Entwicklung eines Dashboards zur Verbesserung des Bewerbungsprozesses für Unternehmen und Bewerber

Inhalt

[Einführung 3](#_Toc151559280)

[Motivation 3](#_Toc151559281)

[Anforderungsanalyse 3](#_Toc151559282)

[Konzept 4](#_Toc151559283)

[Umsetzung 4](#_Toc151559284)

[Evaluation 4](#_Toc151559285)

[Fazit 4](#_Toc151559286)

# Einführung

Derzeit arbeite ich bei der 2Orgu GmbH in Oldenburg. Das Unternehmen beschäftigt sich mit der Internetpräsenz dieser Unternehmen, speziell deren Karriereseiten. Zu den Karriereseiten gehören eine Suchseite für alle Jobs eines Unternehmens, das Job Posting, die Bewerbungsseite sowie ein eine Profilseite für Bewerber, über die alle Bewerbungen und Job Alerts koordiniert werden können. Weiterhin gibt es noch eine interne Lösung für Mitarbeiter der Kundenfirmen, um den Bewerbungsprozess mit allen Bewerbern koordinieren zu können. Die Kunden haben hierbei eines gemeinsam: deren Daten liegen komplett in SuccessFactors, einer Cloud-basierten Softwarelösung in zum Bewältigen von Aufgaben in der Personalwirtschaft. Hierbei liegen alle Daten und in der Systemlandschaft von SuccessFactors. Die Softwareprodukte von 2OrgU schmiegen sich fast nahtlos an diese an und ermöglicht es, den Kundenwünschen entsprechend angepasste Webseiten zu erstellen.

# Theorieteil: Grundlagen

# Grundlagenliteratur

Keine

# Fachbegriffe (+Definition):

Typescript ist eine Skriptsprache die Javascript erweitert. Javascript ist eine populäre Sprache in der Webentwicklung und wird im Frontend und Backend benutzt. Typescript erweitert diese Sprache um Typsicherheit.

Json ist ein Textdateiformat. Es ist eine einfache Textdatei dessen Formatierung bestimmten Regeln unterliegt. Dieses Format ist häufig benutzt um Daten von Servern zu Klienten zu übertragen und die Daten im „Key-Value-Pair“ zu organisieren. Es können beliebige Schlüssel mit Werten hinzugefügt werden, die Menge ist hierbei auch frei bestimmbar.

Application ist eine Struktur im Json Format. Dieses Objekt spiegelt alle notwendigen und optionalen Daten von einer Bewerbung. Gespeichert werden mindestens folgende Daten: Id der Bewerbung welche automatisch und einmalig generiert wird. Es wird die Id der Stellenausschreibung gespeichert, jeder einzelne Statuswechsel und das dazugehörige Datum. Statuswechsel sind von jedem Kunden eigenständig definiert, die Anzahl variiert stark aber ist sehr fein granuliert und es gibt teilweise über 60 verschiedene Status. Die Status fangen an bei der Bewerbung, ob die Bewerbung weiter geleitet wurde, ob ein Vorstellungsgespräch stattfindet, ob es telefonisch oder vor Ort stattfindet, ob die Bewerbung abgelehnt oder abgesagt wurde. Zudem gibt es weitere vom Bewerber ausgefüllte Daten die aber nicht relevant für diese Arbeit sein werden. Bei der Bewerbung müssen Fragen beantwortet werden, erforderliche sind z.B. die persönlichen Daten des Bewerbers wie Vor- und Nachname, Wohnadresse, Email-adresse, Telefonnummer, Gehaltsvorstellung und ein weiteres vom Kunden definierte Zusammenstellung an Fragen.

Job ebenfalls im Json Format gespeichert. Enthält die Id der Stellenausschreibung. Weiterhin sind noch relevante Daten daran gespeichert wie der Ausschreibungstext des Jobs, die Ids der Bewerbungen auf diesen Job, wann die Jobausschreibungen begonnen hat und wann das Ablaufdatum der Ausschreibung ist.

Frontend- dies ist, seitens des Codes, alles was auf dem Gerät des Nutzers stattfinden. Der Nutzer ist hierbei ein Klient, der sich in einem Webfähigen Gerät eine Website aufruft und den Inhalt der Seite herunterlädt und im Webbrowser anzeigen lässt. Zu dem hier ausgeführten Code gehören in erster Linie das Strukturdokument, HTML (Hypertext Markup Language), das Styling Dokument css( cascaded style sheet) und die Logik ausführenden Skripte js (Javascript).

Dashboard – In dieser Arbeit wird das komplette Frontend „Dashboard“ genannt da es die Funktion einer Anzeigeseite für Statistiken hat. Es stellt die wesentlichen Informationen über das Fachgebiet der Bewerbungen und Stellenausschreibungen dar.

SAP ist ein deutscher Softwarekonzern. Dieser ist vertreten in den branchen Datawarehousing sowie im Finanzsektor und bietet eigenen Datenbanklösungen an.

SuccessFactors ist ein Teil von SAP und bietet eine Softwarelösung an um das Personalmanagement von Unternehmen digital zu steuern und zu leiten. Diese Lösung ist aufgeteilt in viele Module, dazu gehört auch der CareerSiteBuilder.

CareerSiteBuilder ist teil von SuccessFactors. Er bildet nur eines der vielen Module ab. Hier geht es um das Managing der Seiten bezüglich der Karriereseiten. Von der Startseite auf der man landet über die Jobsuche, das Bewerbungsformular, das Profil können hier beliebig viele Seiten hinzugefügt werden. Hierbei kann purer HTML code eingefügt werden. Über ein globales Skript, das auf jeder Seite injiziert wird, kann immer zusätzlicher code ausgeführt werden. Diese nutzen wir um eigene Skripte und Seiten zu laden. Weiterhin können noch globale benutzt werden um, wenn diese korrekt im Code eingebunden wurden, generelle Stylingwerte definieren. Diese Werte sind frei anlegbar und können Farbwerte für die Websites sein, Schriftgrößen, Schriftarten und so weiter.

Node.js

Stencil.js

Firebase ist ein Webdevelopment Platform bereitgestellt von Google. Auf dieser werden verschieden Services angeboten.

Firebase Realtime Database ist eine No-SQL Datenbank. Populäre SQL-Datenbanken sind z.B. von Oracle, PostGres, MySQL und einige mehr. Diese Form der Datenbank speichern Daten in einer festgelegten Struktur ab und werden mithilfe von SQL-Abfragen angesprochen. Sie sind in der Regel Relationale Datenbanksystem in der komplexe Verknüpfungen miteinander bestehen können. Hingegen ist Firebase eine No-SQL Datenbank bei der nur JsonObjekte mit beliebigem Inhalt abgespeichert werden können. Es werden bei einer Datenbank Abfrage keine Relationen zwischen den Daten hergestellt.

Firebase Cloud Functions bieten die Möglichkeit, eigene Funktionen in NodeJS auf dem Server auszuführen. Diese Funktionen können durch Anfragen ausgeführt werden. Diese Funktionen haben einen direkteren Zugriff auf eine angebundene Datenbank. In diesen Funktionen können z.B. Anfragen komplex bearbeitet und ein einfaches Resultat kann an den Anfragensteller zurückgeliefert werden. Beispielhaft können hier Statistiken aus den Daten einer Datenbank berechnet und aufbereitet werden. Das Auslöseverhalten der Funktionenkann hierbei den Bedürfnissen angepasst werden. Es ist möglich die Funktionen in festgelegten Zeitintervallen ausführen zu lassen. Sie können auch ausgeführt werden sobald sich ein Datensatz in der angebundenen Firebase Realtime Database geändert hat(CRUD – Create, Read, Update, Delete). Letzten Endes sind auch einfache http requests möglich.

http request - über das http-Protokoll können anfragen an webserver formuliert werden. Grundlage sind die Befehle GET, POST, HEAD, OPTIONS, TRACE sowie url-parameter

# Diskussionen/Kontroversen

Keine

# Motivation

Die Kunden von 2OrgU haben einen sehr hohen Durchsatz von Bewerbungen jeden Tag. Jeder Kunde hat viele Mitarbeiter, teilweise über 380.000. Hierbei fallen, laut Firmeninterner Analysen, teilweise mehrere tausend Bewerbungen pro Tag auf einige tausend Ausschreibungen in verschiedenen Ländern statt, pro Kunde. Bei dieser Größenordnung an Bewerbungen kann es besonders den Unternehmen schnell passieren, dass sie die Übersicht verlieren und sich Bewerbungsverfahren unnötig in die Länge ziehen oder nie komplett abgeschlossen werden. Die Kundenfirmen haben hier noch keine Lösung um Statistiken hierzu berechnet und angezeigt zu bekommen. Somit bietet es sich an, eine leicht zu bedienende und für Firmen einfach verfügbare Oberfläche zu programmieren.

# Anforderungsanalyse

Es sollen verschiedene Daten, die während des Bewerbungsprozesses entstehen, ausgewertet werden. Es soll der komplette Zeitraum, vom Beginn einer Bewerbung bis zum Ende, erfasst werden. Zu den erhobenen Daten gehören unter anderem die Statuswechsel einer Bewerbung. Jeder Wechsel soll, vereinfacht kategorisiert, erfasst werden. Es muss zudem bei jedem Statuswechsel das Datum mitgeliefert werden damit der Verlauf der Bewerbung in sich und mit anderen Bewerbungen verglichen werden kann.

Es soll möglich sein alle Statistiken von allen Bewerbungen für ein einziges Job Posting zu aggregieren. Hierbei sollen einfach Durchschnittswerte für die Statistiken angezeigt werden können.

All diese Statistiken sollen gefiltert werden können mit vordefinierten Zeiträumen. Die Zeiträume sind für die letzte Woche, den letzten Monat und das letzte Jahr. Für die letzten Woche sollen die Werte pro Tag angezeigt werden sowie für den Filter im letzten Monat. Über das Jahr dürfen die Daten pro Woche zusammengefasst werden. Weiterhin ist hier der Zeitraum eingegrenzt auf das Start- und Enddatum des Job Postings.

Letzten Endes soll es noch möglich sein die Statistiken aller Bewerbungen für alle Jobs zusammenzufassen. Auch hier sollen simple Mittelwerte angezeigt werden über einen definierten Zeitraum.

Damit jeder die Möglichkeit hat die Auswertungen sich visuell ansprechend ansehen zu können soll eine Internetseite existieren. Beim Öffnen der Internetseite werden aussagekräftige und einfache Statistiken, welche über alle Job Postings aggregiert sind, angezeigt. Zudem kann man durch das Selektieren einzelner Statistiken eine detaillierte Ansicht anschauen. Es soll möglich sein bei dieser Übersicht auf die Zusammenfassung eines einzelnen Jobs zu kommen. Von diesem Job aus kann man sich auch einzelne Daten anschauen.

Die Oberfläche der Internetseite soll einfach zu verstehen und einfach navigierbar sein. Das heißt auch dass keine Flut von Daten angezeigt werden soll. Zudem sollen kurze Ladezeiten zu einem flüssigen Erlebnis auf der Internetseite beitragen.

# Anforderungsanalyse Kunde 1

Wie läuft der prozess ab:

Jemand ist auf karriereseite->schaut sich die stellen an und bewirbt sich darauf:

->Wollen wissen, wie kommen die auf die stellen

->ausschreibungskanalanalyse(von wo kommen die bewerber, karriereseite, andere job boards)

->verkaufsgebiete/verkaufsregionen(ebenen)

->kartenansichten

->wieviele bewerbungen mobile vs desktop

->kunde will fragen, warum gibt es mehr vbewerbungen von einer plattform vs eine andere.

->quantitave analyse(wie viele bewerbungen kommen über welchen kanal[freunde, portale…])

->qualitative ergebnisse(wie viele menschen werden eingestellt im vergleich zu abgelehnten kandidaten, zu ausschreibungen.)

->frage besonders nach kanal weil diese ausschreibungen geld kosten.

->problem: dm hat verteiltes recruiting system, einmal in  SF und in interviewUI.

->wie viele statuswechsel gibt es dazwischen

->zeit von start bis ende.

->wie viele interviewtermine gibt es in einer bewerbung.

->aufteilung in bearbeitungszeit/wartezeit(auf bewerber)/liegezeiten(weil sich keiner drum kümmert).

->wenn z.B. status termin ist und der termin ist erst in 3 wochen muss das ausgerechnet werden.

->wie lange liegen die interviewtermine zeitlich auseinander.

Beim dem absagen status:

Wenn der bewerber seine bewerbung zurückzieht schauen ob es mit der bewerbungszeit zusammenhängt.

Abgelehnt bei vorauswahl, abgelehnt nach ersteinsicht: ob es eine verknüpfung zu der gesamtanzahl der bewerbungen auf einen suchauftrag. Das verhältnis aller „abgesagt“ statuse auf einen bestimmten job (z.b. filialleiter) und wieviel prozent wurden eingestellt. Einmal quer durch deutschland. Bezogen auf eine jobgruppe immer nur.

Wie viele gespräche haben stattgefunden bis der auswahlprozess abgegebochen wurde. Muss deswegen was am bewerbungsprozess geändert werden?

Differenzierung von internen zu externen bewerbungen.

Wunsche: jeden tag könnte eine neue frage sein und aus den rohdaten soll die antwort neu generiert werden können.

Gehaltsanforderungen von bewerbern: z.b. runtergefiltert auf filialleiter: spanne der wunschgehälter anzeigen können.

Ausblick: wie viele von den mitarbeitern sind nach einem halben jahr noch da?

Warum ist der unterschied der anforderungen zwischen unseren kunden unterschiedlich?

(mm hat die stellen aus den mäkrten nicht in successfactors aber in einem anderen ATS system)

Kpmg hat z.b. eine andere anforderung als retailer wie MMS oder DM.

# Konzept

Das Dashboard

# Umsetzung

Für das Backend wird Firebase benutzt, eine von Google zur Verfügung gestellte Entwicklungsplattform. Über diese Plattform nutze ich den Cloud Firestore. Dies ist eine NO-SQL Datenbank Vorteil ist das Einspeichern von JSON-Daten jeglicher Form, es muss vorher keine Struktur definiert werden. Zudem sind Schreib- und Lesezugriffe sehr schnell. Ein weiterer genutzter Service sind die Cloud Functions von Firebase. Hierbei können Funktionen im Backend geschrieben werden welche einen einfachen Zugriff auf die Datenbank haben. Die Funktionen können dann durch das Frontend, also dem Klienten, einfach per HTTP-Request angesprochen werden. Zudem können die Cloud Functions automatisiert angesprochen werden oder in festgelegten Intervallen.

Für die Erstellung des Dashboards wird Node.JS benutzt, welches auch auf dem Backend Server läuft. Zusätzlich wird die Webseite mit Stencil.js gebaut, einer Bibliothek zum Bauen von Bibliotheken von Komponenten, welche wiederbenutzbar sind.

# Evaluation

# Fazit