt, welche auf dieses spezielle Einsatzgebiet zugeschnitten wurde, dies wiederum ist kostenintensiv. So existiert in der Fertigung immer eine spezielle Form der Observierung, die immer funktionsgebunden, kosten gebunden und an das Einsatzgebiet angepasst ist.

3 RECHERCHE DER DARSTELLUNGSFORMEN

Das Ziel dieses Kapitels ist es für die vorhandenen Maschinendaten geeignete Darstellungsformen zu finden. Es sollen Darstellungsformen für einzelne Werte sowie für zusammenhängende Listen gefunden werden, diese können sich überschneiden. Zudem soll es möglich sein den Darstellungsschwerpunkt, durch die Wahl einer Darstellungsform, zu verändern um zwischen der Anzeige verschiedener Stadien von Werten und dem Werteverlauf zu wechseln. Diagrammformen werden mit einer graphischen Vorschau und einer Begründung vorgestellt, warum sie als geeignet für die angestrebte Anwendung eingestuft wurden.

3.1 LINIENDIAGRAMM

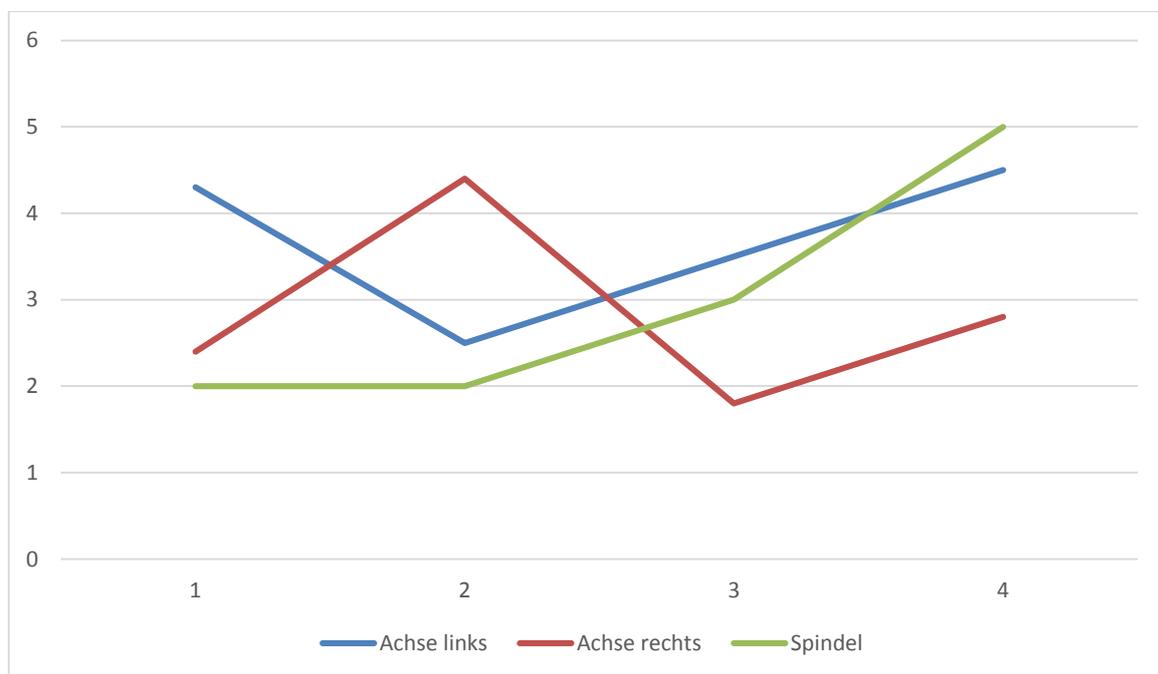


Abbildung 4 Liniendiagramm Darstellungsform

Liniendiagramme eignen sich für die Darstellung eines Werteverlaufs. Dabei handelt es sich um ein 2 dimensionales Diagramm dessen horizontale Achse für den Zeitverlauf genutzt werden kann und die vertikale Achse um Werte von Maschinendaten darzustellen. Diese Diagrammform eignet sich für die angestrebte Anwendung um einzelne sowie Listen Werte darzustellen und deren Verlauf übersichtlich darzustellen um Extrema hervorzuheben.

3.2 TABELLEN DARSTELLUNG

Wert	Ist - Wert	Maximum	Minimum
Beispiel Wert	41,2	52	1,4

Abbildung 5 Tabellen Darstellungsform

Die Tabellen Darstellung eignet sich für die Darstellung von Ist-Werten und fixen Attributen, welche das Minimum, Maximum, Median oder den arithmetischen Mittelwert umfassen können. Dies eignet sich für die angestrebte Anwendung um einzelne sowie Listen Werte darzustellen und diese auf fixe Attribute über Zeit zu observieren. In Abbildung 5 Tabellen Darstellungsform befindet sich eine Tabelle, welche den aktuellen Wert sowie das Maximum und Minimum einer Variablen angibt.

3.3 BALKENDIAGRAMM

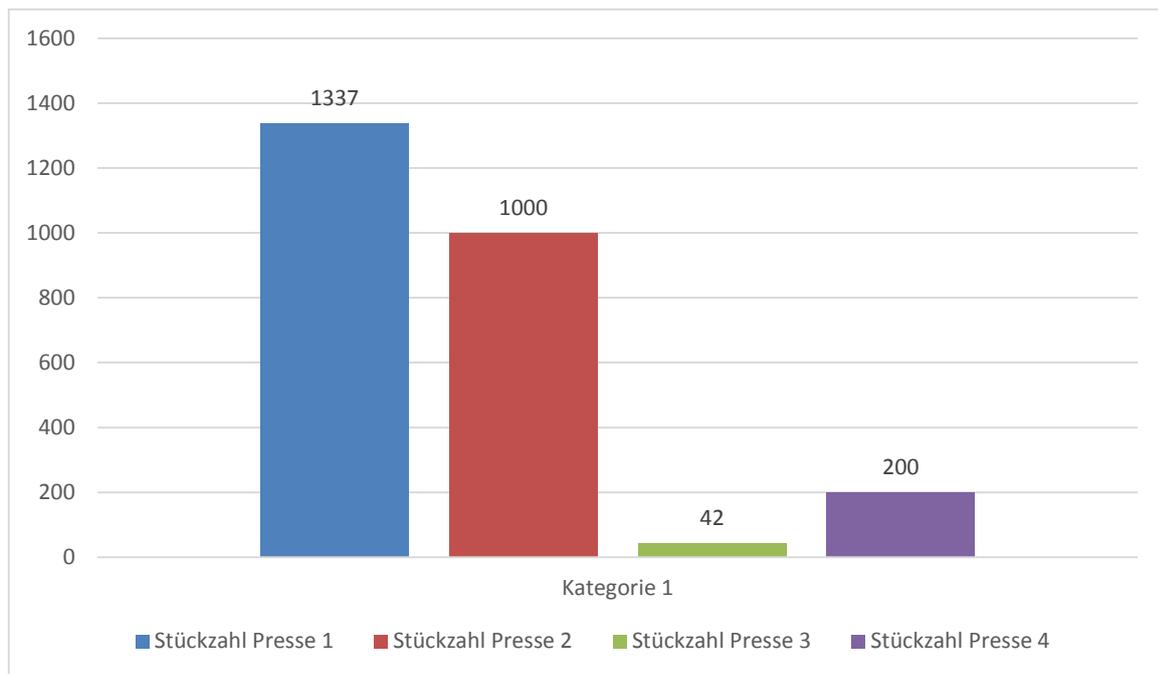


Abbildung 6 Balkendiagramm Darstellungsform

Die Balkendiagramm-Darstellung eignet sich für die Visualisierung von Ist-Werten und deren Differenzen zu weiteren Ist-Werten. Balkendiagramme werden innerhalb dieser Arbeit nicht verwendet um Werteverläufe darzustellen, da dies auf Smartphone-Anzeigen aus Platzgründen als nicht optimal angesehen wurde. Diese Diagrammform eignet sich für die angestrebte Anwendung um einzelne sowie Listen Werte darzustellen und einen schnellen Überblick über mehrere Werte zu erhalten.

3.4 TEXTUELLE DARSTELLUNG

Beispiel Wert: 1337,42

Abbildung 7 Textuelle Darstellung

Die textuelle Darstellung eignet sich um einzelne Ist-Werte darzustellen und kann innerhalb der angestrebten Anwendung für diese genutzt werden. Diese Darstellungsform eignet sich für die Visualisierung von einzelnen Werten. Hierbei wird der zu observierende Wert innerhalb eines Präfix sowie Suffix eingeschlossen, welche aus einem statischen Text bestehen. Innerhalb der Abbildung 7 Textuelle Darstellung wird der Wert 1337,42 mittels des Präfix „Beispiel Wert:“ sowie eines leeren Suffix umschlossen.

4 RECHERCHE ÜBER SMARTPHONE BETRIEBSSYSTEME

4.1 IOS

- **Hersteller:** Apple
- **Hersteller unterstützte Programmierplattform:** Objective C
- **Aktuelle Version:** 7.0.6
- **Erst Veröffentlichung:** 09.01.2007
- **Marktanteil 2Q/2013:** 13,2% [5]
- **Unterstützte Geräte:** iPhone, iPad, iPod, Apple Smart TV

Beschreibung:

IOS basiert auf einem Mac OS X – Kern, welcher wiederum auf einem Unix- Kern basiert. Es wird von Apple entwickelt und fand seinen Ursprung 2005, als von Steve Jobs [6] die Idee angestoßen wurde ein Telefon Betriebssystem zu entwickeln. IOS repräsentiert den Start der Smartphone-Ära und war nach seiner Veröffentlichung das erste Smartphone-Betriebssystem, das größeren Anklang in der allgemeinen Bevölkerung fand. So wurden in den ersten beiden Quartalen nach der Veröffentlichung mehr als 3 Millionen [5] Geräte verkauft. IOS – Anwendungen können nur aus dem Apple App Store bezogen werden. Die Entwicklung von Anwendungen wird durch das von Apple zur Verfügung gestellte IOS SDK ermöglicht und ist auf Mac OS X Betriebssysteme beschränkt. Bevor eine Anwendung auf dem IOS Betriebssystem installiert werden kann muss sie einen Zertifizierungsprozess von Apple durchlaufen, danach kann sie manuell installiert oder der Öffentlichkeit über den Apple App Store zur Verfügung gestellt werden. Das Betriebssystem selbst ist auf die Eingabe über Gesten ausgelegt, welche über einen berührungsempfindlichen Bildschirm eingegeben werden und ist nur für Apple eigene Geräte verfügbar, darunter zählen die Geräte der Klasse iPhone, iPad, iPod und Apple Smart TV [7] [8].

4.2 ANDROID

- **Hersteller:** Open Handset Alliance
- **Hersteller unterstützte Programmierplattform:** Java
- **Aktuelle Version:** 4.4.2
- **Erst Veröffentlichung:** 21.10.2008
- **Marktanteil 2Q/2013:** 79,3% [5]
- **Unterstützte Geräte:** Smartphone, Tablet, Smart TV, Notebook, PC

Beschreibung:

Android basiert auf einem stark angepassten Linux, es wurde für Berührungsgesten und Energieeffizienz optimiert. Es wird von der Open Handset Alliance entwickelt und seinen Start fand es 2005 als Google, Hauptmitglied der Open Handset Alliance, die Entwicklung eines Betriebssystems für Mobiltelefone anstrebte. Es handelt sich um ein Offenes Betriebssystem dessen Quellcode der Öffentlichkeit zur Verfügung steht und dabei unter der Apache V2 Lizenz steht. Anwendungen können unter jedem

Betriebssystem und in theoretisch jeder Programmiersprache verfasst werden, jedoch ist die unterstützte Programmierplattform Java und wird mit dem Android SDK unterstützt. Das Installieren von Anwendungen kann manuell oder durch zuvor installierte App Stores automatisiert werden, dabei ist der größte Vertreter der Google Play Store. Android ist mit einem Marktanteil von 79,3% [5] das am weitesten verbreitete Betriebssystem unter den Smartphones [9] [10] [2].

4.3 WINDOWS PHONE

- **Hersteller:** Microsoft
- **Hersteller unterstützte Programmierplattform:** C/C++, C#, VB.NET
- **Aktuelle Version:** 8.0.10521.155
- **Erst Veröffentlichung:** 21.10.2010
- **Marktanteil 2Q/2013:** 3,7% [5]
- **Unterstützte Geräte:** Smartphone

Beschreibung:

Windows Phone basiert auf Microsoft Windows-NT, welches in seiner Funktionalität angepasst und auf Energieeffizienz optimiert wurde. Es wird von Microsoft entwickelt und war 2010 die Reaktion auf den wachsenden Smartphone-Markt. Die Oberfläche gilt aufgrund seiner schicht gehaltenen Ansicht, welche die Anordnung von Anwendungen und Optionen auf Kacheln minimiert, als Vorreiter des Flat-Design [11]. Anwendungen selbst können nur auf dem Betriebssystem Windows, mit der Hilfe des Windows Phone SDK in den Programmierplattformen C/C++, C# und VB.NET entwickelt und dann manuell auf dem eigenen Smartphone installiert oder über einen Zertifizierungsprozess von Microsoft im Windows Store veröffentlicht werden. Die Installation von Anwendungen ist nur über den Windows Store möglich. Microsoft verfolgt bei diesem mobilen Betriebssystem eine Lizenz Strategie ähnlich zu seinem Windows Produkten und bietet gegen eine Gebühr die Nutzung von Windows Phone an, diese variiert abhängig von der Zielhardware und der Abnahmemenge. Das Betriebssystem selbst konnte sich noch nicht in den Smartphone-Markt etablieren, Windows Phone besitzt im Q2 2013 einen Marktanteil von 3,7% [5] [12] [13].

4.4 SYMBIAN

- **Hersteller:** Nokia
- **Hersteller unterstützte Programmierplattform:** C/C++, Java
- **Aktuelle Version:** Symbian^3
- **Erst Veröffentlichung:** 24.06.2008
- **Marktanteil 2Q/2013:** 0,2% [5]
- **Unterstützte Geräte:** Mobiltelefon ,Smartphone, PDA

Beschreibung:

Symbian ist eine Erweiterung der EPOC - Plattform und wurde unter dem Konsortium von Nokia, Sony Ericsson, Motorola und NTT DoCoMo entwickelt, bis Nokia 2008 alle Rechte an dem Projekt erwarb. Anwendungen können unter C/C++ und Java entwickelt werden. Die Installation von Anwendungen ist manuell möglich und konnte bis zum 01.01.2014 automatisiert über den App Store erfolgen. Die Oberfläche ist für eine Tastenbedienung ausgelegt und setzt für eine optimale Bedienung Navigationstasten voraus. Langfristig war Nokia der einzige Hersteller welcher das Betriebssystem über einen längeren Zeitraum genutzt hatte, bis dieser selbst die Unterstützung Ende 2012 einstellte. Symbian besaß bis zum 2Q 2013 einen Marktanteil von 0,2% [5] und wird langsam von den Konkurrenten IOS und Android verdrängt. Es wird nur der Vollständigkeit erwähnt [14].

4.5 BLACKBERRY OS

- **Hersteller:** Blackberry ehemalg Motion In Research
- **Hersteller unterstützte Programmierplattform:** C/C++, Java
- **Aktuelle Version:** 10.2.1.537
- **Erst Veröffentlichung:** 01.01.2006
- **Marktanteil 2Q/2013:** 2,9% [5]
- **Unterstützte Geräte:** Mobiltelefon ,Smartphone

Beschreibung:

Blackberry OS basiert auf dem Echtzeitbetriebssystem QNX und wurde von Blackberry entwickelt. Es wurde zunächst für die Tastenbedienung optimiert und im fortlaufenden Entwicklungsprozess an die Bedienung durch Berührungsgesten und berührungsempfindliche Displays angepasst. Das Betriebssystem ist nur auf mobilen Geräten der Marke Blackberry erhältlich und besitzt als besondere Funktionen eine auf Einhandbedienung optimierte Oberfläche, Push-Benachrichtigung für E-Mails, einen Schwerpunkt auf Datensicherheit und eine meist integrierte Hardwaretastatur. Blackberry Produkte richten sich, durch den Schwerpunkt auf Sicherheit, an Industriekunden. Trotz dessen erfuhr es besonders in den U.S.A eine sehr gute Resonanz in der Bevölkerung, so besaß Blackberry OS 2010 einen Marktanteil von 17%. Blackberry OS Anwendungen können über das Blackberry Nativ SDK in C/C++ oder über das Blackberry JDE in Java entwickelt werden, diese können über einen Zertifizierungsprozess der Öffentlichkeit über die BlackBerry App World zur Verfügung gestellt werden. Die Installation von Anwendungen kann manuell oder über die BlackBerry App World automatisch geschehen, eine Besonderheit bei der manuellen Installation stellt die Möglichkeit dar ab der Version 10.0 Android Anwendungen zu installieren. Im zweiten Quartal 2013 besaß Blackberry OS einen Marktanteil von 2,9% [5] [15].

5 ANALYSE DER PROGRAMMIERPLATTFORMEN FÜR SMARTPHONE-ANWENDUNGEN

In diesem Kapitel wird eine Analyse über die zur Auswahl stehenden Programmierplattformen vorgenommen. Diese Plattformen resultieren aus einer Internet Recherche in Kombination mit Literatur über die Entwicklung von Smartphone-Anwendungen. Es werden zuerst die zusammen getragenen Kriterien vorgestellt, nach diesen die einzelnen Programmierplattformen bewertet werden. Daraufhin werden die jeweiligen Programmierplattformen vorgestellt und in Bezug auf die Erfüllung dieser Kriterien bewertet. Am Ende dieses Kapitels werden innerhalb einer tabellarischen Übersicht die Ergebnisse zusammengefasst.

5.1 KRITERIEN

Im Folgenden werden die Kriterien aufgelistet, mit einer kurzen Beschreibung warum sie ausgewählt wurden und was ihre Kernpunkte sind. Diese Kriterien sind notwendig um die optimale Plattform auszuwählen, sowie zu entscheiden welche Smartphone Betriebssysteme durch die angestrebte Anwendung unterstützt werden und wie hoch die prozentuale Abdeckung des aktuellen Smartphone Marktes theoretisch liegt.

5.1.1 WEBSERVICE UNTERSTÜTZUNG

Um einen Evergreen-Webservice zu nutzen ist es nötig, dass die genutzte Plattform Webservices unterstützt. Dabei muss sie die Funktionalität des Evergreen-Webservice in der Version 1.0 [4] zugänglich machen und vollständig unterstützen. Es handelt sich um ein K.O. Kriterium. Würde eine Plattform dies nicht unterstützen, könnte sie nicht genutzt werden, da sonst die angestrebte Anwendung keine Möglichkeit besitzt mit dem Evergreen-Server zu kommunizieren und somit nicht in der Lage ist Maschinendaten anzuzeigen.

5.1.2 DIAGRAMMDARSTELLUNG

Ein wesentlicher Bestandteil der angestrebten Smartphone-Anwendung ist die Darstellung von Maschinendaten in Diagrammform. Somit muss die Programmierplattform die Darstellung von Werten in Diagrammform unterstützen. Dies ist ein K.O. Kriterium.

5.1.3 SMARTPHONE FÄHIG

Um eine Smartphone-Anwendung zu entwickeln muss die resultierende Anwendung einer Programmierplattform auf einem Smartphone ausführbar sein, sollte dies nicht der Fall sein ist diese Plattform für diesen Anwendungsfall unbrauchbar, was wiederum dieses Kriterium zu einem K.O. Kriterium macht.

5.1.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Dieses Kriterium umfasst, dass sich die genutzte Version der Programmierplattform in einer Release Version befindet, so dass man auf eine sich nicht verändernde Plattform referenzieren kann. Dies spielt eine wichtige Rolle, da genutzte Funktionalitäten mit der fortschreitenden Entwicklung einer Plattform abgewandelt und eventuell für die angestrebte Anwendung unbrauchbar würden, wenn sich die Referenzierte Version in einem Prä-Alpha, Alpha oder Beta Status befindet. Des weiteren wird mit einer stabilen Version die Zukunftssicherheit dieser Plattform assoziiert, diese sollte zum Zeitpunkt dieser Arbeit verfügbar sein und es sollten keine Ankündigungen existieren, die eine Unbrauchbarkeit dieser Plattform in Aussicht stellen. Dieses Kriterium ist ein K.O. Kriterium.

5.1.5 PERFORMANZ

Performanz ist ein wichtiges K.O. Kriterium. Die angestrebte Anwendung muss eine stabile Leistung vorweisen und darf nicht von äußeren Faktoren beeinflusst werden, welche keinen allgemeinen Einfluss auf die gesamte Leistung des Betriebssystems zur Folge hat. Zudem beinhaltet dieses Kriterium die funktionale Anforderung innerhalb eines oder mehrerer Diagramme bis zu 50 Maschinendaten zu observieren, mit einer maximalen Aktualisierungsrate von 50 Millisekunden ohne dass eine Beeinträchtigung der Bedienbarkeit zustande kommt. Für dieses Kriterium wird eine Schätzung vorgenommen, welche die prognostizierte Leistung in Bezug auf die angestrebte Anwendung beinhaltet, da dies aus zeitlichen Gründen nicht in jeder Plattform einzeln überprüft werden kann.

5.1.6 MARKTABDECKUNG

Dieses Kriterium ist das einzige nicht K.O. Kriterium und dient als Entscheidungskriterium. Sollte es am Ende dieser Analyse mehrere Programmierplattformen geben, die alle K.O. Kriterien erfüllen, wird durch dieses Kriterium entschieden, welche Plattform für die angestrebte Anwendung genutzt werden soll. Als Marktabdeckung wird in diesem Fall die prozentuale Abdeckung der im Markt vertretenen Smartphone Betriebssysteme bezeichnet. Würde eine Plattform eine Marktabdeckung von 50% erreichen würde dies bedeuten, dass Anwendungen die mittels dieser Plattform entwickelt wurden auf jedem zweiten Smartphone lauffähig sind.

5.2 QT 5.2 BETA

Bei Qt handelt es sich um ein C++ Framework für die Entwicklung von interaktiven graphischen Benutzeroberflächen. Es wurde von Trolltech am 10.07.1998 in der Version 1.40 veröffentlicht. Trolltech wurde 2008 von Nokia aufgekauft und Nokia gab das Qt- Projekt im Oktober 2011 als Open Source-Projekt frei an die Qt Project-Community. Qt eignet sich für Desktop Anwendungen oder speziell unterstützte Systeme zur Entwicklung von benutzergesteuerten Anwendungen. Zum Zeitpunkt der Recherche befindet sich Qt in der stabilen Version 5.1, welche noch keine Entwicklung von mobilen Anwendungen ermöglicht. Dies soll Teil des Funktionsumfangs von Qt 5.2 werden, welches in einem Beta Status zur Verfügung steht. Anwendungen können in den Sprachen C++, JavaScript und QML entwickelt werden. Es ist auch möglich in anderen Programmiersprachen zu entwickeln, dabei muss aber eine eingeschränkte Unterstützung toleriert werden. Die Entwicklungsumgebung von Qt-Anwendungen erfolgt im Qt Creator, diese unterstützt Syntaxhervorhebung, Quellcode Refactoring, Kontextsensitive Hilfe, Debuggen und Werkzeuge für die Simulation der entwickelten graphischen Benutzeroberfläche [16].

5.2.1 WEBSERVICE UNTERSTÜTZUNG

Mit der Qt Network-Komponente wird innerhalb von Qt eine API für Netzwerkzugriffe bereitgestellt, die es ermöglicht Webservices zu nutzen. Es werden alle benötigten Funktionen unterstützt um einen Evergreen-Webservice in der Version 1.0 [4] zu nutzen [16].

5.2.2 DIAGRAMMDARSTELLUNG

Qt Charts bietet innerhalb von Qt eine Vielzahl an unterstützten Diagrammformen. Es werden Liniendiagramme, Kreisdiagramme, Flächendiagramme, Balkendiagramme und eigen erstellte Diagrammformen unterstützt, somit wird dieses Kriterium erfüllt [16].

5.2.3 SMARTPHONE FÄHIG

Durch Qt Mobile, eine kommerziell vertriebenen Komponente für Qt der Firma Digia, innerhalb der angestrebten Version Qt 5.2 soll die Entwicklung von Anwendungen für die mobilen Betriebssysteme Android und IOS möglich sein, somit wird dieses Kriterium erfüllt [16].

5.2.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Da sich Qt 5.2 noch in einem Beta-Status befindet wird dieses Kriterium nicht erfüllt.

5.2.5 PERFORMANZ

Bei Qt mobile handelt es sich um eine Plattform übergreifende Entwicklungsumgebung, welche die Leistung von nativ entwickelten Applikationen verspricht. Dies stellt optimale Performanz in Aussicht welche dieses Kriterium erfüllt.

5.2.6 MARKTABDECKUNG

Android und IOS sind die Zielplattformen von Qt mobile mit diesen beiden Betriebssystemen wird eine Marktabdeckung von 92,5% [5]erreicht.

5.3 HTML + JAVASCRIPT ANWENDUNG

Bei HTML und JavaScript handelt es sich um zwei Technologien zur Darstellung von Webinhalten, welche in einem Browser dargestellt werden können. HTML wurde am 03.11.1992 entwickelt, es sollte zur strukturierten Datenübertragung zwischen Forschungseinrichtungen der Europäischen Organisation für Kernforschung dienen und wurde über die Jahre zu der größten Plattform für die Darstellung von Webinhalten. Die aktuelle Version ist HTML5 und wird von dem World Wide Web Konsortium entwickelt. Es dient zur statischen Darstellung von Webinhalten, die mit der Hilfe von JavaScript mit einem dynamischen Verhalten versehen werden. JavaScript ist eine Skriptsprache, welche syntaktisch an die Programmiersprache Java angelehnt ist. Es existieren darüber hinaus keine Parallelen. Sie wurde 1996 entwickelt um statischen HTML-Inhalten eine dynamische Verhaltensweise für Benutzereingaben zu ermöglichen und befindet sich in der Version JavaScript 1.8.5. Anwendungen die für Smartphones mittels HTML und JavaScript entwickelt werden, werden über eine externe Layout-Engine interpretiert und dargestellt, dies ist meist ein Browser oder ähnliches. Viele Smartphone Betriebssysteme besitzen auch eigene Komponenten, die speziell für die Darstellung von HTML Anwendungen ausgelegt sind, wie es beispielsweise in Android mittels der WebView der Fall ist. Die Entwicklung selbst wird von vielen Entwicklungsumgebungen unterstützt wie Visual Studio oder Eclipse, darin enthalten sind meist Funktionen wie Syntaxhervorhebung, Quellcode Refactoring und Werkzeuge für die Simulation der entwickelten graphischen Benutzeroberfläche.

5.3.1 WEBSERVICE UNTERSTÜTZUNG

Webservices können in Anwendungen, die mittels JavaScript entwickelt werden, durch die externe Bibliothek Ajax genutzt werden. Sie basiert selbst auf JavaScript und unterstützt jegliche Form von Webservices, welche auf eine Anbindung über das Protokoll HTTP setzen. Hierdurch wird der Evergreen-Webservice in der Version 1.0 [4]vollständig unterstützt, was dieses Kriterium erfüllt.

5.3.2 DIAGRAMMDARSTELLUNG

Diagramme können über mehrere JavaScript Bibliotheken wie Chart.js oder Google Charts erzeugt werden. Es werden alle gängigen Diagrammformen unterstützt wie Liniendiagramme, Balkendiagramme, Kuchendiagramme, Polardiagramme, Flächendiagramme, Netzdiagramme und Doughnut – Diagramme. Dies ist ausreichend um dieses Kriterium zu erfüllen [17] [18].

5.3.3 SMARTPHONE FÄHIG

Da die Darstellung von Webinhalten über einen Browser als Standard für Smartphones gilt und dies die Darstellung von HTML-Inhalten in Kombination mit JavaScript inkludiert, ist jedes Smartphone in der Lage HTML und JavaScript Anwendungen zu nutzen. Was dieses Kriterium erfüllt.

5.3.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

HTML und JavaScript besitzen Release Versionen und sind beide so intensiv in die aktuelle Form des Internets integriert, dass kein Ende von HTML oder JavaScript in Aussicht ist, wodurch dieses Kriterium erfüllt wird.

5.3.5 PERFORMANZ

Es handelt sich bei beiden Technologien um Skriptsprachen die durch eine externe Layout-Engine interpretiert werden müssen. Dies lässt die Performanz von Anwendungen stark schwanken. Die Implementierung der genutzten Layout-Engine kann dabei starke Auswirkungen haben. Zusätzlich sind Anwendungen die interpretiert werden müssen, durch die Interpreter-Schicht, zwangsläufig langsamer als native Anwendungen und müssen eventuell mit weiteren Anwendungen über den Zugang zu diesem Interpreter konkurrieren. Eine Aussage über die garantierte Leistung ist sehr schwer und darum wird dieses Kriterium nicht erfüllt, da die Leistung durch die Interpreter-Schicht stark beeinträchtigt wird [19] [20].

5.3.6 MARKTABDECKUNG

Da jedes Smartphone in der Lage ist Anwendungen aus der Kombination von HTML und JavaScript auszuführen liegt die prozentuale Marktabdeckung bei 100%.

5.4 GOOGLE CHROME APP

Es handelt sich hierbei um Anwendungen die durch den Google Browsers Google Chrome ausgeführt werden, welcher sich zum Zeitpunkt dieser Arbeit in der Version 33.0 befindet. Der Browser wird hauptsächlich von Google entwickelt und wurde am 11.12.2008 erstmals für Windows veröffentlicht. Teile des Projekts werden als Chromium – Projekt unter der BSD-Lizenz der Öffentlichkeit zur Verfügung gestellt, somit können Communitys oder andere Hersteller den Quellcode weiter entwickeln und bei Bedarf für sich nutzen. Ziel von Google Chrome Apps ist es Anwendungen, die auf die Nutzung von Webtechnologien wie HTML und JavaScript setzen, in ihrem Funktionsumfang zu erweitern sowie ihnen das „Look and Feel“ von nativ implementierten Anwendungen zu geben. Hierbei werden Google Chrome Apps nicht mehr innerhalb eines Browsers ausgeführt sondern agieren als eigenständiger Prozess, der seine eigene Anzeige besitzt. Die Entwicklung selbst unterscheidet sich minimal von der Entwicklung einer HTML und JavaScript Anwendung. Der Unterschied liegt in der zur Verfügung stehenden Bibliotheken, so wurden die Schnittstellen erweitert um Zugang zu USB, TCP/IP, Bluetooth und weiteren Plattformdiensten zu erhalten. Zudem wurden Komponenten die von Google als sicherheitskritisch eingestuft wurden überarbeitet und durch Google Chrome Bibliotheken ersetzt, welche ihren Funktionsumfang subsumieren und mehr Sicherheit garantieren sollen. Zusätzlich wurden diese Bibliotheken mit einer Komponente für die Handhabung der eigenständigen Anzeige ergänzt. Die Entwicklungsumgebung von Google Chrome Apps ist identisch zu der von HTML und JavaScript Anwendungen. Da Google Chrome Apps teilweise aus HTML und JavaScript Anwendungen bestehen überschneidet sich die Abdeckung der Kriterien aufgrund der gemeinsam genutzten Bibliotheken und der gleichbleibenden Ausführungsform [21].

5.4.1 WEBSERVICE UNTERSTÜTZUNG

Webservices können in Anwendungen, die mittels JavaScript entwickelt werden, durch die externe Bibliothek Ajax genutzt werden. Sie basiert selbst auf JavaScript und unterstützt jegliche Form von Webservices, welche auf eine Anbindung über das Protokoll HTTP setzen. Hierdurch wird der Evergreen-Webservice in der Version 1.0 [4] vollständig unterstützt, was dieses Kriterium erfüllt.

5.4.2 DIAGRAMMDARSTELLUNG

Diagramme können über mehrere JavaScript Bibliotheken wie Chart.js oder Google Charts erzeugt werden. Es werden alle gängigen Diagrammformen unterstützt wie Liniendiagramme, Balkendiagramme, Kuchendiagramme, Polardiagramme, Flächendiagramme, Netzdiagramme und Doughnut – Diagramme. Dies ist ausreichend um dieses Kriterium zu erfüllen [17] [18].

5.4.3 SMARTPHONE FÄHIG

Für die Ausführung von Google Chrome Apps wird ein installierter Google Chrome Browser, in der Version 30.0 oder höher, benötigt. Dieser ist für Windows Phone, Android und IOS verfügbar, somit ist dieses Kriterium erfüllt.

5.4.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Google Chrome, HTML und JavaScript besitzen Release Versionen. Es liegen zum Zeitpunkt dieser Arbeit keine Informationen vor, dass einer dieser drei Technologien in Zukunft nicht weiter unterstützt werden soll, wodurch dieses Kriterium erfüllt wird.

5.4.5 PERFORMANZ

Es handelt sich bei den genutzten Technologien HTML und JavaScript um Skriptsprachen die durch eine externe Layout-Engine interpretiert werden müssen. Durch diese Interpreter-Schicht, welche sich in der genutzten Google Chrome Installation befindet, sind Google Chrome Apps zwangsläufig langsamer als native Anwendungen und müssen mit weiteren Anwendungen über den Zugang zu diesem Interpreter konkurrieren. Eine Aussage über die garantierte Leistung ist nicht möglich. Sie liegt im direkten Vergleich in anderen Anwendungsgebieten weit hinter nativ entwickelten Anwendungen und daher wird dieses Kriterium nicht erfüllt [19] [20].

5.4.6 MARKTABDECKUNG

Die zwingend notwendige Google Chrome Anwendung, in der Version 30.0, ist für die Smartphone Betriebssysteme Android, Windows Phone sowie Blackberry OS verfügbar. Dies führt zu einer prozentualen Marktabdeckung von 95,4%[9].

5.5 ANDROID APP

Bei einer Android App handelt es sich um eine Anwendung die für das Smartphone-Betriebssystem Android entwickelt wurde. Eine solche Anwendung ist auf Geräten mit den Betriebssystemen Android sowie Blackberry OS, ab der Version Blackberry OS 10, ausführbar. Die Entwicklung selbst wird durch das Android SDK ermöglicht, welches in viele Entwicklungsumgebungen eingebunden werden kann, die sich für die Entwicklung von Java Anwendungen eignen, welches wiederum die Programmiersprache darstellt um Android Apps mit Hilfe des Android SDK zu entwickeln. Die verbreitetsten Entwicklungsumgebungen sind Eclipse und Intelij Idea. Sie bieten Werkzeuge für Syntaxhervorhebung, Quellcode Refactoring, Kontextsensitive Hilfe, Debuggen, Simulation der zu entwickelnden graphischen Be-

nutzeroberfläche sowie einen Emulator, welcher die Simulation eines Android Betriebssystems übernimmt. Diese Entwicklungsumgebungen sind für die Betriebssysteme Windows, Linux und Mac OS verfügbar. Zudem ist es auch möglich über die Android App AIDE Anwendungen auf dem Android Betriebssystem für Android zu erstellen, dabei muss allerdings auf Funktionalitäten wie Debuggen, Quellcode Refactoring, den Emulator für ein extra Android-Betriebssystem und die Simulation der zu entwickelnden graphischen Benutzeroberfläche verzichtet werden [2].

5.5.1 WEBSERVICE UNTERSTÜTZUNG

Webservices können in Anwendungen, die mittels Java geschrieben werden, über automatisch generierte Java Sockets genutzt werden. Hierfür kann das Werkzeug Apache Axis 2 genutzt werden, welches aus einem Webservice funktionsfähigen Java Quellcode generiert. Der generierte Quellcode ermöglicht es jegliche Formen von Webservices zu nutzen. Dadurch ist die Nutzung des Evergreen-Webservices ohne Einschränkungen möglich, somit wird dieses Kriterium erfüllt.

5.5.2 DIAGRAMMDARSTELLUNG

Diagramme können über die Diagramm Komponente AChartEngine erzeugt werden. Es werden alle gängigen Diagrammformen unterstützt wie Liniendiagramme, Balkendiagramme, Kuchendiagramme, Polardiagramme und Flächendiagramme. Dies ist ausreichend um dieses Kriterium zu erfüllen [22].

5.5.3 SMARTPHONE FÄHIG

Da es sich bei Android Apps um Anwendungen für das mobile Betriebssystem Android handelt sind diese auf Smartphones mit diesem Betriebssystem lauffähig, was dieses Kriterium erfüllt.

5.5.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Das Android SDK und die Entwicklungsumgebung Eclipse befinden sich in den stabilen Versionen Android SDK r22.2.1 und Eclipse 4.3.1. Es existieren keine Informationen bezüglich der Abschaffung für diese Plattform somit wird dieses Kriterium erfüllt.

5.5.5 PERFORMANZ

Da es sich um eine native Anwendung auf dem Betriebssystem Android handelt ist eine optimale Leistung zu erwarten, was dieses Kriterium erfüllt.

5.5.6 MARKTABDECKUNG

Android Apps sind auf dem Betriebssystem Android lauffähig, zusätzlich können sie auf dem Betriebssystem Blackberry 10 ausgeführt werden. Somit erreichen Android Apps eine theoretische prozentuale Marktabdeckung von 82,2% [5] [23].

5.6 IOS APP

Bei einer IOS App handelt es sich um eine Anwendung die für das Smartphone Betriebssystem IOS entwickelt wurde. Eine solche Anwendung ist nur auf Geräten mit dem Betriebssystem IOS lauffähig. Die Entwicklung wird durch Apple über die Entwicklungsumgebung Xcode IDE ermöglicht, diese Anwendung ist Max OS exklusiv. Apple stellt eine Sammlung an Werkzeugen für die Entwicklung von IOS Apps zur Verfügung. Sie werden unter dem Paket Xcode Tools zusammengefasst. Das Paket beinhaltet die Entwicklungsumgebung Xcode, das IOS SDK und Werkzeuge für das Entwerfen von Schnittstellen und Patches . Eine IOS App wird in der Xcode Umgebung mittels der Programmiersprache Objective-C programmiert, dabei enthält Xcode Funktionen wie Syntaxhervorhebung, Quellcode Refactoring, Kontextsensitive Hilfe, Debuggen, Werkzeuge für die Simulation der entwickelten graphischen Benutzeroberfläche sowie einen Emulator, welcher die Simulation eines IOS Betriebssystems übernimmt. Um eine IOS App zu entwickeln wird eine Entwicklerlizenz benötigt, welche über eine jährliche 99\$ Gebühr zu erwerben ist [7].

5.6.1 WEBSERVICE UNTERSTÜTZUNG

Die Nutzung von Webservices wird durch das externe Werkzeug EasyWSDL erleichtert. Es generiert den nötigen Quellcode um einen Evergreen-Webservice zu nutzen womit dieses Kriterium erfüllt wird.

5.6.2 DIAGRAMMDARSTELLUNG

Diagramme können über die Diagramm Komponente JBCharView erzeugt werden. Es werden Diagrammformen unterstützt wie Liniendiagramme, Balkendiagramme, Kuchendiagramme und Flächendiagramme. Dies ist ausreichend um dieses Kriterium zu erfüllen [24].

5.6.3 SMARTPHONE FÄHIG

Da es sich bei IOS Apps um Anwendungen für das mobile Betriebssystem IOS handelt sind diese auf Smartphones mit diesem Betriebssystem lauffähig, was dieses Kriterium erfüllt.

5.6.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Das IOS SDK sowie Xcode befinden sich zum Zeitpunkt dieser Arbeit in den stabilen Versionen IOS SDK7.0.6 und Xcode 5.0. Es existieren keine Informationen bezüglich der Abschaffung für diese Plattform, somit wird dieses Kriterium erfüllt.

5.6.5 PERFORMANZ

Da es sich um eine native Anwendung auf dem Betriebssystem IOS handelt, ist eine optimale Leistung zu erwarten, was dieses Kriterium erfüllt.

5.6.6 MARKTABDECKUNG

IOS Apps sind auf Smartphones mit dem Betriebssystem IOS lauffähig und sonst auf keinem anderen Betriebssystem für Smartphones, somit liegt die prozentuale Marktabdeckung gleich mit der des Betriebssystems bei 13,2% [5].

5.7 BLACKBERRY APP

Eine Blackberry App ist eine Anwendung die für das Betriebssystem Blackberry OS entwickelt wurde. Sie sind ausschließlich auf diesem System ausführbar und werden auch als native Blackberry Anwendungen bezeichnet. Blackberry Apps können für die aktuelle Version Blackberry 10.2.1.537 in den Programmiersprachen Java oder auch in C++ entwickelt werden. Die Entwicklung wird von Blackberry, abhängig von der genutzten Programmiersprache, unterstützt. Bei Java handelt es sich um Blackberry JDE und für die C++ Entwicklung wird das Blackberry Nativ SDK bereitgestellt. Die Firma Blackberry stellt als Entwicklungsumgebung eine modifizierte Version von Eclipse zur Verfügung, die speziell für die Entwicklung von Blackberry Apps angepasst wurde. Die angepasste Version von Blackberry trägt den Namen Momentics und bietet wie Eclipse Syntaxhervorhebung, Quellcode Refactoring, Kontext-sensitive Hilfe, Debuggen, Werkzeuge für die Simulation der entwickelten graphischen Benutzeroberfläche. Zusätzlich wurde es mit einem Emulator ergänzt, welcher die Simulation eines Blackberry Betriebssystems übernimmt. In dieser Analyse wird als Programmiersprache Java als Referenz für die Erfüllung der Kriterien genutzt [15].

5.7.1 WEBSERVICE UNTERSTÜTZUNG

Webservices können in Anwendungen, die mittels Java geschrieben werden, über automatisch generierte Java Sockets genutzt werden. Hierfür kann das Werkzeug Apache Axis 2 genutzt werden, welches aus einem Webservice funktionsfähigen Java Quellcode generiert. Der generierte Quellcode lässt es zu jegliche Formen von Webservices zu nutzen, dadurch ist die Nutzung des Evergreen-Webservices ohne Einschränkungen möglich, somit wird dieses Kriterium erfüllt.

5.7.2 DIAGRAMMDARSTELLUNG

Diagramme können über die Diagramm Komponente TeeChart erzeugt werden. Es werden alle gängigen Diagrammformen unterstützt wie Liniendiagramme, Balkendiagramme, Kuchendiagramme, Polar-diagramme, Flächendiagramme und spezielle 2D/3D Diagramme. Dies ist ausreichend um dieses Kriterium zu erfüllen [25].

5.7.3 SMARTPHONE FÄHIG

Da es sich bei Blackberry Apps um Anwendungen für das mobile Betriebssystem Blackberry OS handelt sind diese auf Smartphones mit diesem Betriebssystem lauffähig, was dieses Kriterium erfüllt.

5.7.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Die Entwicklungsumgebung Momentics sowie das unterstützende Blackberry JDE befinden sich in den stabilen Versionen Momentics Tool Suite 6.5 und Blackberry JDE 7.1. Es existieren zum Zeitpunkt dieser Arbeit keine Informationen bezüglich der Abschaffung für diese Plattform, somit wird dieses Kriterium erfüllt.

5.7.5 PERFORMANZ

Da es sich um eine native Anwendung auf dem Betriebssystem Blackberry OS handelt ist eine optimale Leistung zu erwarten, was dieses Kriterium erfüllt.

5.7.6 MARKTABDECKUNG

Blackberry Apps sind auf Smartphones mit dem Betriebssystem Blackberry OS lauffähig und sonst auf keinem anderem Betriebssystem für Smartphones, somit liegt die prozentuale Marktabdeckung gleich mit der des Betriebssystems Blackberry OS bei 2,9% [5].

5.8 WINDOWS PHONE APP

Als Windows Phone App wird eine Anwendung bezeichnet welche für das Smartphone Betriebssystem Windows Phone entwickelt wurde. Eine solche Anwendung ist nur auf dem Betriebssystem Windows Phone ausführbar und wird auch als native Windows Phone Anwendung bezeichnet. Die Entwicklung von Windows Phone Apps wird durch den Hersteller Microsoft mittels des Windows Phone SDK unterstützt. Es handelt sich hierbei um ein Paket mit mehreren Werkzeugen, welche die Entwicklung ermöglichen und erleichtern soll. Windows Phone SDK enthält die Entwicklungsumgebung Visual Studio und das Werkzeug Microsoft Expression, das für die erleichterte Anpassung von graphischen Benutzeroberflächen ausgelegt ist. Die Entwicklungsumgebung Visual Studio ist für das Betriebssystem Windows verfügbar und bietet für die Entwicklung einer Windows Phone App, in den Programmiersprachen C/C++, C# und VB.NET, gängige Funktionalitäten wie Syntaxhervorhebung, Quellcode Refactoring, Kontextsensitive Hilfe, Debuggen, Werkzeuge für die Simulation der entwickelten graphischen Benutzeroberfläche. Zusätzlich enthält Visual Studio einen Emulator, welcher die Simulation eines Windows Phone Betriebssystems übernimmt [13].

5.8.1 WEBSERVICE UNTERSTÜTZUNG

Webservices können in der Entwicklungsumgebung Visual Studio als Service Referenz eingebunden werden. Die Entwicklungsumgebung generiert den benötigten Quellcode automatisch und macht diesen für die Windows Phone Anwendung verfügbar. Es wird hierbei der gesamte Funktionsumfang des Evergreen-Webservice unterstützt, womit dieses Kriterium erfüllt wird.

5.8.2 DIAGRAMMDARSTELLUNG

Diagramme können über die Diagramm Komponente AMCharts erzeugt werden. Es werden Diagrammformen unterstützt wie Liniendiagramme, Balkendiagramme und Flächendiagramme. Dies ist ausreichend um dieses Kriterium zu erfüllen [26].

5.8.3 SMARTPHONE FÄHIG

Da es sich bei Windows Phone Apps um Anwendungen für das mobile Betriebssystem Windows Phone handelt sind diese auf Smartphones mit diesem Betriebssystem lauffähig, was dieses Kriterium erfüllt.

5.8.4 STABILE VERSION DER PLATTFORM IST VERFÜGBAR

Das Windows Phone SDK befindet sich zum Zeitpunkt dieser Arbeit in der stabilen Version Windows Phone SDK 8.0. Es existieren keine Informationen bezüglich der Abschaffung für diese Plattform, somit wird dieses Kriterium erfüllt.

5.8.5 PERFORMANZ

Da es sich um eine native Anwendung auf dem Betriebssystem Windows Phone handelt ist eine optimale Leistung zu erwarten, was dieses Kriterium erfüllt.

5.8.6 MARKTABDECKUNG

Windows Phone Apps sind auf Smartphones mit dem Betriebssystem Windows Phone lauffähig und sonst auf keinem anderen Betriebssystem für Smartphones, somit liegt die prozentuale Marktabdeckung gleich mit der des Betriebssystems bei 3,7% [5].

5.9 ÜBERSICHT UND RESULTAT

Das Resultat der Analyse ergab, dass alle nativen Entwicklungsplattformen, ausgenommen von Symbian, die nötigen Kriterien für die angestrebte Anwendung erfüllen. So wird aufgrund des Entscheidungskriteriums Marktabdeckung Android mit einem Wert von 82,2 % Marktabdeckung als Plattform gewählt. Im Folgenden wird innerhalb der Tabelle 1 Programmierplattform Analyse Übersicht in einer tabellarischen Darstellung das Gesamtergebnis dargestellt. Die K.O. Kriterien wurden orange eingefärbt und das Entscheidungskriterium Marktabdeckung wurde grau eingefärbt.

Tabelle 1 Programmierplattform Analyse Übersicht

Plattform	Webservice	Diagramme	Smartphone	Stabile Version	Performanz	Marktabdeckung
Qt 5.2	Ja	Ja	Ja	Nein	Ja	92,5%
HTML + JavaScript	Ja	Ja	Ja	Ja	Nein	100%
Google Chrome App	Ja	Ja	Ja	Ja	Nein	95,4%
Android App	Ja	Ja	Ja	Ja	Ja	82,2%
IOS App	Ja	Ja	Ja	Ja	Ja	13,2%
Blackberry App	Ja	Ja	Ja	Ja	Ja	2,9%
Windows Phone App	Ja	Ja	Ja	Ja	Ja	3,7%

6 KONZEPTION

Innerhalb dieses Kapitels werden die Resultate des Entwurfs vorgestellt, welche ausführlich mit dem Kunden besprochen wurden. Es beinhaltet die zusammengetragenen Anwendungsfälle, welche die möglichen Nutzungsszenarien der Anwendung umfassen, sowie den Prototyp für die graphische Benutzeroberfläche.

6.1 ANWENDUNGSFÄLLE

Anwendungsfälle reflektieren eingeplante Interaktionen zwischen Nutzer und System. Diese entstehen innerhalb der Entwurfsphase und basieren auf Informationen des Kunden und reflektieren die Entwicklersicht der Anwendung. Die nötigen Informationen wurden über mehrere Kundengespräche zusammengetragen, in diesen zugleich aktuelle Anwendungsfalldiagramme diskutiert wurden. Das Ziel dieses multiphasigen Prozesses ist die Vorstellung des Entwicklers an die Vorstellung des Kunden anzupassen und zugleich die Anforderungen des Kunden an die dem Entwickler zur Verfügung stehenden Mittel anzugleichen. Im Folgenden werden die extrahierten Anwendungsfälle innerhalb von zwei Abbildungen visualisiert sowie durch eine Beschreibung erklärt. Sie wurden zugunsten der Übersicht in die Kategorien Authentifizierung und Konfiguration eingeteilt.

6.1.1 AUTHENTIFIZIERUNG

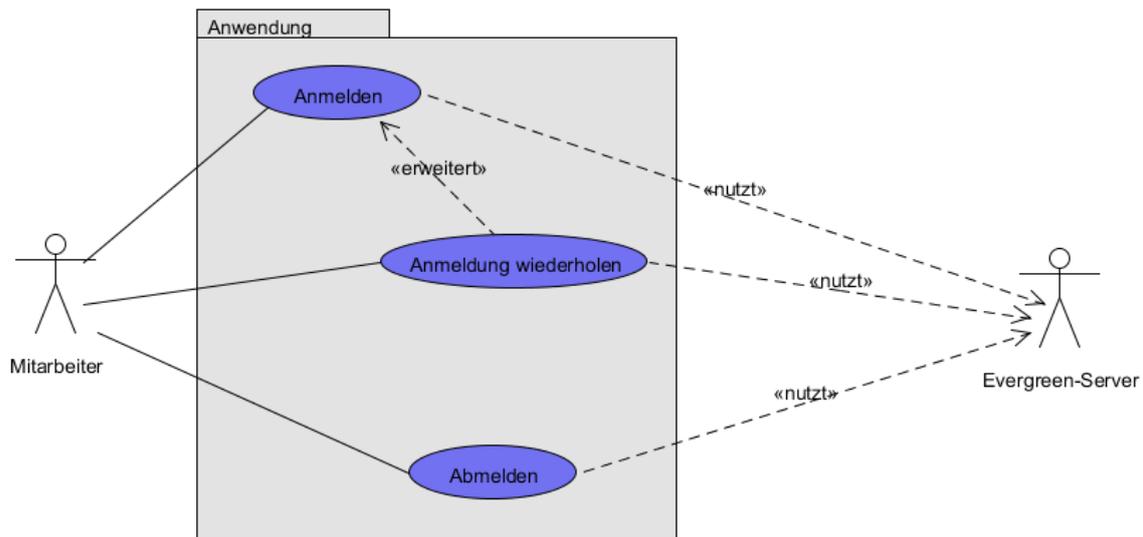


Abbildung 8 Authentifizierungsanwendungsfälle

6.1.1.1 Anmelden

Die Interaktion mit einem Evergreen-Server benötigt eine gültige Session, welche wiederum eine aktive und gültige Anmeldung widerspiegelt und aus einer positiven Anmeldung resultiert. Die Anmeldung benötigt die Evergreen-Server-Adresse, ein gültiges Nutzerkonto sowie das dazugehörige Passwort. Wurde die Anmeldung erfolgreich durchgeführt gelangt der Mitarbeiter in das Hauptfenster. Eine Fehlerhafte Anmeldung wird durch eine Fehlermeldung repräsentiert, welche eine Wiederholung der Aktion ermöglicht oder den gesamten Vorgang abubrechen. Die Anmeldung ist innerhalb des Anmeldefenster möglich, in dieses gelangt der Mitarbeiter bei Anwendungsstart oder wenn er sich zuvor abgemeldet hat.

6.1.1.2 Anmeldung wiederholen

Läuft während der Nutzung die Session aus und wird somit ungültig, ist es nötig die Session zu erneuern oder die Anwendung zu schließen. Dieser Anwendungsfall ermöglicht es dem Mitarbeiter seine Session zu erneuern und die Observierung fortzuführen. Ist er nicht in der Lage mit dem entsprechenden Nutzerkonto sich erneut anzumelden, kann er in das Anmeldefenster wechseln und sich mit einem anderen Nutzerkonto anmelden oder die Anwendung schließen. Eine gültige Neuansmeldung erfordert eine Verbindung zu dem adressierten Evergreen-Server und das gültige Passwort des zuvor angemeldeten Mitarbeiters.

6.1.1.3 Abmelden

Wird eine gültige Session nicht mehr benötigt besitzt der Mitarbeiter die Möglichkeit über den Abmelden Anwendungsfall diese zu schließen. Meldet sich ein Mitarbeiter ab gelangt er vom Hauptfenster in das Anmeldefenster von dem aus er die Anwendung schließen oder sich erneut anmelden kann. Die Abmeldung benötigt eine Verbindung zu einem Evergreen-Server und eine gültige Session.

6.1.2 KONFIGURATION

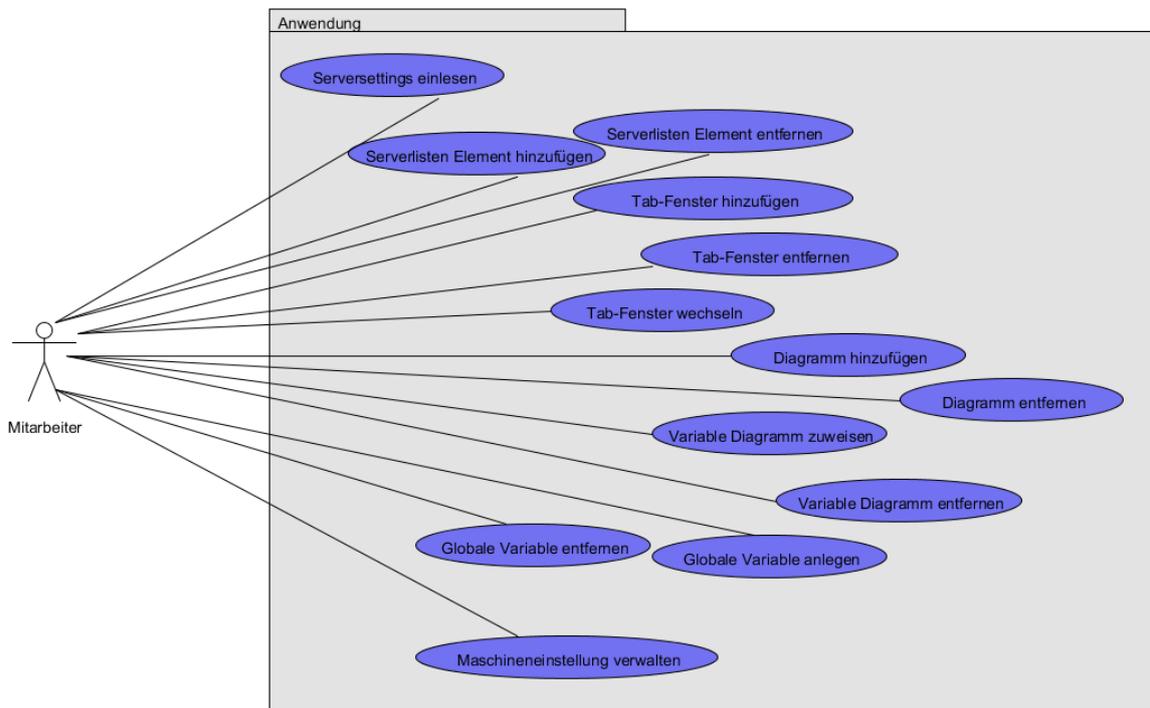


Abbildung 9 Konfigurationsanwendungsfälle

6.1.2.1 Serverlisten Element hinzufügen

Die Anmeldung benötigt die Adresse eines Evergreen-Servers. Diese Information ist innerhalb von Serverlistenelementen gespeichert. Diese Elemente befinden sich innerhalb einer globalen Serverliste. Ein Mitarbeiter ist innerhalb des Anmeldefensters in der Lage ein neues Serverlisten Element hinzuzufügen. Er benötigt hierfür dessen Adresse, Port und einen Namen zur Identifikation.

6.1.2.2 Serverlisten Element entfernen

Befindet sich innerhalb der globalen Serverliste ein Element, welches Fehler aufweist oder nicht mehr benötigt wird, ist der Mitarbeiter in der Lage dieses aus der globalen Serverliste zu entfernen. Diese Funktion steht innerhalb des Anmeldefensters zur Verfügung.

6.1.2.3 Serversettings einlesen

Hierbei handelt es sich um eine Zusatzfunktion, welche eine Konfigurationsdatei einliest, um gebündelte Konfigurationen von Serverlisten Elementen einzulesen und zu übernehmen. Der Hauptvorteil liegt innerhalb der mehrfachen Konfiguration von Anwendungen, um diese auf einen Ausgangszustand zurückzusetzen oder auf mehreren Endgeräten zu konfigurieren. Das Einlesen der Konfiguration ist über eine Textdatei oder NFC möglich.

6.1.2.4 Tab-Fenster hinzufügen

Tab-Fenster dienen der Gruppierung von Diagrammen und erzielen eine verbesserte Übersicht über die Anwendung und deren konfigurierten Diagramme. Der Mitarbeiter ist über diesen Anwendungsfall in der Lage ein neues Tab-Fenster und somit eine neue Gruppierung für Diagramme zu erstellen. Die Erstellung eines neuen Tab-Fensters ist innerhalb des Hauptfensters möglich und benötigt einen eindeutigen Namen.

6.1.2.5 Tab-Fenster entfernen

Wird ein Tab-Fenster nicht mehr benötigt, kann es über diesen Anwendungsfall entfernt werden. Diese Funktion steht innerhalb des Hauptfensters zu Verfügung und benötigt ein existierendes Tab-Fenster, welches entfernt werden soll.

6.1.2.6 Tab-Fenster wechseln

Tab-Fenster wechseln stellt eine Navigationsfunktion dar, welche für das navigieren zwischen mehreren Tab-Fenstern benötigt wird. Navigiert der Mitarbeiter zu einem Tab-Fenster ersetzt es das aktuelle Tab-Fenster innerhalb der Hauptanzeige, somit werden die darin konfigurierten Diagramme innerhalb des Hauptfensters angezeigt. Für diesen Anwendungsfall werden 2 oder mehrere Tab-Fenster benötigt, um innerhalb des Hauptfensters zwischen diesen zu navigieren.

6.1.2.7 Diagramm hinzufügen

Die Visualisierung von Werten benötigt Diagramme, diese stellen die aktuellen oder vergangenen Werte dar. Die Darstellung selbst variiert abhängig von dem gewählten Diagrammtyp. Es existieren textuelle Diagramme, Tabellen Diagramme, Balkendiagramme und Liniendiagramme. Ein Diagramm kann innerhalb des Hauptfensters hinzugefügt werden, während dieses Prozesses wird der Diagrammtyp festgelegt und die Gruppierung in Form eines Tab-Fensters zugeordnet, dies erfolgt durch den Mitarbeiter. Diese Funktion benötigt ein existierendes Tab-Fenster, um diesem ein Diagramm zuzuordnen.

6.1.2.8 Diagramm entfernen

Wird ein Diagramm nicht mehr benötigt ist der Mitarbeiter in der Lage durch diesen Anwendungsfall dieses zu entfernen. Diese Funktion steht innerhalb des Hauptfensters zur Verfügung und erfordert ein existierendes Diagramm innerhalb eines Tab-Fensters, welches entfernt werden soll.

6.1.2.9 Variable Diagramm zuweisen

Diagramme benötigen für die Visualisierung Variablen, welche diesen zugewiesen werden muss. Der Mitarbeiter ist in der Lage innerhalb des Hauptfensters diese Funktion zu nutzen und Diagrammen globale Variablen eines Serverlisten Elements zuzuweisen. Für diese Funktion wird ein Tab-Fenster mit mindestens einem Diagramm sowie einer globalen Variable benötigt, welche selbst noch nicht dem Ziel Diagramm zugewiesen wurde. Eine globale Variable kann einem Diagramm einfach zugewiesen werden, eine mehrfache Zuweisung ist nicht möglich.

6.1.2.10 Variable Diagramm entfernen

Wird eine Variable innerhalb eines Diagrammes nicht mehr benötigt, ist der Mitarbeiter mittels dieses Anwendungsfalles in der Lage diese aus einem entsprechenden Diagramm zu entfernen. Die Funktion steht innerhalb des Hauptfensters zur Verfügung und benötigt ein Tab-Fenster, welches ein Diagramm mit einer zugewiesenen globalen Variable beinhaltet.

6.1.2.11 Globale Variable anlegen

Globale Variablen repräsentieren einen Wert auf der Fertigungsmaschine. Sie können innerhalb der Anwendung Diagrammen zu gewiesen werden, welche deren Werte visualisiert. Der Mitarbeiter kann innerhalb des Hauptfensters neue globale Variablen anlegen, benötigt aber ein tieferes technisches Verständnis als bei allen anderen Anwendungsfällen. Er benötigt die abstrakte Adresse der Variablen, welche die ID darstellt, sowie dessen Typ, um eine korrekte Verwendung innerhalb der Anwendung zu gewährleisten.

6.1.2.12 Globale Variable entfernen

Wird eine globale Variable nicht mehr benötigt kann der Mitarbeiter diese innerhalb des Hauptfensters entfernen. Er benötigt hierfür eine existierende globale Variable, welche keinem Diagramm zugeteilt wurde.

6.1.2.13 Maschineneinstellungen verwalten

Innerhalb dieses Anwendungsfalles ist der Mitarbeiter in der Lage die Aktualisierungsrate einzustellen, in welcher die globalen Variablen durch eine Serveranfrage an den Evergreen-Server aktualisiert werden. Diese Funktion ist innerhalb des Hauptfensters zu erreichen.

6.2 PROTOTYP

Ein Prototyp dient als Vorabversion um das optische Erscheinungsbild und Funktionen einer Anwendung vorzustellen. Für diese Anwendung entschied man sich für einen horizontalen Prototypen, welcher keine Funktionalität besitzt und rein das optische Erscheinungsbild darstellt. Er wurde genutzt um zusammen mit dem Kunden Funktionen und Verhaltensweisen der Anwendung zu diskutieren, sowie Missstände auf Seiten des Entwicklers aufzudecken. Bei dem genutzten Prototyp handelt es sich um einen Paperprototype. Im Folgenden wird dieser Prototyp in seinem Endstadium erläutert. Es handelt sich um Ansichten einzelner geplanter Zustände der Anwendung, welche über einen Titel der Ansicht, den dazugehörigen Prototypen sowie einer Beschreibung verfügen.

6.2.1 ANMELDEFENSTER

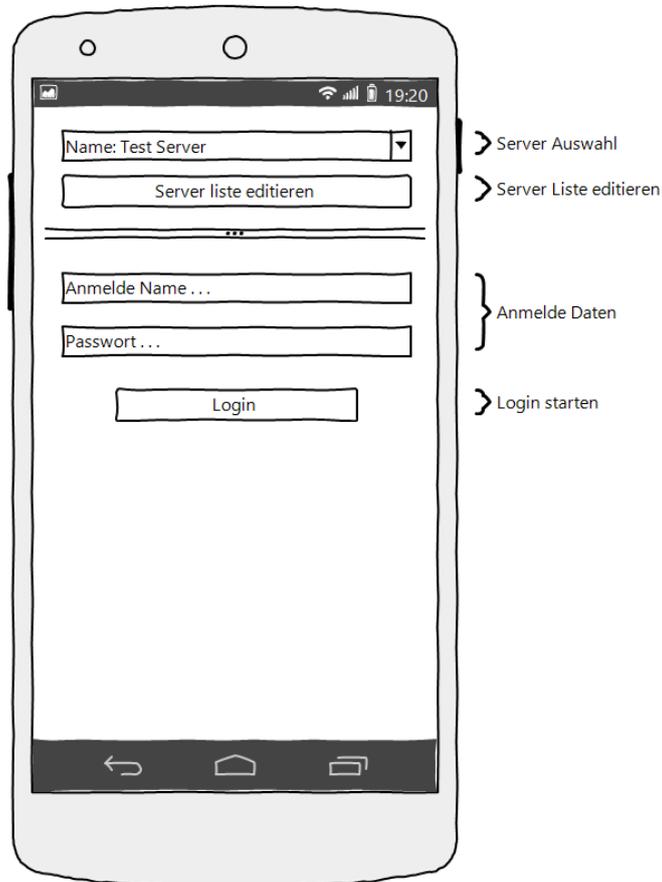


Abbildung 10 Anmeldefenster

Der Prototyp innerhalb der Abbildung 10 Anmeldefenster repräsentiert das Startfenster der Anwendung. Es ermöglicht die Serverauswahl und das Login zudem enthält es die Schaltflächen für den Serverliste editieren Dialog. Die Ansicht besteht aus der Serverauswahl, welche über ein Dropdown Element realisiert wurde. Darunter befindet sich die Schalfläche für den Serverliste editieren Dialog in welchem die Server Auswahl angepasst werden kann. Zwei Eingabefelder ermöglichen die Eingabe des Nutzerkontos sowie das dazu gehörige Passwort, dieses wird nicht im Klartext angezeigt. Eine Schaltfläche für den Login Start nutzt die Daten des ausgewählten Servers sowie den Inhalt der Anmelde-Daten um einen Anmelde-Versuch zu starten. Befinden sich die Server- und Anmelde-Daten in einem validen Format und sind zudem korrekt erfolgt eine positive Anmeldung, aus welcher eine gültige Session resultiert, zusätzlich wird das Anmeldefenster geschlossen und der Nutzer in das Hauptfenster navigiert.

6.2.2 SERVERLISTE EDITIEREN DIALOG

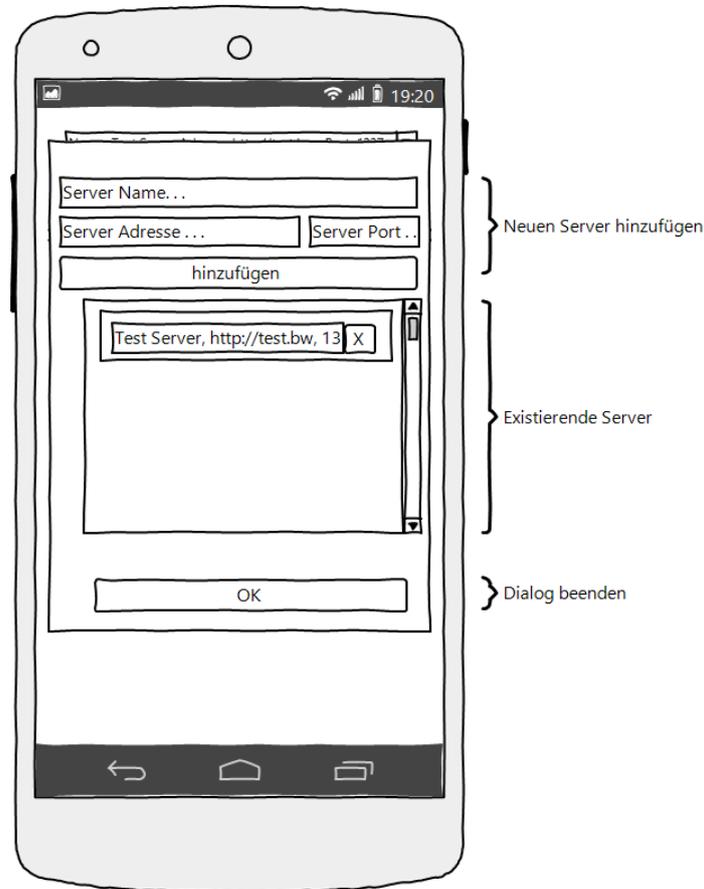


Abbildung 11 Serverliste editieren Dialog

Der Serverliste editieren Dialog bietet die Funktionalität die global verfügbare Serverliste zu manipulieren. Innerhalb des oberen Bereichs findet sich die neue Server Hinzufügen-Funktion, diese besteht aus drei Eingabefeldern, welche zur Eingabe der nötigen Informationen Servername, Serveradresse und Server-Port auffordern und einer Hinzufügen-Schaltfläche, welche bei ihrer Aktivierung die Eingabe validiert und aus einer validen Eingabe ein neues Serverlisten Element der Liste hinzufügt. Gefolgt wird dies von einer Liste der existierenden Server, sie bildet die Serverliste ab. Jedes Serverlistenelement wird als einzelnes Listenelement dargestellt mit seinen Informationen Servername, Serveradresse und Server-Port als Inhalt und einer Entfernen-Schaltfläche, welche bei ihrer Aktivierung das jeweilige Serverlistenelement aus der Serverliste entfernt. Das Ende dieses Dialoges bildet eine OK-Schaltfläche dessen Aktivierung den Dialog beendet. Die Änderungen, welche während dieses Dialoges vorgenommen werden, sind persistent und müssen nicht über eine weitere Schaltfläche bestätigt werden. Dieser Dialog ist aus dem Anmeldefenster über die Schaltfläche Serverliste editieren aufrufbar.

6.2.3 HAUPTFENSTER

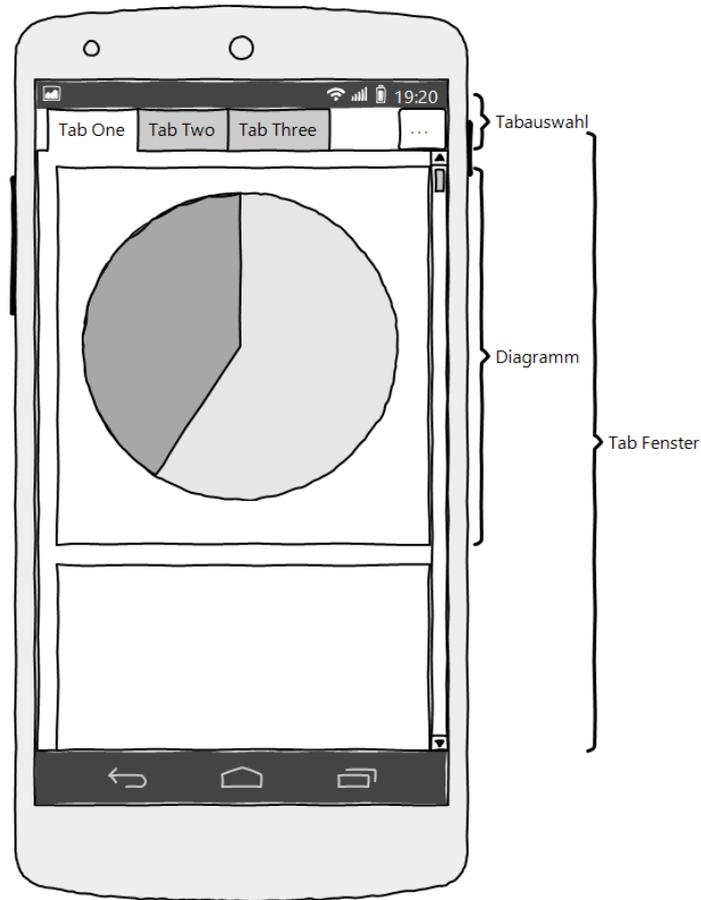


Abbildung 12 Hauptfenster

Wurde eine Anmeldung erfolgreich absolviert gelangt der Mitarbeiter in das in Abbildung 12 Hauptfenster abgebildete Hauptfenster. Dieses Fenster bildet den zentralen Punkt der Anwendung. Jegliche Konfiguration und jegliche Observierung von Variablen wird innerhalb dieses Fensters vorgenommen oder wird innerhalb dieses Fensters begonnen. Es besitzt am oberen Rand die Tab-Auswahl, welche das darunterliegende Tab-Fenster mit der Auswahl ersetzt. Das Tab-Fenster nimmt den restlichen Anzeigebereich ein und beinhaltet die konfigurierten Diagramme für die Visualisierung der Maschinendaten.

6.2.4 OPTIONEN

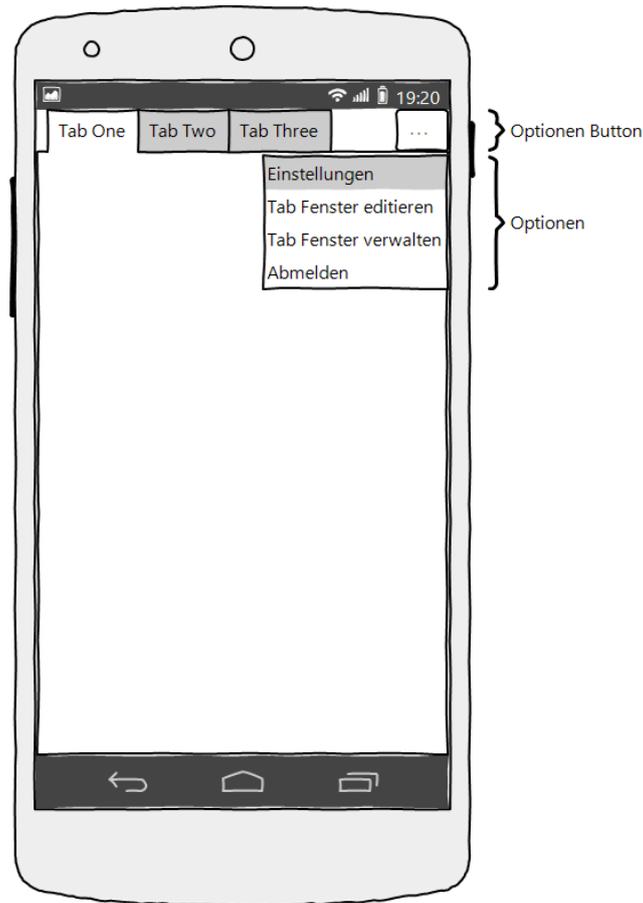


Abbildung 13 Optionen

Die Optionen sind innerhalb des Hauptfensters über die Schaltfläche der Optionen Button zu erreichen. Wird diese Schaltfläche betätigt erscheinen die verfügbaren Optionen. Es sind die Optionen Einstellungen, Tab-Fenster editieren, Tab-Fenster verwalten und Abmelden verfügbar. Diese Optionen leiten bei ihrer Auswahl einen Dialog ein, welcher die nötigen Konfigurationsmöglichkeiten bietet.

6.2.5 TAB-FENSTER VERWALTEN DIALOG

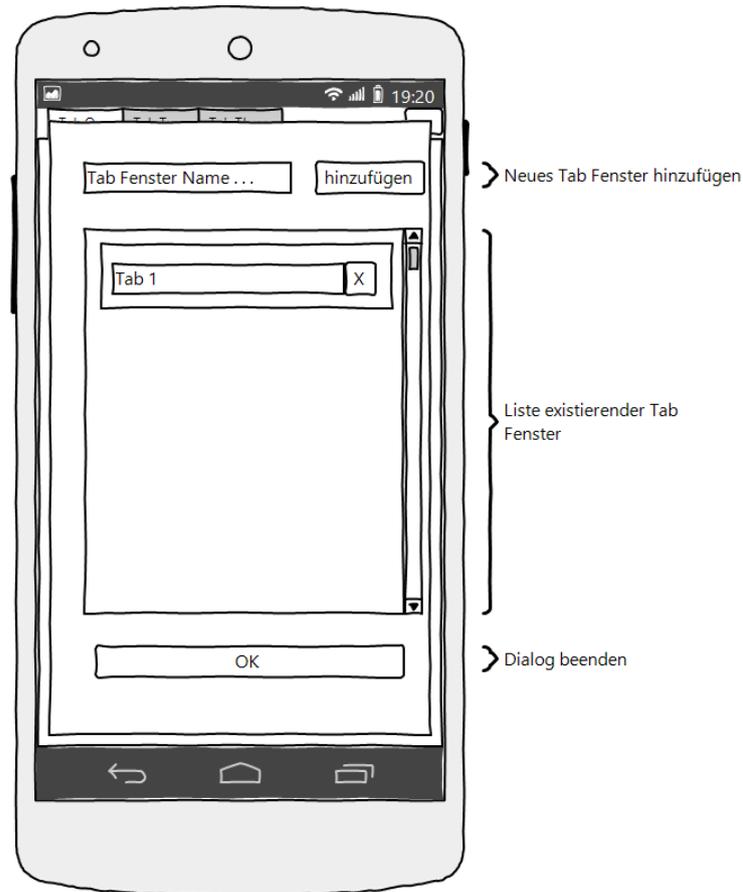


Abbildung 14 Tab-Fenster verwalten

Der Tab-Fenster verwalten Dialog dient der Organisation der Tab-Fenster. Er bietet im oberen Bereich die neue Tab-Fenster Hinzufügen-Funktion, diese besteht aus einem Eingabefeld für den Tab-Fenster-namen sowie einer Hinzufügen-Schaltfläche welche ein neues Tab-Fenster der Liste hinzufügt, welches den eingegebenen Namen trägt. Darunter befindet sich die Liste existierender Tab-Fenster, diese zeigt einzelne Tab-Fenster als Listenelement an mit ihrem Namen und einer Schaltfläche um diese aus der Liste zu entfernen. Am unteren Ende des Dialoges befindet sich eine OK-Schaltfläche mit dieser kann der Dialog geschlossen werden. Dieser Dialog wird aus den Optionen innerhalb des Hauptfensters mit der Option Tab-Fenster verwalten aufgerufen.

6.2.6 TAB-FENSTER EDITIEREN DIALOG

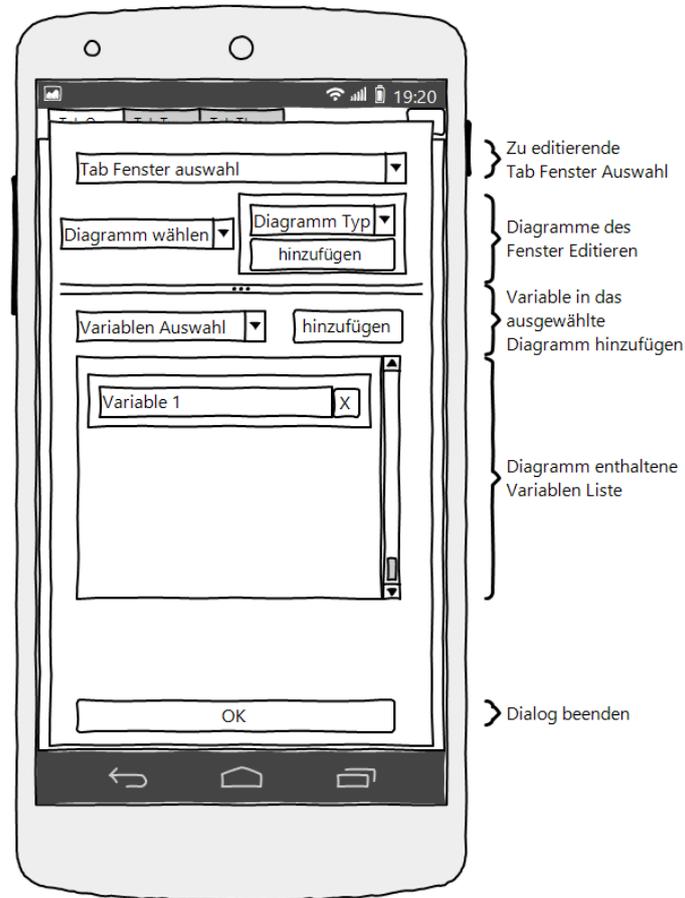


Abbildung 15 Tab-Fenster editieren

Der Tab-Fenster editieren Dialog aus Abbildung 15 Tab-Fenster editieren bietet Funktionalität um den Inhalt eines Tab-Fensters zu gestalten, in dem der Mitarbeiter die nötigen Diagramme einfügt und diesen globale Variablen zuweist. Dabei wählt der Mitarbeiter zunächst innerhalb des oberen Teils des Dialoges das gewünschte Tab-Fenster aus, welches er editieren möchte. Darunter befinden sich die Funktionen um ein existierendes Diagramm auszuwählen oder ein weiteres Diagramm dem Tab-Fenster hinzuzufügen. Wählt der Mitarbeiter ein Diagramm aus wird darunter die Option eingeblendet globale Variablen in das ausgewählte Diagramm einzufügen oder aus der im Diagramm enthaltenen Variablenliste zu entfernen. Dieser Dialog wird aus den Optionen innerhalb des Hauptfensters mit der Option Tab-Fenster editieren aufgerufen.

6.2.8 EINSTELLUNGEN DIALOG

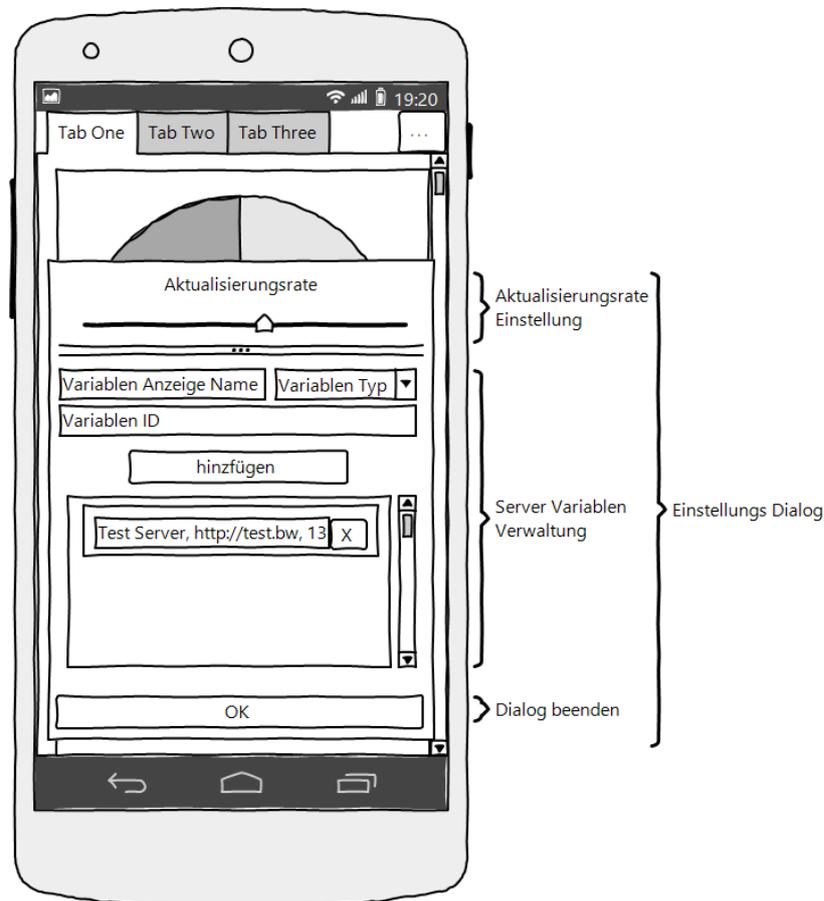


Abbildung 16 Einstellungen

Der Einstellungsdialog aus Abbildung 16 Einstellungen organisiert die zur Verfügung stehenden globalen Variablen und die Aktualisierungsrate, welche angibt in welchen Zeitabständen Anfragen an den Server erfolgen. Der Dialog selbst besteht am oberen Teil aus der Aktualisierungseinstellung gefolgt von der Variablen Verwalten-Funktion, diese besteht aus der Variablen Hinzufügen-Funktion und einer Variablenliste. Die Variablen Hinzufügen-Funktion ermöglicht die Eingabe eines individuellen Namens, die Adresse der Variablen und deren Typ um diese der Liste hinzuzufügen. Über die Variablenliste können alle existierenden globalen Variablen eingesehen werden, diese werden als jeweils einzelnes Listenelement dargestellt und beinhalten die Information Variablen-Name, Variablen-Adresse und Variablen-Typ sowie eine Entfernen-Schaltfläche deren Aktivierung die jeweilige Variable aus der globalen Liste entfernt. Dieser Dialog wird aus den Optionen innerhalb des Hauptfensters mit der Option Einstellungen aufgerufen.

6.2.9 ANMELDUNG WIEDERHOLEN DIALOG

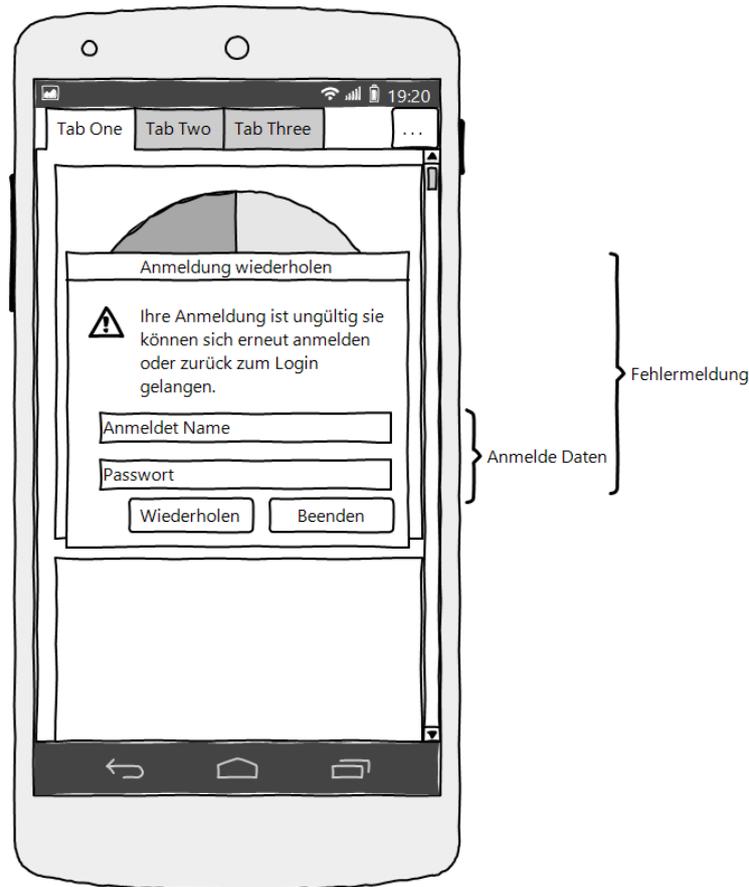


Abbildung 17 Anmeldung wiederholen

Läuft eine Session während der Anwendungsverwendung aus wird der Anmeldung wiederholen-Dialog automatisch aufgerufen. Dieser fordert den Mitarbeiter auf seine Anmeldung zu wiederholen und hierdurch eine neue gültige Session zu erlangen. Für eine positive Anmeldung werden die korrekten Anmeldedaten benötigt, welche ein existierendes Nutzerkonto mit dem dazugehörigen Passwort umfasst. Ist der Mitarbeiter nicht in der Lage eine positive Anmeldung zu erwirken kann er über diesen Dialog zurück zum Anmeldefenster navigieren. Dieser Dialog ist nicht von einem Nutzer aufrufbar und kann nur von dem System aufgerufen werden, er bewirkt ein Anhalten der Aktualisierung, da diese induktiv keine weiteren positiven Anfragen generieren kann.

7 SYSTEMMODELL

Das Systemmodell beschreibt den Aufbau der Anwendung in Form eines Komponentendiagrammes, welches einen Überblick über die bestehenden Komponenten der Anwendung beinhaltet und einem Klassendiagramm, welches einen detaillierteren Einblick in den Aufbau der jeweiligen Komponente ermöglicht und ihre Abhängigkeiten beschreibt. Als Entwurfsmuster wurde das MVC-Pattern [1] genutzt um größere Abhängigkeiten zwischen Komponenten zu vermeiden.

7.1 KOMPONENTEN

Innerhalb dieses Kapitels werden die Komponenten der entwickelten Anwendung beschrieben. Eine Komponente beschreibt eine abstrahierte Ebene der Anwendung, welche Funktionselemente als Komponente zusammenfasst. Es werden im Folgenden die innerhalb der Abbildung 18 Komponentendiagramm abgebildeten Komponenten beschrieben und deren Abhängigkeiten zueinander erklärt, hierfür werden Funktionalitäten der einzelnen Klassen der Komponente erläutert. Dieser detaillierterer Einblick wird innerhalb der Abbildung 20 System Klassendiagramm visualisiert.

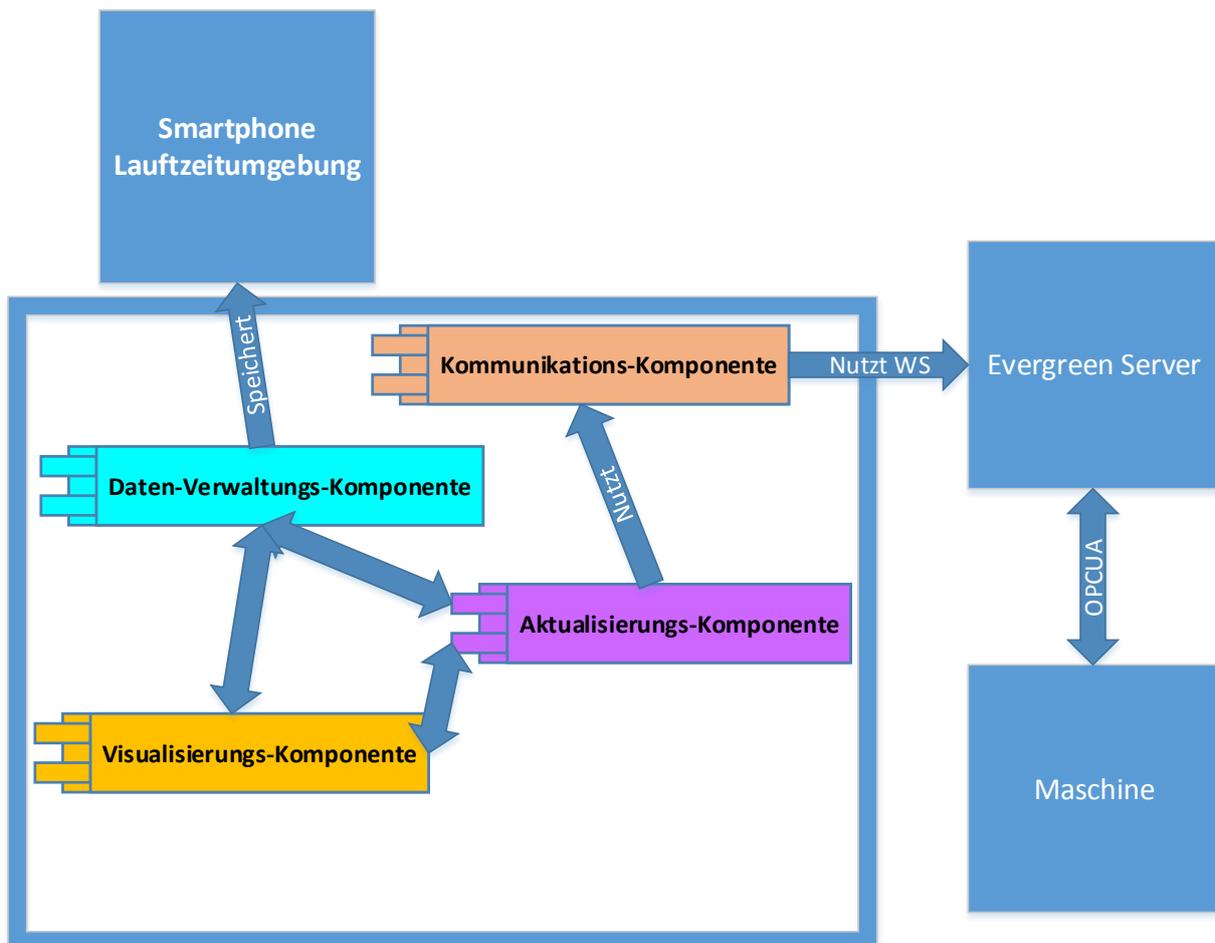


Abbildung 18 Komponentendiagramm

7.1.1 SMARTPHONE LAUFZEIT UMGEBUNG

Die Smartphone Laufzeit Umgebung repräsentiert eine externe Komponente deren relevante Funktionalität für diese Anwendung sich darauf bezieht Daten aus der Datenverwaltungskomponente persistent zu speichern.

7.1.2 MASCHINE

Es handelt sich hierbei um die abstrakte zu observierende Fertigungsmaschine, welche über ein Kommunikationsprotokoll mit dem Evergreen Server verbunden ist und Daten für die mögliche Observierung bereitstellt.

7.1.3 EVERGREEN SERVER

Der Evergreen-Server dient als Abstraktionskomponente für die über das Kommunikationsprotoll verbundene Maschine. Er bietet die Möglichkeit über den Evergreen-Webservice eine Authentifizierung vorzunehmen und die abstrahierten Maschinendaten auszulesen. Diese externe Komponente wird für die Authentifizierung sowie das Auslesen von Daten von der Kommunikationskomponente genutzt.

7.1.4 KOMMUNIKATIONSKOMPONENTE

Die Kommunikationskomponente verwaltet direkte Interaktionen mit dem Evergreen-Server, hierfür nutzt sie dessen bereitgestellten Evergreen-Webservice. Sie selbst besitzt keine weitere Verarbeitungslogik. Diese Komponente wird von der Aktualisierungskomponente genutzt um Anfragen an den Evergreen-Server zu stellen.

Der Aufbau dieser Komponente wurde automatisch aus der XML-Deklaration der Evergreen-Webservice Schnittstelle generiert. Sie besteht aus einer Funktionsklasse der **EvergreenBinding**-Klasse, welche alle Interaktionen mit dem Evergreen-Server ermöglicht. Die genutzten Funktionen innerhalb dieser Anwendung beschränken sich auf das Lesen von Variablen, Anmeldung und das Abmelden.

Die Anmeldung erfordert die Informationen eines Nutzerkontos sowie das dazugehörige Passwort und erstellt hieraus ein **LoginInformationType**-Objekt mittels dessen eine Anmeldeanfrage an den Evergreen-Server gestellt wird. Aus einer positiven Anmeldung resultiert ein **AuthenticationType**-Objekt, welches eine gültige Session oder eine Fehlermeldung beinhaltet.

Um eine Leseoperation auszuführen wird ein **AuthenticationType**-Objekt benötigt, welches eine gültige Session beinhaltet sowie ein **VarIdSet**-Objekt, welches die Adressen derer Variablen beinhaltet die von dem Evergreen-Server gelesen werden sollen. Das Resultat ist ein **VarValueResponseSet**-Objekt, welches die gelesenen Werte der angeforderten Variable beinhaltet oder eine jeweilige Fehlermeldung, welche aufgetretene Fehler widerspiegelt.

7.1.5 AKTUALISIERUNGSKOMPONENTE

Die Aktualisierungskomponente wird von der Visualisierungskomponente genutzt. Sie organisiert die Aktualisierung der Daten, bereitet die bestehende Konfiguration als Anfrage für die Kommunikationskomponente auf und verarbeitet die Session für eine gültige Authentifizierung. Sie stellt ein Interface bereit um Aktualisierungen an Datenobjekte weiterzuleiten.

Diese Komponente organisiert über Instanzen der **BackgroundRequester**-Klasse die Aktualisierungen. Die Einstellung der Aktualisierungsrate erfolgt dynamisch zur Laufzeit. Datenobjekte, welche das **DiagramDataObserverInterface** Interface implementieren, abonnieren Aktualisierungen bezüglich Variablen und erhalten diese über die Updatefunktion des Interfaces in Form einer **DataSetMessage** Nachricht, welche die neuen Werte für einzelne sowie Listen-Variablen beinhaltet. Für die Aktualisierungsanfragen wird die Kommunikationskomponente genutzt und die nötigen Authentifizierungsinformationen werden von der Instanz der **AuthenticationManager**-Klasse bereitgestellt.

Die **AuthenticationManager**-Klasse wurde nach dem Singleton-Pattern entworfen, dies hat zur Folge, dass während der Laufzeit maximal eine Instanz der Klasse existiert, somit kann gewährleistet werden, dass keine Komplikationen auftreten oder parallel mehrere Anmeldungen aktiv sind. Die **AuthenticationManager**-Instanz organisiert die gültigen Sessions und bereitet die Daten für das Anmelden, Abmelden auf und bietet hierfür Funktionen als Schnittstelle der Komponente an.

7.1.6 DATENVERWALTUNGSKOMPONENTE

Die Datenverwaltungskomponente realisiert die persistente Datenspeicherung von Serverkonfigurationen und nutzt hierfür die Smartphone-Laufzeitumgebung. Es werden Datencontainer bereitgestellt um Werte über die Aktualisierungskomponente zu aktualisieren und ihren Verlauf auf ein dynamisch zur Laufzeit festgelegtes Maximum bereit zu stellen.

Die persistente Datenspeicherung erfolgt als JSON valider Text, dessen Datenmodell innerhalb der Abbildung 19 Datenmodell abgebildet wird. Es besteht aus einem Speichermodul, welches eine Liste an Servermodulen beinhaltet. Ein Servermodul enthält einen eindeutigen Namen, eine Evergreen-Serveradresse, dessen Port, eine Liste von Variablenmodulen und eine Liste von Tab-Fenstermodulen, diese besitzen einen eindeutigen Namen und eine Liste von Diagrammmodulen. Ein Diagrammmodul besitzt einen Typ und eine Liste von zugewiesenen Variablenmodulen, welche als Abstraktion von Maschinendaten realisiert wurden und hierfür einen eindeutigen Namen, eine eindeutige Adresse und einen Typ besitzen. Eine Instanz der **GlobalenAppDataHandler**-Klasse ermöglicht es diese Informationen zu laden und konfigurierte Varianten zu speichern. Ein Servermodul wird als Serverkonfiguration bezeichnet und ein Speichermodul als Anwendungsconfiguration.

Innerhalb von Instanzen der **DiagramDataContainer**-Klasse werden aktuelle Werte sowie Werteverläufe von Variablen verwaltet, welche einem einzelnen Diagramm zugeordnet wurden. Um Aktualisierungen zu erhalten implementiert sie das **DiagramDataObserverInterface** Interface mittels dessen können Instanzen Aktualisierungen zu Variablen bei einer **BackgroundRequester**-Instanz abonnieren und Aktualisierungen in Form einer **DataSetMessage** – Nachricht erhalten.

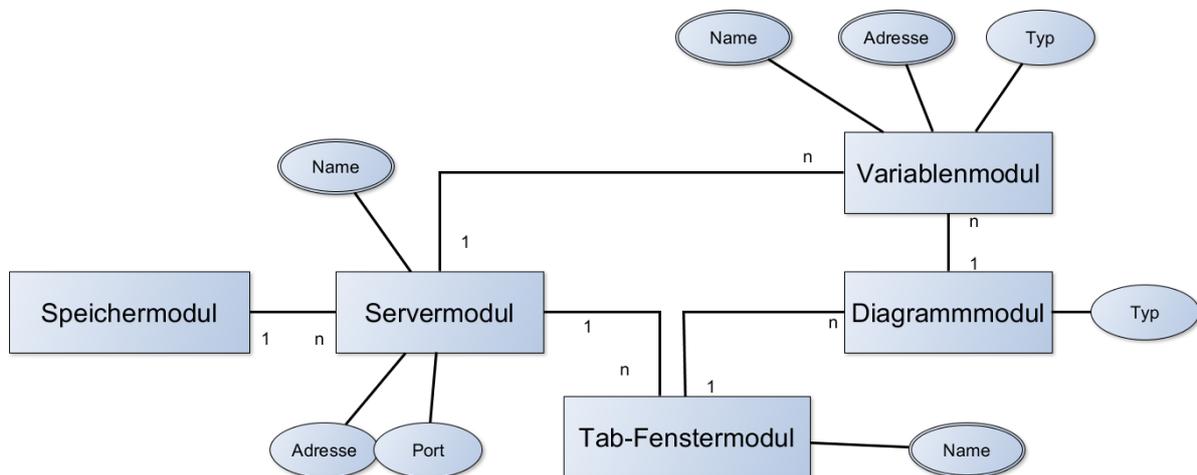


Abbildung 19 Datenmodell

7.1.7 VISUALISIERUNGSKOMPONENTE

Die Visualisierungskomponente übernimmt die Darstellung der graphischen Benutzeroberfläche, koordiniert ihr Verhalten und verarbeitet Nutzereingaben. Es wurde darauf geachtet die gesamte Komponente modular aufzubauen um Erweiterungen wie weitere Diagramme oder eine erweiterte Datenstruktur zu ermöglichen.

Sie besteht aus zwei Activities, welche innerhalb einer Android-Anwendung ein separates Fenster darstellt. Es handelt sich hierbei um die **LoginActivity** und die **MainActivity**.

Die **LoginActivity** stellt das Anmeldefenster (s. Pkt.6.2.1) dar. Es nutzt die Datenverwaltungskomponente um Anwendungskonfigurationen zu laden und dem Anwendungsnutzer zur Verfügung zu stellen, damit dieser einen Server für die Anmeldung auswählen kann. Der Anwendungsnutzer kann diese Anwendungskonfiguration über den Serverliste editieren-Dialog (s. Pkt. 6.2.2) manipulieren, indem er neue Serverlistenelemente der Anwendungskonfiguration hinzufügt oder aus dieser entfernt. Erfolgt eine positive Anmeldung des Nutzers an einem Evergreen-Server startet die **LoginActivity** die **MainActivity** und übergibt ihr den eindeutigen Namen der gewählten Serverkonfiguration. Die **LoginActivity** nutzt für den Anmeldevorgang die Singleton-Instanz der **AuthenticationManager**-Klasse, welche die erworbene gültige Session verwaltet.

Das Hauptfenster (s. Pkt. 6.2.3) wird von der **MainActivity** dargestellt und verwaltet. Diese erhält über einen Intent die gewählte Serverkonfiguration aus dem Anmeldefenster. Aus dieser Serverkonfiguration generiert sie die Anzeige. Hierbei werden die einzelnen Datenobjekte aus der Datenverwaltungskomponente auf graphische Elemente projiziert. Eine Serverkonfiguration stellt das gesamte Hauptfenster dar, welches innerhalb der oberen Tab Leiste (s. Pkt. 6.2.3) ein Navigationselement besitzt, um über die existierenden Tab-Fenster zu navigieren, welche aus der Serverkonfiguration extrahiert werden. Diese Tab-Fenster enthalten die zur Visualisierung genutzten Diagramme. Diagramme werden über das **DiagramViewInterface**-Interface abstrahiert. Hierüber werden die graphischen Elemente nach außen für die Activity Verfügbar gemacht und Aktualisierungen diesbezüglich können von außen eingespeist werden, um die Darstellung zu aktualisieren. Es werden die geforderten Diagrammformen (s. Pkt. 3) über die Klassen **BlockChart**, **LineChart**, **TableDiagram** und **TextDiagram** realisiert. Die benötigten Dialoge für die Konfiguration der Anwendungskonfiguration sowie der Serverkonfiguration werden innerhalb der **DialogCreator**-Klasse erzeugt, welche nach dem Factory-Pattern entworfen wurde.

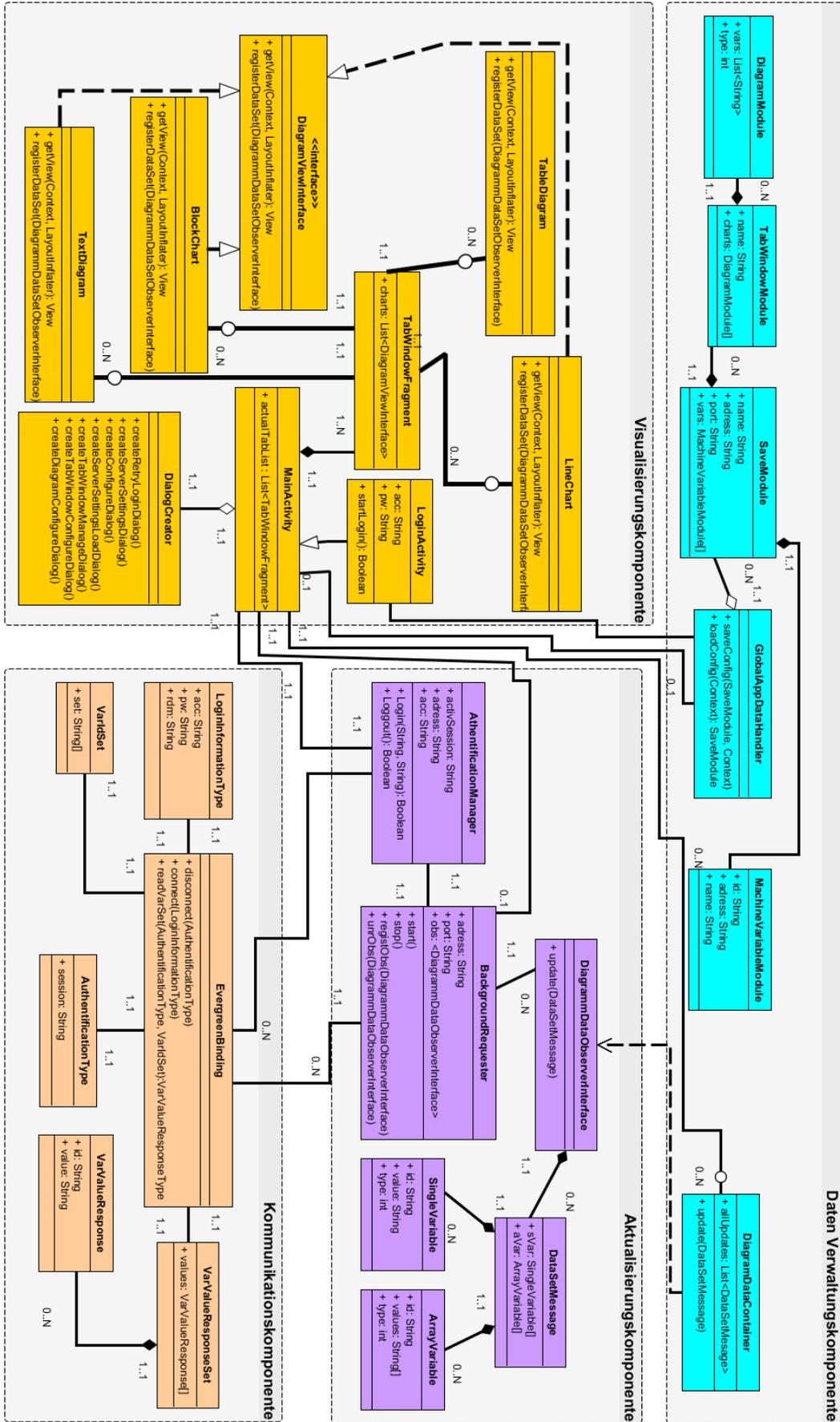


Abbildung 20 System Klassendiagramm

8 IMPLEMENTIERUNG

8.1 ENTWICKLUNGSPROZESS

Der Entwicklungsprozess ist ein Kernpunkt der Entwicklung, er gibt die Reihenfolge der Aktionen innerhalb eines Prozesses vor und wie auf Ausnahmen reagiert werden muss.

Innerhalb dieser Entwicklung wurde ein agiler Entwicklungsprozess gewählt, da dieser den Kunden eines Produktes stärker einbezieht. Hierfür wurden mit dem Kunden Meilensteine vereinbart, welche eine zeitliche Begrenzung für einzelne Abschnitte vorgab. Während der Entwicklung wurden regelmäßige Sitzungen zwischen Kunde und Entwickler gehalten um den aktuellen Stand des Entwicklers mit den Vorstellungen des Kunden abzugleichen sowie die bevorstehenden Schritte sinnvoll zu planen. Dies vermeidet eine Fehlentwicklung des Entwicklers. Es wurde hierbei ein Schwerpunkt auf genaue Definitionen und Grundlagen gelegt, was zur Folge hatte, dass gravierende Fehlentscheidungen des Entwicklers vermieden werden konnten.

8.2 RESULTAT

Innerhalb dieses Kapitels wird die resultierende Anwendung beschrieben. Es wird aus der Sicht eines Nutzers beschrieben, welcher einen informierten Wissenstand besitzt und die Anwendung mit der Hilfe von Abbildungen beschreibt. Es wird hierbei auf die dazugehörigen Prototypen verwiesen sowie abweichende Designentscheidungen erläutert. Diese Beschreibungsform wurde aufgrund des linearen Funktionsaufbaus gewählt. Als Hilfestellung beinhalten die Abbildungen nummerierte Markierungen, welche innerhalb des erklärenden Textes, von Klammern umgeben, eingefügt werden. Dies soll verdeutlichen, welche Interaktion eines Nutzers die jeweilig markierte Schalt- oder Eingabefläche nutzt.

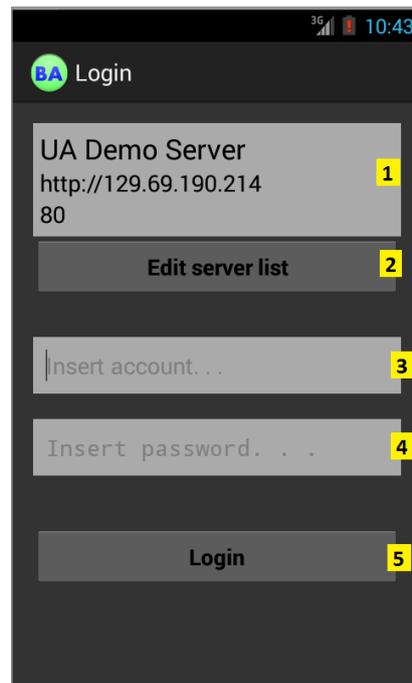


Abbildung 21 Anmeldefenster Resultat

Startet der Mitarbeiter die Anwendung gelangt er in das innerhalb der Abbildung 21 Anmeldefenster Resultat abgebildete Fenster. Die Darstellung entspricht dem entworfenem Prototypen (s. Pkt.6.2.1). Für eine Anmeldung wählt der Mitarbeiter einen Evergreen-Server aus (1) und gibt sein Nutzerkonto (3) sowie das dazugehörige Passwort (4) an. Danach wählt er die Login Schaltfläche (5) und startet die Anmeldung. Befindet sich der benötigte Evergreen-Server nicht in der Serverliste, kann über die Serverliste editieren Schaltfläche (2) der Serverliste editieren Dialog aufgerufen werden.

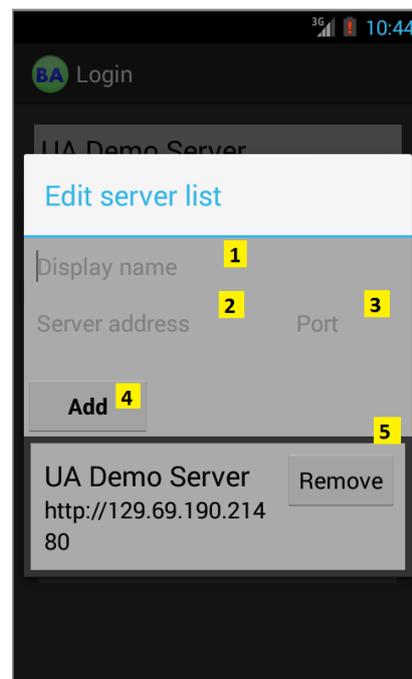


Abbildung 22 Serverliste editieren Resultat

Der Serverliste editieren Dialog, abgebildet in Abbildung 22 Serverliste editieren Resultat, wurde mit einer minimalen Änderung an dessen Prototyp (s. Pkt. 6.2.2) übernommen. Ein Nutzer, welcher die Serverliste zu manipulieren beabsichtigt, besitzt innerhalb dieses Dialoges die Möglichkeit, über die Eingabe eines eindeutigen Servernamens (1), dessen Netzwerkadresse (2) und Port (3), mittels der Hinzufügen-Schaltfläche (4) einen neuen Server der Liste hinzuzufügen. Diese Serverliste wird innerhalb dieses Dialoges als Liste dargestellt (5) deren Elemente die Informationen Servername, Serveradresse und Server Port visualisieren. Zudem besitzen diese jeweils eine entfernen Schaltfläche, um nicht mehr benötigte Server aus der Liste zu entfernen. Innerhalb der Implementierung wurde für die entfernen Schaltfläche kein Symbol sondern eine beschriftete Schaltfläche genutzt, um die Auswirkung aus Nutzersicht direkt ersichtlich zu machen.

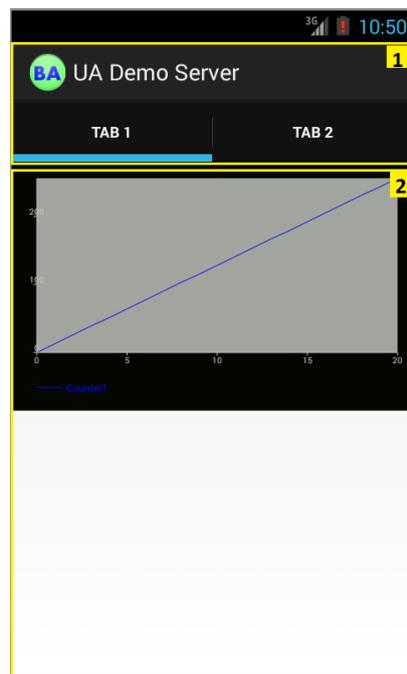


Abbildung 23 Hauptfenster Resultat

Nach einer positiven Anmeldung gelangt der Nutzer in das Hauptfenster, abgebildet in Abbildung 23 Hauptfenster Resultat. Die Anordnung der Elemente orientieren sich am Prototypen (s. Pkt. 6.2.3) wurden aber zugunsten einer intuitiveren Bedienung angepasst. Die Tab-Leiste wurde durch eine Tab Actionbar (1) ersetzt. Das Tab-Fenster (2) beinhaltet die für die Visualisierung zuständigen Diagramme und die zusätzliche Funktion über eine Wischgeste von rechts nach links oder von links nach rechts zwischen benachbarten Tab-Fenstern zu navigieren.

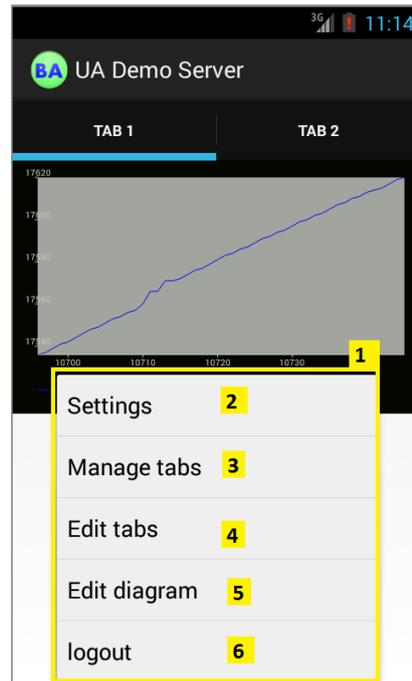


Abbildung 24 Optionen Resultat

Die Optionen, abgebildet in Abbildung 24 Optionen Resultat, wurden wie im Prototypen (s. Pkt. 6.2.4) designt, bei der Abbildung handelt es sich um das Android 4.4.2, welches auf einem simulierten LG Nexus 4 läuft. Die Optionen Schaltfläche, für den Aufruf der Optionen, wurde entfernt und mittels der Android Menüführung ersetzt, für eine dem genutzten Gerätes individuelle und optimierte Menüführung. Die zur Verfügung stehenden Optionen (1) beinhalten die innerhalb des Prototyps geplanten Optionen. Es stehen die Optionen Maschineneinstellungen (2), Tab-Fenster verwalten (3), Tab-Fenster editieren (4), Diagramm editieren (5) und abmelden (6) zur Verfügung. Die Tab-Fenster verwalten Option aus dem Prototyp wurde zugunsten der Bedienbarkeit in zwei Optionen, Tab-Fenster editieren (4) und Diagramm editieren (5), aufgeteilt. Diese Optionenleiste ermöglicht den Zugang zu allen für die Konfiguration einer Serverkonfiguration nötigen Funktionen.

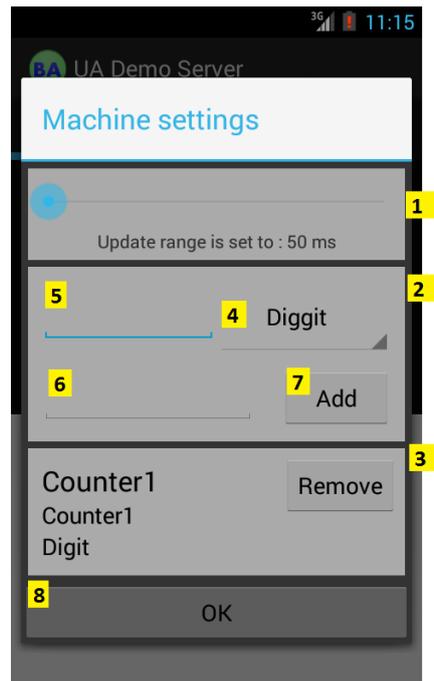


Abbildung 25 Maschineneinstellungen Dialog Resultat

Für eine sinnvolle Nutzung benötigt der Nutzer zunächst Variablen, deren Werte observiert werden sollen. Die Verwaltung wird innerhalb des Maschineneinstellungen-Dialoges ermöglicht, abgebildet in Abbildung 25 Maschineneinstellungen Dialog Resultat. Dieser ermöglicht über einen Schieberegler die Einstellung der Aktualisierungsrate (1) und das Verwalten der verfügbaren Variablen, welche Maschinendaten repräsentieren. Innerhalb des mittleren Bereichs (2) dieses Dialoges, können neue Variablen der globalen Variablenliste hinzugefügt werden. Hierfür gibt der Nutzer einen innerhalb der Serverkonfiguration eindeutigen Variablennamen (5), Variablenadresse (6) und dessen Typ (4) an. Danach kann er über die hinzufügen Schaltfläche (7) die Variable erstellen und der globalen Variablenliste hinzufügen. Diese wird über eine Liste (3) dargestellt deren Elemente den Variablennamen, die Variablenadresse sowie dessen Typ visualisieren. Jedes Element besitzt eine Entfernen-Schaltfläche um das jeweilige Element und dessen visualisierte Variable aus der Liste zu entfernen. Der Dialog kann über die Bestätigen-Schaltfläche (8) geschlossen werden. Das Design wurde aus dem Prototyp (s. Pkt. 6.2.8) übernommen, mit der Änderung, dass die Entfernen-Schaltfläche (8) eine Beschriftung trägt und kein Symbol. Hieraus wird die Auswirkung dieser Schaltfläche für den Nutzer besser ersichtlicher.

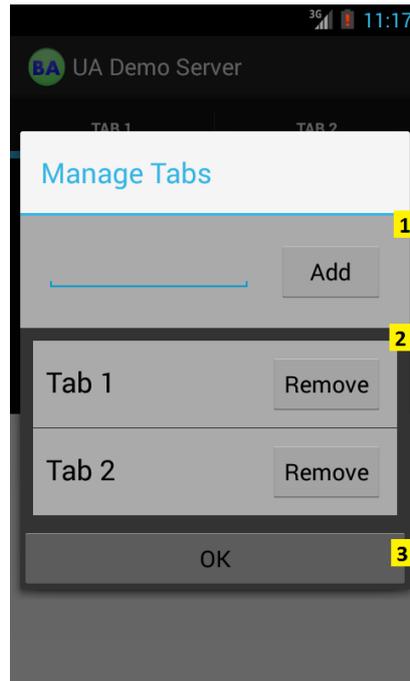


Abbildung 26 Tab-Fenster verwalten Dialog Resultat

Die Darstellung von Diagrammen benötigt mindestens ein Tab-Fenster, welches als Container fungiert. Tab-Fenster werden innerhalb des Tab-Fenster verwalten-Dialoges verwaltet, abgebildet in Abbildung 26 Tab-Fenster verwalten Dialog Resultat . Im oberen Bereich (1) können, durch die Eingabe eines innerhalb der Serverkonfiguration eindeutigen Namens und der hinzufügen Schaltfläche, neue Tab-Fenster der Liste (2) hinzugefügt werden. Innerhalb der Tab-Fenster verwalten Liste (2) werden die jeweiligen Tab-Fenster als separate Elemente mit ihrem Namen dargestellt, welche über eine Entfernen-Schaltfläche aus der Liste entfernt werden können. Der Dialog kann über eine Bestätigen-Schaltfläche (3) geschlossen werden. Das Design entspricht dem Prototyp (s. Pkt. 6.2.5). Es wurden die Entfernen-Schaltflächen zugunsten der Bedienbarkeit anstelle mit Symbolen mit einer Beschriftung versehen, um die Funktion klarer zu deklarieren und Missverständnisse zu vermeiden.

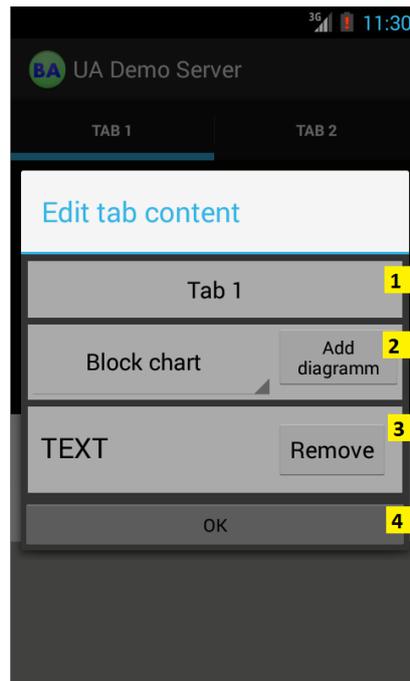


Abbildung 27 Tab-Fenster editieren Dialog Resultat

Für die Darstellung von Variablen werden Diagramme benötigt. Diagramme können innerhalb des Tab-Fenster editieren Dialoges, abgebildet in Abbildung 27 Tab-Fenster editieren Dialog Resultat, zu den jeweiligen Tab-Fenstern hinzugefügt werden. Hierfür wählt der Nutzer ein existierendes Tab-Fenster aus (1) und kann im Anschluss einen Diagrammtyp wählen und diesem der Liste an Diagrammen innerhalb des gewählten Tab-Fensters über eine Hinzufügen-Schaltfläche hinzufügen (2). Über eine Liste (3) werden die enthaltenen Diagrammtypen visualisiert. Diese können über eine Entfernen-Schaltfläche aus der Liste und dem Tab-Fenster entfernt werden. Der Dialog kann über eine Bestätigen-Schaltfläche (4) geschlossen werden. Das Design weicht stark vom Prototyp (s. Pkt. 6.2.6) ab, da dieser für die Bedienung auf einem Smartphone mit 3,5 bis 5 Zoll Displaydiagonale sich als sehr überladen erwiesen hat. Darum wurde die Variablenverwaltung der einzelnen Diagramme in einen weiteren Dialog ausgelagert.

Dieser Diagramm editieren Dialog, abgebildet in Abbildung 28 Diagramm editieren Dialog Resultat, ermöglicht die Konfiguration einzelner Diagramme. Hierbei wird zunächst das entsprechende Tab-Fenster gewählt (1). Danach kann ein existierendes Diagramm innerhalb eines Tab-Fensters gewählt werden (2). Wählt der Nutzer ein Diagramm aus wird die Variablenauswahl (3) freigeschaltet und die Variablenliste (5) mit den bereits an das Diagramm gebundenen Variablen befüllt. Der Nutzer ist nun in der Lage über die Variablenliste (5) einzelne Variablen aus der Anzeige eines Diagrammes zu entfernen oder er wählt eine noch nicht gebundene Variable aus (3) und fügt diese über die Hinzufügen-Schaltfläche (4) der Variablenliste des jeweiligen Diagrammes hinzu. Die Variablenliste (5) repräsentiert die Variablen, welche über das Diagramm visualisiert werden. Die Darstellung selbst hängt vom jeweiligen Diagrammtyp ab. Dieser Dialog kann über die bestätigen Schaltfläche (6) geschlossen werden. Das Design wurde an die bereits existierenden Dialoge angepasst und neu entworfen, da es für diesen Dialog kein Prototyp existiert.

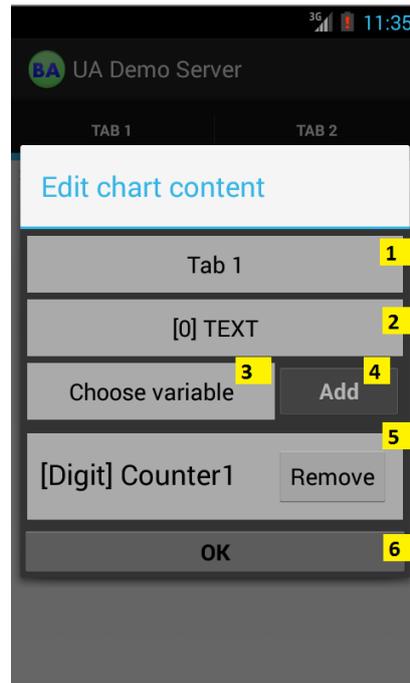


Abbildung 28 Diagramm editieren Dialog Resultat

9 ZUSAMMENFASSUNG UND AUSBLICK

Innerhalb dieser Arbeit wurden mobile Geräte aus dem Smartphone-Segment auf die Möglichkeit analysiert mittels eines externen Evergreen-Servers Fertigungsmaschinen zu observieren. Dabei wurden mögliche Darstellungsformen recherchiert aus denen die Darstellungsformen Liniendiagramm, Balkendiagramm, textuelle Darstellung sowie eine tabellarische Darstellung resultieren. Gefolgt von einer Recherche über die auf dem Markt existierenden Smartphone-Betriebssysteme und ihrer prozentualen Verteilung. Hieraus resultierten die Betriebssysteme Android, IOS, Symbian, Blackberry und Windows Phone mit einer prozentualen Verteilung von 79,2%, 13,2%, 0,2%, 2,9% und 3,7%. Aufbauend auf diesen zwei Recherchen wurde nach einer optimalen Programmierplattform für eine Beispielanwendung gesucht. Dies wurde mittels einer Analyse durchgeführt. In dieser wurden Kriterien aufgestellt, welche aus mehreren K.O. – Kriterien und einem Entscheidungskriterium bestanden. Die K.O. Kriterien repräsentierten die für eine Umsetzung nötigen Eigenschaften einer Plattform um die Entwicklung einer Anwendung zu ermöglichen sowie eine für den Kunden nachvollziehbare Entscheidung der Auswahl zu treffen. Das Entscheidungskriterium wurde genutzt um eine Plattform auszuwählen sollten mehrere Plattformen alle K.O. – Kriterien erfüllen. Programmierplattformen, welche aus einer Recherche resultierten, wurden auf Erfüllung dieser Kriterien hin geprüft. Das Resultat ergab, dass jede Programmierplattform die Möglichkeit bietet eine Anwendung für die gewünschte Observierung zu erstellen. Trotz dessen wurden anhand der K.O. Kriterien manche Programmierplattformen aus der näheren Auswahl entfernt, Qt mobile besitzt zum Zeitpunkt dieser Arbeit keine Stabile Version, eine native Symbian-Anwendung wurde aufgrund des eingestellten Supports ausgeschlossen und Interpreter basierte Plattformen wie HTML in Kombination mit Javascript sowie Google Chrome Apps wurden aufgrund einer negativ prognostizierten Performance nicht mit einbezogen. Es resultierte eine native Android App als optimale Programmierplattform, diese erfüllt alle K.O. - Kriterien und deckt einen Marktanteil von 82,2% ab, welcher der höchste Wert für das Entscheidungskriterium unter den zur Auswahl stehenden ergab.

Aufbauend auf diesen Informationen wurde innerhalb eines agilen Entwicklungsprozesses eine Demonstrator Anwendung entworfen und anschließend umgesetzt.

Aufgrund der Informationen aus dem Analyseteil, der Entwicklung der angestrebten Anwendung sowie den Gesprächen mit dem Kunden ist der Ausblick für Smartphones positiv. Eine Maschine mittels der entwickelten Anwendung zu observieren ist in der Relation zu einer Eigenentwicklung einfacher und benötigt wenig technisches Verständnis über Kommunikationsmechanismen oder die zu observierende Fertigungsmaschine. Zusätzlich hinterlässt die Bedienung einen intuitiven Eindruck und das PreisLeistungsverhältnis überzeugt. Dennoch ist die Nutzung ohne Informationen über die zu observierenden Maschinendaten nicht möglich und benötigt hierfür einen Mitarbeiter, welcher darin kundig ist die existierenden Maschinendaten als Variablen in die Anwendung einzuspeisen. Es besteht zusätzlich die Gefahr einer Fehlinterpretation, wenn Wertespitzen von Variablen nur sporadisch auftreten und von der Anwendung nicht im richtigen Zeitpunkte angefordert werden, so werden diese auch nicht innerhalb eines Diagrammes angezeigt. Darum sollte für eine finale Anwendung trotz höherer Belastung auf die Abonnier Funktion eines Evergreen Servers gesetzt werden und dieser für eine besseren Abstraktion der Variablen mittels einer Lesefunktion über bestehende Maschinendaten erweitert werden.

10 LITERATURVERZEICHNIS

- [1] T. Neil, Mobile Design Pattern Gallery, O'Reilly Media, 2012.
- [2] M. Gargenta, Einführung in die Android Entwicklung, O'Reilly, 2011.
- [3] The JSON Group, „Einführung in JSON,“ 26 4 2014. [Online]. Available: <http://www.json.org/json-de.html>.
- [4] Evergreen, „Evergreen Spezifikation,“ Stuttgart, 2013.
- [5] T. Költzsch, „Mobile Betriebssysteme,“ 25 4 2014. [Online]. Available: <http://www.golem.de/news/mobile-betriebssysteme-android-laeuft-auf-80-prozent-der-smartphones-1308-100854.html>.
- [6] W. Isaacson, Steve Jobs, C. Bertelsmann Verlag, 2011.
- [7] J. N. J. L. F. O. David Mark, Beginning iOS 6 Development: Exploring the iOS SDK, Apress, 2013.
- [8] Apple Inc., „IOS Dev Center,“ 26 4 2014. [Online]. Available: <https://developer.apple.com/devcenter/ios/index.action>.
- [9] Developers, „Android SDK,“ 26 4 2014. [Online]. Available: <http://developer.android.com/sdk/index.html>.
- [10] Open Handset Alliance, „alliance,“ 25 4 2014. [Online]. Available: http://www.openhandsetalliance.com/oha_overview.html.
- [11] t3n, „Flat Design,“ 25 4 2014. [Online]. Available: <http://t3n.de/tag/flat-design>.
- [12] Microsoft, „Windows Phone SDK,“ 25 4 2014. [Online]. Available: <http://www.microsoft.com/de-de/download/details.aspx?id=35471>.
- [13] E. C. Henry Lee, Beginning Windows Phone 7 Development, Apress, 2010.
- [14] M. Tasker, Professional Symbian Programming: Mobile Solutions on the EPOC Platform, Wrox Press Limite, Erscheinungsdatum2000.
- [15] J. M. Wargo, BlackBerry Development Fundamentals, Addison-Wesley, 2009.
- [16] M. S. Jasmin Blanchette, C++ GUI Programmierung mit Qt 4, München: Addison-Wesley, 2007.
- [17] M. Stückler, „Chart.js: Schicke JavaScript-Diagramme mit HTML5 und CSS3,“ 25 4 2014. [Online]. Available: <http://t3n.de/news/chartjs-schicke-450755/>.

-
- [18] Google, „Using Google Charts,“ 26 4 2014. [Online]. Available: <https://developers.google.com/chart/interactive/docs/index>.
- [19] A. Kingsley-Hughes, „The BIG browser benchmark (January 2013 edition),“ 26 4 2014. [Online]. Available: <http://www.zdnet.com/the-big-browser-benchmark-january-2013-edition-7000009776/>.
- [20] D. Würstl, „Unterschiede und Vergleich native Apps vs. Web Apps,“ 26 4 2014. [Online]. Available: <http://www.app-entwickler-verzeichnis.de/faq-app-entwicklung/11-definitionen/107-unterschiede-und-vergleich-native-apps-vs-web-apps>.
- [21] Google Inc, „What Are Chrome Apps?,“ 25 4 2014. [Online]. Available: https://developer.chrome.com/apps/about_apps.
- [22] The 4ViewSoft Company, „AChartEngine,“ The 4ViewSoft Company, 25 4 2014. [Online]. Available: <http://www.achartengine.org/content/demo.html>.
- [23] BlackBerry Limited, „Runtime for Android apps,“ 26 4 2014. [Online]. Available: <http://developer.blackberry.com/android/tools/>.
- [24] terryworona, „JBChartView,“ 26 4 2014. [Online]. Available: <https://github.com/Jawbone/JBChartView>.
- [25] „TeeChart for BlackBerry Charting Library released,“ 25 4 2014. [Online]. Available: http://www.steema.com/entry/140/TeeChart_for_BlackBerry_Charting_Library_released.
- [26] ailon, „AMCharts,“ 26 4 2014. [Online]. Available: <https://github.com/ailon/amCharts-Quick-Charts>.
- [27] The Apache Software Foundation, „Apache License,“ 25 4 2014. [Online]. Available: <http://www.apache.org/licenses/LICENSE-2.0>.
- [28] The Apache Software Foundation, „Welcome to Apache Axis2/Java,“ 25 4 2014. [Online]. Available: <http://axis.apache.org/axis2/java/core/>.
- [29] Apple Inc., „Apple Reports,“ 26 4 2014. [Online]. Available: <http://images.apple.com/pr/>.

11 ANHANG

11.1 ANWENDUNGSFÄLLE (USE CASES)

11.1.1 AKTEURE

Evergreen-Server	Ein Evergreen-Server, der für die Kommunikation zwischen dem Client (hier ein Smartphone) und der Maschine selbst von Nöten ist.
Mitarbeiter	Ein Mitarbeiter der Firma, der berechtigt ist ein Smartphone mit der Applikation zu nutzen

11.1.2 PRIORITÄTEN

Niedrig: 6 Use Cases

Mittel: 4 Use Cases

Hoch: 6 Use Cases

11.1.3 ANMELDEN

Die Interaktion mit einem Evergreen Server setzt eine gültige Session voraus, diese wird über eine erfolgreiche Anmeldung erhalten, welche mittels eines gültigen Nutzers sowie des dazu gehörenden Passwortes erfolgreich vollzogen werden kann.

Ziel		Der Mitarbeiter nutzt seinen Login sowie das dazugehörige Passwort um sich zu Authentifizieren und eine gültige Session zu erhalten.	
Akteure		Mitarbeiter , Evergreen-Server	
Beschreibung:			
Ebene		Benutzersicht	
Priorität		Hoch	
Normalablauf			
Vorbedingung:		Mitarbeiter besitzt keine gültige Session.	
1	Mitarbeiter	Mitarbeiter wählt einen Ziel Evergreen-Server aus und gibt seine Anmeldedaten ein	
2	Mitarbeiter	Mitarbeiter sendet seine Anmeldedaten an den Evergreen-Server	
		Bedingung für Sonderfall Keine Verbindung zu einem Evergreen Server	Alternativablauf 2a
		Bedingung für Sonderfall Adresse des Servers ist ungültig	Alternativablauf 2b
3	Evergreen-Server	Server empfängt Anmeldedaten und gibt bei korrekten Daten eine Session zurück	
		Bedingung für Sonderfall Anmeldedaten sind inkorrekt	Alternativablauf 3a
4	Smartphone	Die Session wird als aktive Session festgelegt sowie der eingestellte Server als Ziel Server.	
5	Smartphone	Das "Anmelden" Fenster wird geschlossen und es wird zum Hauptfenster navigiert	
Nachbedingung:		Der Mitarbeiter besitzt nun eine gültige Session um mit dem Server zu interagieren Das Hauptfenster ist das aktive Fenster	

Alternativablauf 2a		
Vorbedingung:		Keine Verbindung zu einem Evergreen-Server
2a1	Mitarbeiter	Es wird ein Verbindungsfehler visualisiert
Nachbedingung:		Mitarbeiter wurde über eine fehlende Verbindung informiert
Alternativablauf 2b		
Vorbedingung:		Adresse des Servers ist ungültig
2b1	Mitarbeiter	Es wird ein Verbindungsfehler visualisiert der informiert über einen fehlenden oder nicht erreichbare Adresse
Nachbedingung:		Mitarbeiter ist informiert über eine Fehlerhafte Adresse oder eines nicht erreichbaren Evergreen-Servers
Alternativablauf 3a		
Vorbedingung:		Anmeldedaten sind inkorrekt
3a1	Evergreen-Server	Server gibt einen Fehler anstelle einer Session zurück
3a2	Mitarbeiter	Bekommt den Fehler über falsche Login Daten innerhalb seiner Applikation angezeigt
Nachbedingung:		Mitarbeiter wurde auf inkorrekte Login-Daten hingewiesen Mitarbeiter ist nicht in Besitz einer gültigen Session

11.1.4 ABMELDEN

Möchte ein Mitarbeiter die Verwendung der Anwendung beenden und somit eine gültige Session schließen, wird eine Abmeldung vollzogen, welche bei der Verwendung einer gültigen Session diese schließt und sie für eine weitere Nutzung unbrauchbar macht.

Ziel		Der Mitarbeiter möchte sich abmelden	
Akteure		Mitarbeiter , Evergreen-Server	
Beschreibung:			
Ebene		Benutzersicht	
Priorität		Hoch	
Normalablauf			
Vorbedingung:		Das Hauptfenster ist das aktive Fenster Es existiert eine aktive Session	
1	Mitarbeiter	Mitarbeiter wählt die "Abmelden" Option aus	
2	Mitarbeiter	Abmelden Request wird an den Evergreen-Server versendet	
		Bedingung für Sonderfall Keine Verbindung zu einem Evergreen-Server	Alternativablauf 2a
3	Evergreen-Server	Server empfängt Anmeldedaten und gibt bei korrekten Daten eine Session zurück	
		Bedingung für Sonderfall Eine ungültige Session	Alternativablauf 3a
4	Smartphone	Session wird aus der Umgebung entfernt, der Aktualisierungsprozess wird im Hintergrund beendet, das Hauptfenster wird geschlossen und es wird zum "Anmelde" Fenster navigiert.	
Nachbedingung:		Die Session wurde deaktiviert und gelöscht Der Aktualisierungsprozess im Hintergrund ist beendet Das aktive Fenster ist das "Anmelde" Fenster	
Alternativablauf 2a			

Vorbedingung:		Keine Verbindung zu einem Evergreen-Server
2a1	Mitarbeiter	Es wird ein Verbindungsfehler visualisiert
Nachbedingung:		Mitarbeiter wurde über eine fehlende Verbindung informiert
Alternativablauf 3a		
Vorbedingung:		Eine ungültige Session
3a1	Evergreen-Server	Server erkennt ungültige Session
3a2	Mitarbeiter	Es wird eine Meldung visualisiert, die dem Mitarbeiter über eine ungültige Session informiert
Nachbedingung:		Mitarbeiter ist informiert über eine Fehlerhafte Session

11.1.5 *DIAGRAMM ENTFERNEN*

Wird ein verwendetes Diagramm nicht mehr benötigt oder wurde unabsichtlich hinzugefügt, kann über diesen Anwendungsfall dieses entfernt werden.

Ziel		Der Mitarbeiter möchte ein Diagramm aus einem aktiven Tab-Fenster entfernen	
Akteure		Mitarbeiter	
Beschreibung:			
Ebene		Benutzersicht	
Priorität		Niedrig	
Normalablauf			
Vorbedingung:		Mitarbeiter möchte ein Diagramm aus einem aktiven Tab-Fenster entfernen	
1	Mitarbeiter	Mitarbeiter wählt die "Entfernen" Optionen bei einem Diagramm aus.	
2	Mitarbeiter	Bedingung für Sonderfall Mitarbeiter wollte das Diagramm nicht entfernen und wählt "Abbrechen"	Alternativablauf 2a
3	Smartphone	Diagramm wird aus dem aktiven Tab-Fenster entfernt, der Hintergrundprozess für die Aktualisierung wird beendet sowie der Dialog wird geschlossen.	
Nachbedingung:		Mitarbeiter besitzt nun eine gültige Session um mit dem Server zu interagieren	
Alternativablauf 2a			
Vorbedingung:		Mitarbeiter wollte das Diagramm nicht entfernen und wählt "Abbrechen"	
2a1	Smartphone	Dialog wird geschlossen	
Nachbedingung:		Diagramm wird nicht entfernt	

11.1.6 NEUES DIAGRAMM HINZUFÜGEN

Wird für einen bestimmten Datensatz eine eigene oder noch nicht vorhandene Darstellung benötigt, wird über diesen Anwendungsfall dies ermöglicht.

Ziel		Der Mitarbeiter möchte ein neues Diagramm in sein Tab-Fenster hinzufügen	
Akteure		Mitarbeiter	
Beschreibung:		Der Mitarbeiter möchte in sein aktives Tab-Fenster ein neues Diagramm hinzufügen um dieses zur Anzeige von Maschinendaten zu nutzen.	
Ebene		Benutzersicht	
Priorität		Mittel	
Normalablauf			
Vorbedingung:		Mitarbeiter möchte ein Diagramm in ein aktives Tab-Fenster hinzufügen	
1	Mitarbeiter	Mitarbeiter wählt die Neues Diagramm hinzufügen -Option aus	
2	Smartphone	Es wird ein Dialog angezeigt, der alle zur Wahl stehenden Diagramme anzeigt sowie eine Auswahl über die aktiven Tab-Fenster in dieses es eingefügt werden soll.	
		Bedingung für Sonderfall es existiert kein aktives Tab-Fenster	Alternativablauf 2a
3	Mitarbeiter	Nutzer wählt ein Diagramm und ein Tab-Fenster aus und bestätigt seine Auswahl	
4	Smartphone	Es wird ein Hintergrundprozess angelegt der für zukünftig hinzugefügte Variablen die Aktualisierung übernimmt	
Nachbedingung:		Es wurde ein Diagramm in dem ausgewählten aktiven Tab-Fenster angelegt	
Alternativablauf 2a			
Vorbedingung:		Es existiert kein aktives Tab-Fenster	
2a1	Mitarbeiter	Mitarbeiter kann den Dialog nicht mit Ok bestätigen und wählt somit die Abbrechen Option auf	

2a2	Smartphone	Diagramm hinzufügen Dialog wird geschlossen
Nachbedingung:		Diagramm hinzufügen Dialog wird geschlossen

11.1.7 VARIABLE AUS EINEM DIAGRAMM ENTFERNEN

Durch diesen Anwendungsfall wird es ermöglicht Variablen aus einem Diagramm zu entfernen, dies kann daraus resultieren, dass diese nicht weiter benötigt wird oder nicht absichtlich hinzugefügt wurde. Es hat zur Folge, dass alle korrelierenden Werte mit aus dem Diagramm entfernt werden sowie die Anzeige erneut dargestellt wird.

Ziel		Mitarbeiter möchte eine Variable aus einem Diagramm entfernen, da er diese nicht mehr benötigt.
Akteure		Mitarbeiter
Beschreibung:		
Ebene		Benutzersicht
Priorität		Niedrig
Normalablauf		
Vorbedingung:		Es existiert ein Diagramm im aktiven Tab-Fenster
1	Mitarbeiter	Mitarbeiter wählt die Option "Einstellungen" bei dem gewünschten Diagramm aus.
2	Smartphone	Es wird der Optionen Dialog für das gewählte Diagramm angezeigt
3	Mitarbeiter	Der Mitarbeiter wählt die Option "Entfernen" an der gewünschten Variable aus, unter dem Hauptpunkt "Anzeige Variablen"
		Bedingung für Sonderfall Die Ziel Variable ist nicht unter den "Anzeige Variablen"
		Alternativablauf 3a
4	Smartphone	Variable wird aus dem Diagramm entfernt sowie die Variable selbst wird aus dem Hintergrundprozess entfernt der für die Aktualisierung zuständig ist.
Nachbedingung:		Die Ziel Variable wird aus dem Diagramm, dessen Anzeige und dem Hintergrundprozess zur Aktualisierung entfernt
Alternativablauf 3a		

Vorbedingung:		Die Ziel Variable ist nicht unter den "Anzeige Variablen"
3a1	Mitarbeiter	Mitarbeiter kann die Ziel Variable nicht unter den "Anzeige Variablen" finden, somit ist sie nicht unter diesen vorhanden und der Mitarbeiter hat keinen Bedarf mehr den Vorgang fortzusetzen.
Nachbedingung:		Die Ziel Variable ist nicht unter den "Anzeige Variablen"

11.1.8 VARIABLE IN EIN DIAGRAMM HINZUFÜGEN

Ist ein geeignetes Diagramm vorhanden und es soll die Darstellung von einer zusätzlichen Variable übernehmen tritt dieser Anwendungsfall ein, in diesem wird der Ablauf dargestellt. In welchem ein Diagramm über eine weitere Variable für die Anzeige ergänzt wird, dies hat eine erneute Darstellung des Diagramms zur Folge.

Ziel		Mitarbeiter möchte eine Variable zur Anzeige zu einem Diagramm hinzufügen.
Akteure		Mitarbeiter
Beschreibung:		
Ebene		Benutzersicht
Priorität		Hoch
Normalablauf		
Vorbedingung:		Es existiert ein Diagramm im aktiven Tab-Fenster
1	Mitarbeiter	Mitarbeiter wählt die Option "Einstellungen" bei dem gewünschten Diagramm aus.
2	Smartphone	Es wird der Optionen Dialog zum gewählten Diagramm angezeigt
3	Mitarbeiter	Mitarbeiter wählt im "Anzeige Variablen" Bereich die Option "Variable hinzufügen"
4	Smartphone	Es erscheint der "Variablen Auswahl" Dialog der aus der Liste, der zur auswahlstehenden und noch nicht ausgewählten Variablen, die noch offene Auswahl Anzeigt.
5	Mitarbeiter	Der Mitarbeiter wählt die Ziel Variable aus und wählt die "hinzufügen" Option aus
		Bedingung für Sonderfall Die Ziel Variable steht nicht zur Auswahl
		Alternativablauf 5a
6	Smartphone	Der "Variablen Auswahl" Dialog schliesst sich, die Variable wird der Anzeige des Ziel Diagramms hinzugefügt und dem für die Aktualisierung zuständigen Hintergrundprozess
Nachbedingung:		Die Ziel Variable wurde dem Ziel Diagramm hinzugefügt
Alternativablauf 5a		
Vorbedingung:		Die Ziel Variable steht nicht zur Auswahl

5a1	Mitarbeiter	Der Mitarbeiter findet die Ziel Variable nicht und muss den Vorgang abbrechen, hierfür wählt er die "Abbrechen" Option des "Variablen Auswahl" Dialogs aus.
5a2	Smartphone	Der "Variablen Auswahl" Dialog wird geschlossen
Nachbedingung:		Der "Variable hinzufügen" Dialog ist geschlossen und der Mitarbeiter befindet sich wieder in seinem Ausgangszustand

11.1.9 *TAB-FENSTER ERSTELLEN*

Ein neues Tab-Fenster erstellen ist ein Anwendungsfall, welche auftritt wenn für die Gruppierung von Diagrammen ein weiteres Tab-Fenster benötigt wird. Diagramme können nur in solchen Tab-Fenstern existieren und sind strikt an diese gebunden.

Ziel	Der Mitarbeiter möchte ein neues Tab-Fenster erstellen	
Akteure	Mitarbeiter	
Beschreibung:		
Ebene	Benutzersicht	
Priorität	Hoch	
Normalablauf		
Vorbedingung:	Der Mitarbeiter möchte ein neues aktives Tab-Fenster erstellen	
1	Mitarbeiter	Der Mitarbeiter wählt im Navigation Drawer die Option "Neues Tab-Fenster" aus
2	Smartphone	Es wird der "Neues Tab-Fenster" Dialog angezeigt.
3	Mitarbeiter	Der Mitarbeiter wählt einen Namen für das neue Tab-Fenster aus und wählt die "hinzufügen" Option aus
4	Smartphone	Es wird ein neues Tab-Fenster in die Actionbar hinzugefügt sowie dieser wird als neues aktives Tab-Fenster ausgewählt. Der Dialog wird geschlossen.
Nachbedingung:	Ein neues Tab-Fenster wurde angelegt	

11.1.10 *TAB-FENSTER ENTFERNEN*

Wird ein Tab-Fenster nicht mehr benötigt oder dessen gruppierten Diagramme, kann dieses über diesen Anwendungsfall entfernt werden. Daraus resultiert, dass alle darin enthaltenen Variablen abgemeldet und deren umfassende Diagramme verworfen werden.

Ziel		Der Mitarbeiter möchte ein neues Tab-Fenster erstellen
Akteure		Mitarbeiter
Beschreibung:		
Ebene		Benutzersicht
Priorität		Niedrig
Normalablauf		
Vorbedingung:		Der Mitarbeiter möchte ein existierendes Tab-Fenster entfernen
1	Mitarbeiter	Der Mitarbeiter wählt im Navigation Drawer die Option "Tab-Fenster entfernen" aus
2	Smartphone	Es wird der "Tab-Fenster entfernen" Dialog angezeigt.
3	Mitarbeiter	Der Mitarbeiter wählt das Ziel Tab-Fenster aus und wählt die "entfernen" Option aus
4	Smartphone	Das Ziel Tab-Fenster wird entfernt, handelt es sich um das aktive Tab-Fenster wird das nächst gelegene Tab-Fenster zum aktiven Tab-Fenster, falls ein solches existiert. Der Dialog wird geschlossen
Nachbedingung:		Das Ziel Tab-Fenster wurde entfernt

11.1.11 AKTIVES TAB-FENSTER WECHSELN

Dieser Anwendungsfall ermöglicht zwischen mehreren Tab-Fenstern zu wechseln um so einen maximalen Nutzen aus der Gruppierung von Diagramm zu erhalten. Bei einem Wechsel zwischen Tab-Fenstern werde andere Tab-Fenster weiterhin aktualisiert einzig die Darstellung der Diagramme wird erst bei einem Wechsel erneut erstellt.

Ziel	Ein anderes Tab-Fenster zu aktivieren um zwischen unterschiedlichen Diagramm / Anordnungen zu wechseln.	
Akteure	Mitarbeiter	
Beschreibung:		
Ebene	Benutzersicht	
Priorität	Mittel	
Normalablauf		
Vorbedingung:	Es existiert ein inaktives Ziel Tab-Fenster	
1	Mitarbeiter	Der Mitarbeiter wählt in der Actionbar das Ziel Tab-Fenster aus.
2	Smartphone	Es wird das aktuelle Tab-Fenster deaktiviert, das Ziel Tab-Fenster aktiviert und in die Anzeige verschoben
Nachbedingung:	Das Ziel Tab-Fenster ist nun das aktive Tab-Fenster	

11.1.12 *MASCHINEN OPTIONEN EINSEHEN*

Über diesen Anwendungsfall verschafft sich ein Mitarbeiter Zugriff auf die Einstellungen der genutzten Anwendung. Es sind dabei die Aktualisierungsrate der Variablen sowie die global zur Verfügung stehenden Variablen für eine Maschine einsehbar.

Ziel		Der Mitarbeiter möchte Zugriff auf die zur Verfügung stehenden Einstellungen bezüglich der zu observierenden Maschine erhalten
Akteure		Mitarbeiter
Beschreibung:		
Ebene		Benutzersicht
Priorität		Niedrig
Normalablauf		
Vorbedingung:		Der Mitarbeiter befindet sich im Hauptfenster
1	Mitarbeiter	Der Mitarbeiter wählt im Navigation Drawer die Option "Maschinen Einstellungen"
2	Smartphone	Es wird ein neues "Maschinen Einstellungen" Fenster geöffnet
Nachbedingung:		Der Mitarbeiter befindet sich im "Maschinen Einstellungen" Fenster Der Mitarbeiter besitzt nun Zugriff auf die Maschinen Einstellungen

11.1.13 VARIABLE IN GLOBALER LISTE MANUELL HINZUFÜGEN

Dieser Anwendungsfall ermöglicht es einem Mitarbeiter eine Variable zu der globalen Variablenliste hinzuzufügen. Dies wiederum ermöglicht es diese Variable nun innerhalb der Anwendung zu Diagrammen hinzuzufügen.

Ziel		Der Mitarbeiter möchte manuell eine Variable zur global verfügbaren Variablen Liste hinzufügen
Akteure		Mitarbeiter
Beschreibung:		
Ebene		Benutzersicht
Priorität		Hoch
Normalablauf		
Vorbedingung:		Der Mitarbeiter befindet sich im "Maschinen Einstellungen" Fenster Der Mitarbeiter besitzt Kenntnis über die zur Verfügung stehenden Variablen
1	Mitarbeiter	Der Mitarbeiter wählt die Option "Variable hinzufügen" aus.
2	Smartphone	Es wird der "Variable hinzufügen" Dialog angezeigt.
3	Mitarbeiter	Der Mitarbeiter gibt den Variablen Adressschlüssel an und wählt die Option "hinzufügen" aus.
4	Smartphone	Der "Variable hinzufügen" Dialog wird geschlossen und die Variable wird in die globale Variablen Liste aufgenommen
		Bedingung für Sonderfall Es existiert bereits eine Variable mit identischen Informationen
		Alternativablauf 4a
Nachbedingung:		Die Variable ist zu der Globalen Liste hinzugefügt
Alternativablauf 4a		
Vorbedingung:		Es existiert bereits eine Variable mit identischen Informationen
4a1	Smartphone	Da die Variable bereits existiert wird nur der "Variable hinzufügen" Dialog geschlossen, es besteht kein weiterer Handlungsbedarf.

Nachbedingung:	Die Variable befindet sich in der globalen Liste.
-----------------------	---

11.1.14 VARIABLE AUS GLOBALER LISTE MANUELL ENTFERNEN

Wird eine Variable nicht mehr benötigt oder existiert in der Ziel Maschine nicht, so kann diese entfernt werden und steht für die Anzeige innerhalb von Diagrammen nicht mehr zur Verfügung.

Ziel	Der Mitarbeiter möchte eine Ziel Variable aus der globalen Variablen Liste entfernen	
Akteure	Mitarbeiter	
Beschreibung:		
Ebene	Technische Details	
Priorität	Niedrig	
Normalablauf		
Vorbedingung:	Der Mitarbeiter befindet sich im "Maschinen Einstellungen" Fenster Die Ziel Variable existiert	
1	Mitarbeiter	Der Mitarbeiter wählt in der Anzeige der Ziel Variable, im Bereich "Globale Variablen", die "entfernen" Option aus
2	Smartphone	Es wird überprüft ob die Ziel Variable in einem Hintergrundprozess eines Diagramms genutzt wird, ist dies nicht der Fall wird sie aus der Liste entfernt und die Anzeige aktualisiert.
		Bedingung für Sonderfall Die Variable wird in einem Hintergrundprozess eines Diagramms genutzt
		Alternativablauf 2a
Nachbedingung:	Die Variable befindet sich nicht mehr in der Globalen Variablen Liste	
Alternativablauf 2a		
Vorbedingung:	Die Variable wird in einem Hintergrundprozess eines Diagramms genutzt	

2a1	Smartphone	Es wird über einen "Toast" Dialog mitgeteilt, dass die Variable sich in Benutzung befindet und diese erst Beendet werden muss bevor man sie entfernen könne.
Nachbedingung:		Der Mitarbeiter ist informiert darüber, dass die Variable sich in Benutzung befindet.

11.1.15 *EXTERNE VARIABLEN LISTEN EINLESEN ALS DATEI*

Um das Einfügen von Variablen zu erleichtern ermöglicht dieser Anwendungsfall das einlesen von einer Variablen Liste über eine Datei, diese entspricht einer Evergreen Server Browser Datei. Variablen die hierbei doppelt vergeben werden, werden übersprungen und über einen Hinweis Dialog darauf aufmerksam gemacht.

Ziel		Der Mitarbeiter möchte eine vordefinierte Variablen Liste einlesen die als XML Datei hinterlegt wurde	
Akteure		Mitarbeiter	
Beschreibung:			
Ebene		Benutzersicht	
Priorität		Mittel	
Normalablauf			
Vorbedingung:		Der Mitarbeiter befindet sich im "Maschinen Einstellungen" Fenster	
1	Mitarbeiter	Der Mitarbeiter wählt die Option "Externe Liste laden" aus.	
2	Smartphone	Es wird ein Dialog geöffnet der den Mitarbeiter dazu auffordert eine Datei auszuwählen	
3	Mitarbeiter	Der Mitarbeiter wählt in diesem Dialog seine Ziel Datei aus und bestätigt seine Auswahl mit der Option "Laden"	
		Bedingung für Sonderfall Die Ziel Datei kann vom Mitarbeiter nicht gefunden werden oder existiert nicht	Alternativablauf 3a
4	Smartphone	Der "Externe Liste laden" Dialog wird geschlossen.	
5	Smartphone	Es wird ein Fortschritts Dialog angezeigt, der den Mitarbeiter signalisiert, dass im Hintergrund die Daten verarbeitet werden.	

6	Smartphone	Wenn die verarbeitete Datei eine valide XML-Datei ist, werden die darin enthaltenen Variablen zur globalen Variablen Liste hinzugefügt, der Fortschritts Dialog wird geschlossen und ein "Toast" Dialog angezeigt, der das erfolgreiche laden der Datei mitteilt.	Alternativablauf 6a
		Bedingung für Sonderfall Die ausgewählte Datei ist keine valide XML-Datei	
7	Mitarbeiter	Der Mitarbeiter wählt die "OK" Option des "Toast" Dialoges.	
8	Smartphone	Der "Toast" Dialog wird geschlossen	
Nachbedingung:		Die Variablen innerhalb der Ziel Datei wurden der globalen Variablen Liste hinzugefügt Der Mitarbeiter findet sich im "Maschinen Einstellungen" Fenster wieder	
Alternativablauf 3a			
Vorbedingung:		Die Ziel Datei kann vom Mitarbeiter nicht gefunden werden oder existiert nicht	
3a1	Mitarbeiter	Der Mitarbeiter bricht den Auswahldialog ab in dem der die "Abbrechen" Option wählt	
3a2	Smartphone	Der Dialog wird geschlossen und es wird zum Fenster "Maschinen Einstellungen" zurück navigiert.	
Nachbedingung:		Der Dialog ist geschlossen Der Mitarbeiter findet sich im "Maschinen Einstellungen" Fenster wieder Es wurde keine Modifikation an der globalen Variablen Liste vorgenommen	
Alternativablauf 6a			
Vorbedingung:		Die ausgewählte Datei ist keine valide XML-Datei	

6a1	Smartphone	Der Fortschritts Dialog wird geschlossen und ein "Toast" Dialog angezeigt der mitteilt, dass die Datei kein korrektes Format besitzt.
6a2	Mitarbeiter	Der Mitarbeiter wählt die "OK" Option des "Toast" Dialogs um diesen zu schließen
6a3	Smartphone	Der "Toast" Dialog wird geschlossen und es wird zum "Maschinen Einstellungen" Fenster zurück navigiert.
Nachbedingung:		Der Mitarbeiter ist informiert, dass die Ziel Datei kein korrektes Format besitzt Der Mitarbeiter findet sich im "Maschinen Einstellungen" Fenster wieder

11.1.16 *EVERGREEN SERVER DER SERVERLISTE HINZUFÜGEN*

Um einen bestimmten Server bei der Anmeldung auswählen zu können, wird über diesen Anwendungsfall ermöglicht diesen in die Auswahl aufzunehmen.

Ziel	Der Mitarbeiter möchte einen Evergreen Server zu der Serverliste hinzufügen um diesen eventuell später für eine Anmeldung zu nutzen.	
Akteure	Mitarbeiter	
Beschreibung:		
Ebene	Benutzersicht	
Priorität	Hoch	
Normalablauf		
Vorbedingung:	Der Mitarbeiter befindet sich im "Anmelde" Fenster	
1	Mitarbeiter	Der Mitarbeiter wählt die "Server hinzufügen" Option aus
2	Smartphone	Es wird der "Server hinzufügen" Dialog angezeigt.
3	Mitarbeiter	Der Mitarbeiter gibt die Serveradresse, dessen Port und einen Namen, unter dem er wiedererkannt wird, ein.
4	Mitarbeiter	Der Mitarbeiter wählt die "hinzufügen" Option des "Server hinzufügen" Dialoges aus.

5	Smartphone	Der Dialog wird geschlossen, die Serverinformationen werden in die Umgebungsliste für Server hinzugefügt und die Auswahl für vorhandene Evergreen Server wird aktualisiert
Nachbedingung:		Der Mitarbeiter befindet sich im "Anmelden" Fenster Der Server befindet sich in der Umgebungsliste für Server Der Server steht zur Auswahl innerhalb des "Anmelden" Fensters

11.1.17 *EVERGREEN SERVER AUS DER SERVERLISTE ENTFERNEN*

Wird ein Server nicht mehr benötigt, weil er eventuell nicht mehr existiert, kann er über diesen Anwendungsfall aus der Serverliste entfernt werden. Dabei werden alle Einstellungen für diesen Server verworfen.

Ziel		Der Mitarbeiter möchte einen Server aus der Serverliste entfernen
Akteure		Mitarbeiter
Beschreibung:		
Ebene		Benutzersicht
Priorität		Niedrig
Normalablauf		
Vorbedingung:		Der Mitarbeiter befindet sich im "Anmelden" Fenster
1	Mitarbeiter	Der Mitarbeiter wählt die "Server entfernen" Option aus.
2	Smartphone	Es wird der "Server entfernen" Dialog angezeigt.
3	Mitarbeiter	Der Mitarbeiter wählt den Ziel Server aus und wählt die "entfernen" Option aus
4	Smartphone	Der Server wird aus der Umgebungsliste für Server gelöscht und die Anzeige der Server innerhalb des "Anmelde" Fensters wird aktualisiert.
5	Smartphone	Der Dialog wird geschlossen
Nachbedingung:		Der Server wurde aus der Umgebungsliste für Server, sowie aus der Auswahl innerhalb des "Anmelde" Fensters entfernt

11.1.18 ANMELDUNG WIEDERHOLEN

Wird während der Nutzung eine genutzte Session ungültig erfolgt während der Hintergrundaktualisierung ein Fehler welcher einen Fehlerdialog erzeugt. Dieser Dialog ermöglicht es eine neue gültige Session zu erwerben um in dem aktuellen Zustand der Anwendung fortzufahren.

Ziel		Eine neue gültige Session zu erhalten ist das Ziel.	
Akteure		Mitarbeiter, Evergreen-Server	
Beschreibung:		Durch einen Request an den Evergreen Server wurde der Applikation mitgeteilt, dass die verwendete Session nicht mehr gültig ist. Wenn dies passiert werden keine weiteren Requests an den Server geschickt bis wieder eine gültige Session existiert.	
Ebene		Benutzersicht	
Priorität		Mittel	
Normalablauf			
Vorbedingung:		Die aktuelle Session ist ungültig. Es gab eine Fehlermeldung aus einem Hintergrundprozess.	
1	Smartphone	Es wird ein Fehlerevent ausgelöst das dem aktiven Fenster mitteilt, dass keine gültige Session mehr vorliegt.	
2	Smartphone	Im aktiven Fenster wird ein "Anmeldung wiederholen" Dialog angezeigt	
3	Mitarbeiter	Der Mitarbeiter gibt seine Anmeldedaten ein, um eine neue gültige Session zu erhalten. Er wählt die "anmelden" Option des Dialoges aus.	
		Bedingung für Sonderfall Der Mitarbeiter kann oder will keine Anmeldung.	Alternativablauf 3a
4	Smartphone	Es wird ein "Progress" Dialog angezeigt, der dem Mitarbeiter mitteilt, dass eine Anfrage an den Server gestellt wird.	
5	Smartphone	Mit den Anmelde Daten wird eine Anfrage an den Evergreen Server gestellt.	
		Bedingung für Sonderfall Der Server ist nicht erreichbar.	Alternativablauf 5a
6	Evergreen-Server	Empfängt die Anmeldedaten des Mitarbeiters und gleicht diese mit den validen Anmeldedaten ab. Ist das Passwort korrekt wird eine neue gültige Session übergeben.	

		Bedingung für Sonderfall Die Anmeldedaten sind nicht korrekt	Alternativablauf 6a
7	Smartphone	Die gültige Session ersetzt die alte Session in der Umgebung.	
8	Smartphone	Der "Anmeldung wiederholen" Dialog wird geschlossen.	
Nachbedingung:		Es existiert eine gültige Session	
Alternativablauf 3a			
Vorbedingung:		Der Mitarbeiter kann oder will keine Anmeldung.	
3a1	Mitarbeiter	Der Mitarbeiter wählt die "verlassen" Option des "Anmeldung wiederholen" Dialoges.	
3a2	Smartphone	Alle Fenster werden geschlossen und die Applikation geschlossen.	
Nachbedingung:		Die Applikation ist beendet	
Alternativablauf 5a			
Vorbedingung:		Der Server ist nicht erreichbar.	
5a1	Smartphone	Der "Progress" Dialog wird geschlossen und es wird ein Fehler Dialog angezeigt der dem Mitarbeiter mitteilt, dass der Evergreen Server gerade nicht erreichbar sei und er bei einer bestehenden Verbindung diesen Vorgang wiederholen soll.	
5a2	Mitarbeiter	Der Mitarbeiter beseitigt die Ursache für die Verbindungsprobleme und wählt die "wiederholen" Option des Fehler Dialoges.	
		Bedingung für Sonderfall Der Mitarbeiter ist nicht in der Lage die Fehlerursache zu beseitigen	Alternativablauf 5a2a
5a3	Smartphone	Es wird ein "Progress" Dialog angezeigt, während die Anmeldedaten des Mitarbeiters an den Evergreen Server geschickt werden.	
5a4	Evergreen-Server	Empfängt die Anmeldedaten des Mitarbeiters und gleicht diese mit den validen Anmeldedaten ab. Ist das Passwort korrekt wird eine neue gültige Session übergeben.	
		Bedingung für Sonderfall Die Anmeldedaten sind nicht korrekt	Alternativablauf 5a4a

5a5	Smartphone	Die gültige Session ersetzt die alte Session in der Umgebung.
5a6	Smartphone	Der "Anmeldung wiederholen" Dialog wird geschlossen.
Nachbedingung:		Es existiert eine gültige Session
Alternativablauf 5a2a		
Vorbedingung:		Der Mitarbeiter ist nicht in der Lage die Fehlerursache zu beseitigen
5a2a1	Mitarbeiter	Der Mitarbeiter wählt die "abbrechen" Option des Fehler Dialoges.
5a2a2	Smartphone	Der Fehler Dialog wird geschlossen
5a2a3	Mitarbeiter	Der Mitarbeiter wählt die "verlassen" Option des "Anmeldung wiederholen" Dialoges
5a2a4	Smartphone	Alle Fenster werden geschlossen und die Applikation geschlossen.
Nachbedingung:		Die Applikation ist beendet.
Alternativablauf 5a4a		
Vorbedingung:		Die Anmeldedaten sind nicht korrekt
5a4a1	Evergreen-Server	Der Evergreen Server sendet einen Fehler zurück
5a4a2	Smartphone	Der Fehler wird Empfangen und es wird eine Warnung im "Anmeldung wiederholen" Fenster eingeblendet die den Mitarbeiter darauf hinweist, dass seine Daten inkorrekt waren.
5a4a3	Smartphone	Der "Progress" Dialog wird geschlossen.
Nachbedingung:		Der Mitarbeiter weiß, dass seine Eingabe falsch war. Der "Anmeldung wiederholen" Dialog wird angezeigt.
Alternativablauf 6a		
Vorbedingung:		Die Anmeldedaten sind nicht korrekt

6a1	Evergreen-Server	Der Evergreen Server sendet einen Fehler zurück
6a2	Smartphone	Der Fehler wird Empfangen und es wird eine Warnung im "Anmeldung wiederholen" Fenster eingeblendet die den Mitarbeiter darauf hinweist, dass seine Daten inkorrekt waren.
6a3	Smartphone	Der "Progress" Dialog wird geschlossen.
Nachbedingung:		Der Mitarbeiter weiß, dass seine Eingabe falsch war. Der "Anmeldung wiederholen" Dialog wird angezeigt.

ICH VERSICHERE, DIESE ARBEIT SELBSTSTÄNDIG VERFASST ZU HABEN.

ICH HABE KEINE ANDEREN ALS DIE ANGEgebenEN QUELLEN BENUTZT UND ALLE WÖRTLICH ODER SINNGEMÄß AUS ANDEREN WERKEN ÜBERNOMMENE AUSSAGEN ALS SOLCHE GEKENNZEICHNET.

WEDER DIESE ARBEIT NOCH WESENTLICHE TEILE DARAUS WAREN BISHER GEGENSTAND EINES ANDEREN PRÜFUNGSVERFAHRENS.

ICH HABE DIESE ARBEIT BISHER WEDER TEILWEISE NOCH VOLLSTÄNDIG VERÖFFENTLICHT.

DAS ELEKTRONISCHE EXEMPLAR STIMMT MIT ALLEN EINGEREICHTEN EXEMPLAREN ÜBEREIN.

(DATUM, ORT)

(UNTERSCHRIFT)