

Schnittstelle von Patientenverwaltung zu Diagnostikgeräten

Christian Franke, Betreuerin: Kerstin Denecke, Living Case 1, Medizininformatik

Erstellt am 11.12.2042

Einleitung

Dieses Projekt wurde im Auftrag der Eschmann Contactlinsen AG durchgeführt. Das kleine Unternehmen mit Sitz in Bern bietet optometrische Beratung, Sehtests, Brillenglas-Bestimmung und Kontaktlinsen-Anpassungen an. Dafür werden drei Diagnostik-Geräte verwendet, bei denen die Patientenstammdaten bisher manuell eingegeben werden mussten.

Ziel des Projektes war die Umsetzung von Schnittstellen, mit denen Patientenstammdaten vom primären Kundenverwaltungsprogramm an die Diagnostikgeräte übertragen werden. Dadurch sollten repetitive, manuelle Aufgaben reduziert und potentielle Fehlerquellen vermieden werden.

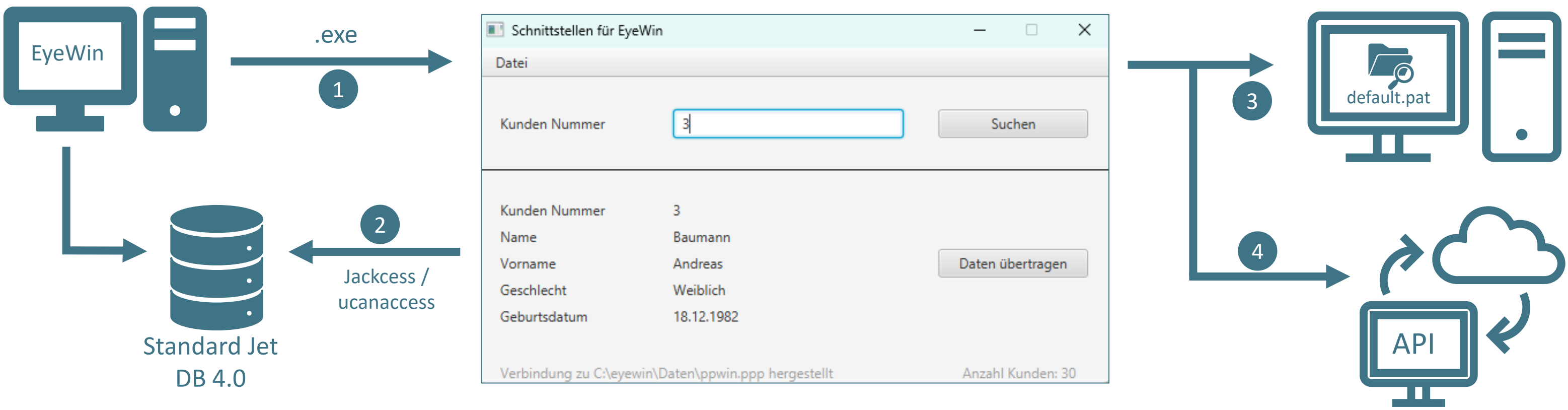
Methode

Es wurden vier Hauptarbeitspakete definiert: der Export der Patientenstammdaten aus der Standard Jet Datenbank des Primärsystems EyeWin sowie drei Importe zu den drei Diagnostik-Geräten von drei unterschiedlichen Herstellern.

Für alle vier Hauptarbeitspakete wurde die Produktdokumentation analysiert sowie Kontakt mit den Herstellern für weiterführende Informationen aufgenommen. Drei der vier Hauptarbeitspakete basieren auf unterschiedlichen Technologien und ein Hersteller hat mitgeteilt, dass keine Importmöglichkeit bestehen würde. Somit musste eine Schnittstelle bereits zu Projektbeginn verworfen werden.

Ergebnisse

Die Java-FX-Applikation wurde in IntelliJ und dem Maven Framework umgesetzt. Für den Export aus der Patientendatenbank wurden die Dependencies Jackcess und ucanaccess genutzt. Die GUI wurde mit dem Scene Builder 2.0 entwickelt. Die Erstellung der exe-Datei wurde mit Lunch4j realisiert. Das Programm mit zehn Klassen wurde auf dem eigenen Rechner entwickelt und innerhalb der Infrastruktur des Auftraggebers getestet.



Nach Eingabe der Patientenstammdaten im EyeWin öffnen die Anwenderinnen die Applikation (1). Sie suchen eine Kundennummer (2) und übertragen die Daten in eine spezifische default.pat Datei im internen Netzwerk (3). Die Patientenverwaltung des ersten Diagnostik-Systems liest die Datei und importiert die Kundendaten. Für das zweite Diagnostik-Gerät wurde eine RESTfulAPI Schnittstelle in einer Test-Umgebung realisiert (4). Die Lizenzkosten des Herstellers für die WebAPI sind, trotz häufigen Nachfragens bei den Vertriebspartnern, unsicher. Vermutlich übersteigen die Kosten auch den Nutzen um ein Vielfaches, sodass diese Schnittstelle derzeit sistiert wurde.

Diskussion

Die gründliche Projektplanung war zwar hilfreich, dennoch sind unerwartete Herausforderungen aufgetaucht. Dazu zählt das Hauptarbeitspaket für den Export aus der Standard Jet Datenbank. Zugriff und die Plausibilisierung waren nicht mit MS Access-Treibern möglich, sodass andere Lösungen gefunden werden mussten. Zudem war die Datenbank passwortgeschützt, aber es gibt Freeware zum Auslesen des Passworts aus einer Jet DB. Zudem wäre ein weiteres Hauptarbeitspaket für die Erstellung einer lauffähigen exe-Datei sinnvoll gewesen. Dieses Thema wurde im Rahmen des Export-Arbeitspaketes bearbeitet, was zum Teil verwirrend war.

<https://hapi.fhir.org/baseR5/Patient?name=Franke>

Über RESTful API hatte ich zu Beginn es Projektes sehr wenig gewusst und musste viel Zeit in Online-Tutorials investieren. Die API-Schnittstellen-Tests wurden letztlich mit Postman durchgeführt (siehe Abb. unten).

```
POST https://hapi.fhir.org/baseR5/Patient
Body:
{
  "resourceType": "Patient",
  "identifier": [
    {
      "use": "official",
      "value": "3012Bern84191603"
    }
  ],
  "name": [
    {
      "use": "official",
      "family": "Franke",
      "given": ["Christian"]
    }
  ],
  "gender": "male",
  "birthDate": "1984-03-16"
}
```

Die «Challenge: 30 days of Postman for developers!» habe ich noch nicht komplett beendet, läuft sie aber noch und ich kann sie empfehlen.

Die Videos von Sidharth Ramesh haben mir praktische Anwendungen von FHIR näher gebracht. Ein von ihm genutzter HAPI-FHIR-Test-Server bietet API-Endpunkte für diverse FHIR-Versionen an. Die Entwicklung der FHIR-Ressource «Patient» (siehe Screenshot) fand mit VS Code statt. Die erstellte FHIR-Test-Ressource könnte ich für die effektive WebAPI-Schnittstelle des zweiten Diagnostik-Gerätes ebenfalls nutzen, falls eine Lizenz gekauft werden würde. Die Anpassung des Endpunkts und des JSON-Request-Bodys wäre mit wenig Aufwand, aber vielen Test möglich.

Fazit

Die technischen Herausforderungen sind das Eine, die organisatorischen Aufgaben etwas ganz Anderes! Neben meinen Auftraggebern sind die Anwenderinnen zwar die wichtigsten Stakeholder. Aber die weiteren Personen und Dienstleister haben einen anderen Fokus sowie eigene Interessen, die man mit dem Projekt in Einklang bringen muss. Für ein erfolgreiches Projekt benötigt es nicht nur Technik, sondern auch viel Kommunikation und Austausch mit weiteren Beteiligten. Und häufig spielen ökonomischen Anreize am Ende eine zentrale Rolle, wieso etwas getan oder gelassen wird. Als Projektleiter müssen wir auch diese organisatorische Ebene im Blick haben.

Es konnten zwar nicht alle Schnittstellen umgesetzt werden, aber die «wichtigste» Schnittstelle funktioniert. Nach dem ersten Release und den ersten Nutzungen berichteten mir die Anwenderinnen glücklich und mit einem Lächeln im Gesicht von Ihren positiven Erfahrungen. Das war für mich das Wichtigste!

Literatur

<https://teletrabbie.github.io/livingcase01/eyewininterfaces.html>
<https://www.postman.com/postman/workspace/30-days-of-postman-for-developers/overview>
<https://www.youtube.com/@sidharthramesh1892>
<https://hapi.fhir.org>