Storyboard PSE Demo Video 1

Shot #01

Thema

Intro

Dauer

10 / 10 Sekunden

Inhalt /Beschreibung

Intro mit Logo von eonum, Projekt PSE mit allen Namen

Off-Stimmt

-

Shot #02

Thema

Was sind Handbücher

Dauer

60 / 70 Sekunden

Inhalt /Beschreibung

Einleitung in Thema Handbücher. Erklärung von Codeverwendung. Man sieht Person in Spital gehen mit Verletzung. Während im Spital werden Behandlungsschritte als Text auf Liste gesetzt. Text verwandelt sich in Codes. Summe wird berechnet und Preis erscheint. Person verlässt verarztet wieder Krankenhaus.

Off-Stimmt

Im Gesundheitswesen, spezifisch in den Spitälern, werden alle am Patienten ausgeführte Behandlungsschritte aufgezeichnet. Als Beispiel nehmen wir Herr M, welcher ein neues Kniegelenk benötigt. Während seines Aufenthaltes werden alle ausgeführten Schritte vom Fachpersonal festgehalten. Damit diese Listen nicht immens lang werden, gibt es für jeden Behandlungsschritt ein spezifischen Code, welcher das ganze zusammenfasst. Die Behandlungsschritte werden von Kodierer*innen mithilfe von Software, sogenanten Grouper-Software, in vierstellige Codes umgewandelt. Der Code für ein neues Kniegelenk ist hier nun I43B. Jede Code hat ein eigenes Kostengewicht. Die Summe der Kostengewichte wird dann mal den Basispreis gerechnet, welchen die Spitäler jährlich mit den Krankenkassen vereinbaren. So wird der Preis des Spitalaufenthaltes und der dazugehörigen Behandlung berechnet. Herr M erhält nun eine Rechnung von 9‘500.-

Shot #03

Thema

Was sind Kodierer*innen

Dauer

25 / 95 Sekunden

Inhalt /Beschreibung

Stock-Footage von durchblättern von Büchern. Beispiel zeigen von einem konkreten Code und deren Feinheiten.

Off-Stimmt

Das Nachschlagen dieser Codes ist eine mühselige, aufwändige und komplizierte Arbeit, für welche es ein eigenen Job gibt. Die sogenannten medizinische Kodierer*innen. Dabei kommt es auf Feinheiten drauf an und benötigt ein grosses Fachwissen.
Hier zum Beispiel sieht man, wie solch ein Code noch in Untercodes unterschieden wird.

Shot #04

Thema

Was sind Kodierer*innen

Dauer

25 / 120 Sekunden

Inhalt /Beschreibung

Alle Handbücher zeigen, plus Seitenzahlen. Scrollen auf Seite oder Grafik.

Off-Stimmt

Nebst allen Handbüchern, von welchen regelmässig neue Versionen erscheinen, mit den jeweilgen Codes (ICDs, CHOPs und DRG) kommt auch mehrmals jährlich ein Update des medizinischen Kodierhandbuchs heraus. Die grosse Anzahl an verschiedenen Handbüchern mit einer Vielzahl von Versionen kann das Nachschlagen für die Kodierer*inne sehr unübersichtlich gestalten.

Shot #05

Thema

Was ist Medcodesearch

Dauer

30 / 150 Sekunden

Inhalt /Beschreibung

Logo von Medcodesearch einblenden. In den Versionen herum klicken und auf jeden Katalog klicken. Ein spezifischer Begriff suchen und die Handhabung der Suchresultate aufzeigen.

Off-Stimmt

Hierbei kommt medcodesearch ins Spiel. Medcodesearch ist eine Suchmaschiene, hauptsächlich für Kodierer*innen, und dient

als digitales Nachschlagewerk für all die Handbücher. Dabei sind alle Versionen enthalten und können kinderleicht durchsucht werden. Dies erleichtet den Arbeitsalltag der Kodiererinnen indem sie nicht mehrer PDFs von Hand durchsuchen müssen. Mit einer simplen Stichwortsuche oder mithlfe der jeweiligen Codes finden sie das Gesuchte blitzschnell.

Shot #06

Thema

Zukunft von Medcodesearch

Dauer

25 / 175 Sekunden

Inhalt /Beschreibung

Durchscrollen von de KLV's und Kodierhandbüchern

Off-Stimmt

Bald schon werden zwei weiterer Meilensteine erreicht sein. Einerseits wird man bei einer Suchanfrage gleich noch über die gesetzlichen Hintergründe mitinformiert. Dies bedeutet, man sieht ob die Krankenkasse die gesuchte Behandlung bezahlt oder nicht. Andererseits können die Kodierer*innen das Änderungshandbuch durchsuchen, welches bislang noch nicht möglich war.

Shot #07

Thema

Wie funktioniert Medcodesearch

Dauer

10 / 185 Sekunden

Inhalt /Beschreibung

Suchbegriff wird eingegeben, Maus wird auf den Such Button bewegt und bei Klick pausiert man das Ganze.

Off-Stimmt

Über ein schlankes Frontend kann ein Benutzer eine Suchanfrage starten und kriegt prompt eine Antwort, doch was passiert eigentlich im Hintergrund?

Shot #08

Thema

Wie funktioniert Medcodesearch

Dauer

45 / 230 Sekunden

Inhalt /Beschreibung

Man sieht wie man herumklickt auf den Seiten und die jeweiligen PDFs manuell herunterlädt. Big picture: Man schraubt am Crawler, welcher dann von Cronjob angestupst wird. Dieser Geht über mehrere Seiten und sendet PDFs in Minenkarre zurück. Am Ende kommt er wieder zurück.

Off-Stimmt

Die Grundlage besteht aus allen Handbüchern und deren verschiedenen Versionen. Bislang wurden diese manuell heruntergeladen von eonum, doch dies war ein sehr mühseliger Prozess, welchen man automatisieren kann. Wir arbeitet nun an einem Webcrawler, welche diese automatisch aus den verschiedenen Quellen holt. Er wird dann regelmässig via Cronjob angestossen, damit er mit seiner Arbeit beginnt. Dabei überprüft der Crawler, welche Versionen er herunterladen muss und macht dies jeweils für alle verfügbaren Sprachen. Meistens ist dies Deutsch, Französisch und Italienisch. Die PDFs speichert er dann auf dem Server ab. Von dort aus übernimmt der Parser sein Arbeit.

Shot #09

Thema

Wie funktioniert Medcodesearch

Dauer

45 / 275 Sekunden

Inhalt /Beschreibung

Man sieht den unterschied zwischen strukturierten und unstrukturierten Dokumenten. Big picture: Die PDFs im Minenkarren fahren nun zum Parser rüber, wo sie reingekippt werden. Unten kommen Ruby-Struct-Objekte raus, welche in einem weiteren Minenkarren landen.

Off-Stimmt

Die bestehend eingebundene Handbücher sind als schön strukturierte Tabellen verfügbar, doch die von uns neu hinzuzufügenden Dokumente sind alle mehr oder weniger Unstrukturiert. Dies erschwert das Parsen der Dokumente, da man viele Ausnahmen und verschiedene Fälle im Parser berücksichtigen muss. Der Parser wandelt den Inhalt der PDFs in Ruby Klassen um via struct-Objekte, welche man auch aus der C-Programmierung kennt. Der Text aus dem PDF wird in eigene Objekte umgebaut, wobei jede Sprache noch in einem eigenen Objekt ist. Diese werden dann in ein Objekt gemerged, damit daraus vollständige Objekte entstehen.

Shot #10

Thema

Wie funktioniert Medcodesearch

Dauer

20 / 295 Sekunden

Inhalt /Beschreibung

Die Minenkarre fährt zur Datenbank, wo die Objekte in ein Präprozessor gegeben werden, welcher Modelobjekte daraus macht und dann die Datenbank befüllt

Off-Stimmt

Die Objekte werden nun umgemünzt in Modelobjekte. Diese können von Ruby on Rails nun praktisch in die PostgreSQL Datenbank eingespiesen werden. Dank den praktischen Schnittstellen in Ruby on Rails kann man direkt mit der Datenbank interagieren ohne grosse Umwege.

Shot #11

Thema

Wie funktioniert Medcodesearch

Dauer

60 / 355 Sekunden

Inhalt /Beschreibung

Es wird ein ElasticSearch Knopf gedrückt. Die Datenbank schüttelt sich eine Schublade geht auf, wo man Indexe sieht.

Off-Stimmt

Damit die grosse Menge and Text sehr effizient durchsucht werden kann, verwenden wir ElasticSearch. Mit ElasticSearch kann man eine bestehende Datenbank benutzerdefinit analysieren lassen. Dabei zerstückelt es Worte und Codes in kleine Schnipsel und Indexiert so den gesamten Datenbank Inhalt. Dies ermöglicht es, eine grosse Menge an Fliesstext in der Datenbank zu durchsuchen und sehr rasch eine Antwort zu erhalten. Wenn man mit einem normalen SQL-Query eine Fliesstextsuche ausführen würde ohne alles Indexiert zu haben, könnte man noch ein Kaffee trinken, bis eine Antwort kommt. Damit die Suche noch schneller ist, wurde eine Art Smart Searching implementiert. Dabei wird im Backend erkannt, ob es sich um eine Fliesstextsuche handelt oder nur ein Code gesucht wird. Entsprechend der Suchanfrage wird der Request anders behandelt und das Backend ist noch einmal schneller mit Antworten.

Shot #12

Thema

Wie funktioniert Medcodesearch

Dauer

40 / 395 Sekunden

Inhalt /Beschreibung

Man sieht die Suchanfrage von vorhin, wie sie via Angular an die API gebracht wird. Diese wird im Minenkarren zur Datenbank gebracht und dort werden die Antworten aus der Indexierten Schublade genommen. Diese werden in den Minenkarren gelegt und zurückgeschickt. Die API übergibt die Antwort an Angular.

Off-Stimmt

Die getätigte Suchanfrage von vorhin wird vom Frontend, welches auf Angular basiert, vorverarbeitet. Diese Anfrage wird dann an ein API-Endpunkt gesendet, welcher nun das Such-Query mit Ruby Controllern spezifisch auf ElasticSearch zuschneidet. Die gesuchten Daten werden aus der Datenbank geholt und dem Frontend als Antwort zugesendet. Im Frontend wird die Antwort nun noch schön dargestellt und dem User dynamisch zur Verfügung gestellt. Dies ist also grob zusammengefasst, was im Hintergrund bei medcodesearch passiert.