Alarm Clock

Jonathan Hyams
Pascal Schmalz

27. Mai 2017

Inhaltsverzeichnis

Zweck des Dokument Dieses Dokument soll dazu dienen, die Anforderungen an das Projekt AlarmClock zu definieren. Typische Leser sind die Entwickler und der Dozent im Requirements Engineering. Auch für den Projektbetreuer kann das Dokument von Interesse sein.

1 Kurzbeschreibung

Das Ziel des Projektes ist einen Ersatz zum Programm kAlarm zu entwickeln. Das Programm kAlarm erlaubt es dem User benutzerdefinierte Erinnerungen zu erstellen. Mittels Pop-Up Windows wird der User dann zur gegebenen Zeit daran erinnert. Im Gegensatz zu kAlarm soll das zu erstellende Produkt platformübergreifend verfügbar sein. Wie kAlarm soll dieses Produkt unter einer Open Source Lizenz entwickelt werden.

2 Projektziele

Unser Auftraggeber ist kein Unternehmen. Es werden somit auch keine Ziele innerhalb eines Unternehmens verfolgt. Ziele, welche ein Benutzer unsere Aplikation verfolgen könnte wären beispielsweise:

- An den Wäschetag erinnert weden.
- Die Wäsche rechtzeitig aus der Wäscheküche holen.
- Den Kuchen nicht im Backofen verbrennen lassen.
- Der Sekretärin zum Geburtstag Blumen schicken.
- usw.

3 Stakeholders

- Auftragsgeber: Prof. Claude Furrer
- Auftragsnehmer: Jonathan Hyams, Pascal Schmalz
- Benutzer: FOSS Communitiy

4 Systemabgrenzung

Es exisitert kein vordefiniertes System, welches mit unserem Programm interagiert. Das Programm soll auf einem Computer laufen, welcher Java 8.1 inklusive JavaFx installiert hat.

4.1 Geschäftsprozesse

Es existieren keine vor oder nachgelagerte Prozesse. Unsere Lösung lässt sich in beliebig viele Prozesse integrieren.

4.2 Systeme

Unser System beinhaltet einen Persistenz Layer um nach einem Neustart den vorherigen Zustand wiederherstellen zu können. Die Persistenz soll an einem zentralen Punkt sichergestellt werden, so dass mit wenig Aufwand eine andere Persistenzmethode, wie eine JPA Datenbank eingebunden werden kann.

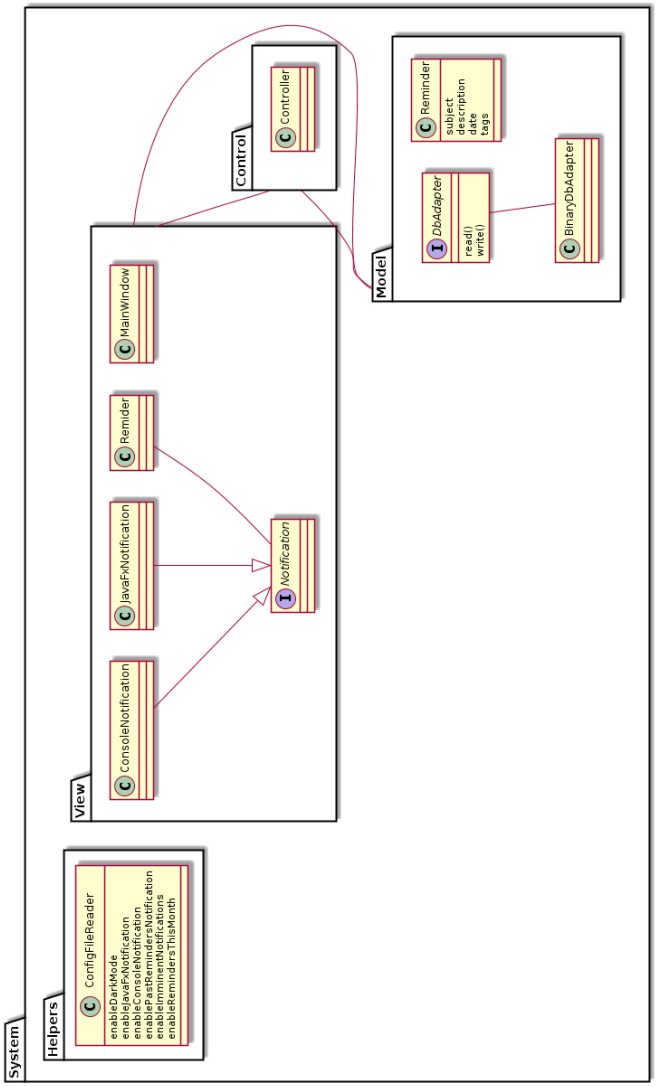

Abbildung 1: Systemübersicht und Abgrenzung

Die Systemübersicht und Abgrenzung Abb. 1 Seite 3 bedient sich der UML Notation, ohne die UML Spezifikation vollständig einzuhalten. Die Komponenten innerhalb des Kasten “System”, insbesondere ihre Beziehung zueinander, sind nicht Teil der Anforderungen, und somit nicht bindend. Sie veranschaulichen lediglich was noch innerhalb des Systems liegt und was ausserhalb angesiedelt ist. Abbildung:3

Falls die Funktionalität vom Auftraggeber verlangt wird, muss das Programm den Nutzer mit E-Mails, SMS und Systemnotifications erinnern. Dies bedingt, dass entsprechende Services / Server eingerichtet und über eine definierte Schnittstelle erreichbar sind, was nicht mehr Teil des Projektes ist.

4.3 Randbedingungen

Die Entwicklung soll mittels Java und JavaFX erfolgen. Das Programm soll Betriebssystem und Datenbank agnostisch sein. Das Projekt muss unter einer Opensource Lizenz veröffentlicht werden. Die Dokumentation muss in LaTeX erstellt werden.

5 Anforderungen

5.1 Basisfaktoren

- Die Schnittstelle zwischen dem Programmcode und der Datenbank soll so gestaltet sein, dass mit einem minimalen Aufwand eine beliebige gängige Datenabank genutzt werden kann.
- Die Bereitstellung eines JPA Drivers der Datenbank ist dabei nicht mehr im Scope des Projektes, sondern wird von den Datenbankherstellern oder von Dritten vorgenommen.
- Das System muss dem Benutzer die Möglichkeit bieten, in dem GUI ein Einzelevent zu erfassen.
- Das System sollte dem Benutzer die Möglichkeit bieten, über das CLI wiederkehrende Events zu erfassen.
- Das System kann anderen Programmen über eine API ermöglichen, Einzelevents zu erfassen.
- Das Löschen und Anpassen von Events soll wie oben beschrieben ebenfalls möglich sein.
- Das System muss den Benutzer mittels einer Notification rechtzeitig, zur voreingestellten Zeit, auf ein Event hinweisen.
- Das Programm muss an einem Zentralen Ort konfiguriert werden.
- Das Programm kann dem Benutzer die Möglichkeit bieten, die Konfiguration über die GUI vorzunehmen und im Configfile zu speichern.

5.2 Leistungsfaktoren

- Der Benutzer soll neue Kategorien hinzufügen, anpassen und löschen können.
- Events sollen durch den Benutzer in Kategorien eingeteilt werden können.
- Der Benutzer soll Events nach Kategorien gruppieren und filtern können.
- Das Programm soll vorkonfigurierte Dienste nutzen können um seine Notifications als SMS und Emails zu versenden.
- Das Programm soll vorkonfigurierte Dienste nutzen können um anstelle einer Notification ein Script laufen zu lassen.
- Die GUI soll benutzerzentriert und ergonomisch sein (Wie kann man dies sinvoll als Anforderung formulieren?)

5.3 Begeisterungsfaktoren

- Das Programm soll vorkonfigurierte Dienste nutzen können um anstelle einer Notification ein IFTTT Event auszulösen.
- Das Programm soll zusätzlich zu den gängigen Computer-Betriebssystemen auch auf Android lauffähig sein.
- Events sollen zwischen allen registrieren Geräten eines registrierten Benutzers synchronisiert werden. Dazu muss der Benutzer in der Lage sein, sich und seine Geräte, welche das Programm installiert haben, zu registrieren.
- Das Programm soll auf den Betriebssystemen, welche eine Notificationinfrastruktur bereit zu stellen in der Lage sein die Notifications über diese Notificationinfrastruktur vorzunehmen.(BSP KDE)
- Das Programm soll sich vom Look and Feel in das jeweilige Desktop Environment anpassen.
- Das System kann dem Benutzer die Möglichkeit bieten, in dem GUI wiederkehrende Events zu erfassen.

5.4 Quellen und Vorgehen

Unsere wichtigste Quelle ist unser Auftraggeber Prof. Furrer. Um die Fragen, welche bei der Erstellung der Anforderungen aufgetaucht sind zu klären, werden wir uns mit ihm treffen. Besondere folgende Fragen gilt es abzuklären:

- Entspricht die bisherige Ausarbeitung der Vorstellung des Auftraggebers.
- Vollständigkeit der Anforderungen.

- Kategoriserung der Anforderungen.
- Priorisierung der Anforderungen.
- Was soll mit der Eventkategorisierung erreicht werden, welche weiteren Anforderungen ergeben sich daraus.
- Abgrenzung des Programmes gegenüber eines kAlarm, Kalenders und gegenüber des Cronjobs.

Da unser Auftraggeber als Professor der Informatik, als weitere Diskusionsgrundlage sind wir daran neben diesem Dokument einen Prototyp des GUI zu erstellen.

Die angegebenen Informationen, insbesonder die abzuklärenden Punkte, sind dementsprechend lediglich eine Work in Progress.

Als weitere Quellen können wir die Dokumentation und Code Basis von kAlarm nutzen. Da wir selber der potentiellen Nutzergruppe des Programms angehören, könnten wir auch uns selber als Quelle nutzen, als Entwickler hat man jedoch oft einen anderen Blickwinkel als ein normaler Benutzer.

Im Sinne einer agilen Entwicklung wollen wir möglichst rasch präsentierbare Artefakte generieren, damit der Auftraggeber frühstmöglich Einfluss nehmen kann und Fehlentwicklungen vermieden werden. Treffen werden bei Bedarf mit dem Auftraggeber ausgemacht.

5.5 Technische Anforderungen

TODO redundanz wirklich gewollt?

- Das Programm muss auf Linux, Windows und OSx lauffähig sein.
- Das Programm muss datenbankagnostisch sein.
- Das Programm muss auf dem Buildsystem kompiliert werden können.
- Das Programm muss auf dem Referenzsystem ohne weitere installationen kompiliert werden können.

5.6 Qualitätsanforderungen

- 80% der Benutzer sollen in der Lage sein, ohne vorgänige Schulung, bei der ersten Benutzung, folgende Aktionen jeweils innert 5 Minuten nach erfolgtem Programmstart, auszuführen:
  - Erstellen eines Events.
  - Löschen eines vorhandenen Events.
  - Editieren eines Events (Zeit, Datum, Notificationtext).
- 60% der Benutzer solle in der Lage sein, die Konfiguration des Programmes mittels ConfigFiles Vorzunhemen.

6 Glossar

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

Android TODO. 8

API TODO Wikipedia. 8

Benutzer TODO Ein Benutzer ist ein Mensch, welcher unser Programm benutzt, falls nicht anders erwähnt, muss dieser weder registriert noch angemeldet sein. 8

Buildystem Linux Mint 17,Java JDK 1.8.1, Intellij 2016.3.5, JavaFx Scene Builder 2.0. 8

CLI TODO Wikipedia. 8

Cronjob TODO. 8

Cronjob TODO. 8

Desktop Environment TODO. 8

Einzelevent TODO Ein Einzelevent ist ein Event, welches nicht periodisch wiederkehrend stattfindet, also können auch ein Treffen, welches unregelmässig stattfindet zu einem Einzelevent werden.. 8

Event TODO Ein Event ist ein Ereignis, welches zu einem bestimmten Zeitpunkt stattfindet. 8

Filtern TODO. 8

Gruppieren TODO. 8

GUI TODO Wikipdeia. 8

IFTTT IFTTT (die Abkürzung von If This Then That, ausgesprochen „ift“ wie in „Gift“[1]) ist ein Dienstanbieter, der es Benutzern erlaubt, verschiedene Webanwendungen (zum Beispiel Facebook, Evernote, Dropbox usw.) mit einfachen bedingten Anweisungen zu verknüpfen. Wikipedia. 8

JPA Driver TODO. 8

Kategorien TODO . 8

Konfiguration Die Konfiguration kann ein Programm auf die Bedürfnisse des Nutzers anpassen. So kann man beispielsweise die Spracheinstellung konfigurieren.. 8

Notification TODO Eine Notification ist eine Aktion des Programms, die den Benutzer auf ein Event aufmerksam machen soll.. 8

Notification Infrastruktur TODO. 8

Programmstart TODO. 8

Referenzsystem Linux Mint 17,Java JRE 1.8.1,Derby10.13.1.1. 8

Registrierter Benutzer TODO. 8

schtask TODO. 8

Standarddatenbank Derby,MySQL,SQLite. 8

7 Anhang

7.1 Abstimmung der Anforderungen

7.2 Definition of Ready - Checklist

8 Versionskontrolle

Manuelle Version: 0.0.2

Automatische Versionierung:
Last compiled: Sat 27 May 18:04:13 CEST 2017
Git HEAD Version: 132