

1. Ausgangssituation

1.1 Projektumfeld

Diese Projektarbeit entsteht im Rahmen der externen, d.h. nicht im Rahmen einer Berufsausbildung stattfindenden Prüfung zum Fachinformatiker für Anwendungsentwicklung. Der Autor ist jedoch seit 2015 in diesem Beruf bei der BioArtProducts GmbH regulär, also ohne ein Ausbildungsverhältnis, angestellt. Aus diesem Grunde wurde sich dafür entschieden, ein Projekt zur späteren Nutzung im Betrieb einzureichen, und dieses in Absprache mit dem Arbeitgeber in der Arbeitszeit durchzuführen.

Die BioArtProducts GmbH ist Dienstleister für medizinische und pharmazeutische Unternehmen und entwickelt u.a. Software zur Datenerfassung, -verarbeitung und -auswertung, z.B. für medizinische oder pharmazeutische Studien oder Register. Die Auswertung dieser Daten erfolgt meist eng an Kundenvorgaben orientiert in unterschiedlichen Ausgabemedien, eine häufig wiederkehrende Anforderung ist jedoch die Darstellung des Datenbestandes als Diagramme. Die gebräuchlichsten Diagrammtypen (zum Beispiel Torten- oder Balkendiagramme) können allerdings nur die Verteilung einzelner Werte oder den Zusammenhang zwischen zwei Werten visualisieren. Die erfassten Daten zeigen sich im Gegensatz dazu in der Regel komplexer; so werden von Patienten üblicherweise Daten aus der Krankengeschichte, Medikamente, Risikofaktoren und bestehende Erkrankungen neben einer Vielzahl von Laborwerten erfasst. Diese Daten werden bislang nur selektiv auf Zusammenhänge hin untersucht, ein schneller Überblick über die Zusammenhänge ist nicht möglich. Hier soll diese Projektarbeit Abhilfe schaffen.

1.2 Prozessschnittstellen

Für die Diagrammerstellung kommt in der BioArtProducts GmbH die JavaScript-Bibliothek d3¹ zum Einsatz. Mit dieser ist seit geraumer Zeit eine Sammlung dynamischer Funktionen in Entwicklung, die ein Konfigurationsobjekt und Daten im Json-Format entgegennehmen, und den Angaben im Konfigurationsobjekt entsprechend Daten aggregieren, zählen und darstellen. Eine Anbindung an diese Funktionen ist gewünscht und soll dadurch realisiert werden, dass in der Oberfläche dieser Arbeit Spalten des Datenbestandes und die Form der Visualisierung ausgewählt, sowie optionale Konfigurationsparameter, wie z.B. Wertebereich, Diagrammtyp, Zuordnung der Daten zu Achsen, übergeben werden können. Dementsprechend soll das

¹ <https://d3js.org/>

Programm die gewählten Daten vorbereiten und die entsprechende Funktion aufrufen, um ein Diagramm zu generieren.

Die Daten sollen im Json-Format entgegengenommen werden. Dies passiert durch Aufruf des Programms mit Übergabe der Daten als Parameter, üblicherweise durch ein zugrundeliegendes Java-Backend, in das es eingebettet wird. Zum Testen und Präsentieren müssen jedoch Beispieldaten erzeugt werden, die von ihrer Art denen, die in Projekten der BioArtProducts GmbH erfasst werden, nachempfunden sind.

1.3 Projektziel

Ziel ist eine Visualisierung, die Kunden sowie potentiellen Kunden der BioArtProducts GmbH angeboten, und die in bereits bestehende Projekte mit geringem Aufwand eingebunden werden kann. Sie soll Auftraggebern von Studien und Registern, aber auch Ärzten und Schwestern, die für diese Studien und Register Daten erfassen, einen schnellen und interaktiven Überblick über den Datenbestand und Abhängigkeiten zwischen einzelnen Variablen bieten und als Vorstufe zur Erstellung von Diagrammen, die eine oder zwei Spalten des Datenbestands betrachten, dienen.

Die Visualisierung erfolgt in Form von Säulen, die die jeweilig erfassten Daten als Punktwolke ("beeswarm plot") darstellen - eine Säule für jede pro Datensatz erfasste Spalte. Für Zahlenwerte wird eine lineare, für Strings eine gewichtete Ordinalskala (Werteblocke, deren Höhe abhängig von der Häufigkeit des jeweiligen Wertes sind) verwendet. Jede Spalte soll dabei frei konfigurierbar bezüglich ihres angezeigten Wertebereichs sein. Auch soll es möglich sein, Grenzwerte zu kennzeichnen. Auf diese Weise lassen sich bereits die jeweiligen Werte und ihre Verteilung im Datenbestand auf einen Blick beurteilen. Nun lassen sich interaktiv einzelne Einträge (i.d.R. Patienten) hervorheben, indem der Mauszeiger über einen Punkt bewegt wird - dieser sowie korrespondierende Punkte anderer Spalten werden dann hervorgehoben. Über Schieberegler am vertikalen Ende der Säulen können Wertebereiche eingegrenzt werden, außerhalb dessen Punkte als deaktiviert ausgegraut werden. Die Auswirkung einer Variablen auf alle anderen Werte kann so sichtbar gemacht werden.

Sind so Zusammenhänge offenbar worden, die genauer dargestellt werden sollen, können die betreffenden Spalten zu einer Auswahl hinzugefügt werden. Für die Darstellung dieser Spalten als Diagramm werden im Folgenden Konfigurationsdetails abgefragt: welche Achse welche Werte darstellen soll, die Art des Diagramms (Linien, Balken) und betrachteter Wertebereich. Dann lässt sich das entsprechende Diagramm generieren oder alternativ als Grafik speichern.

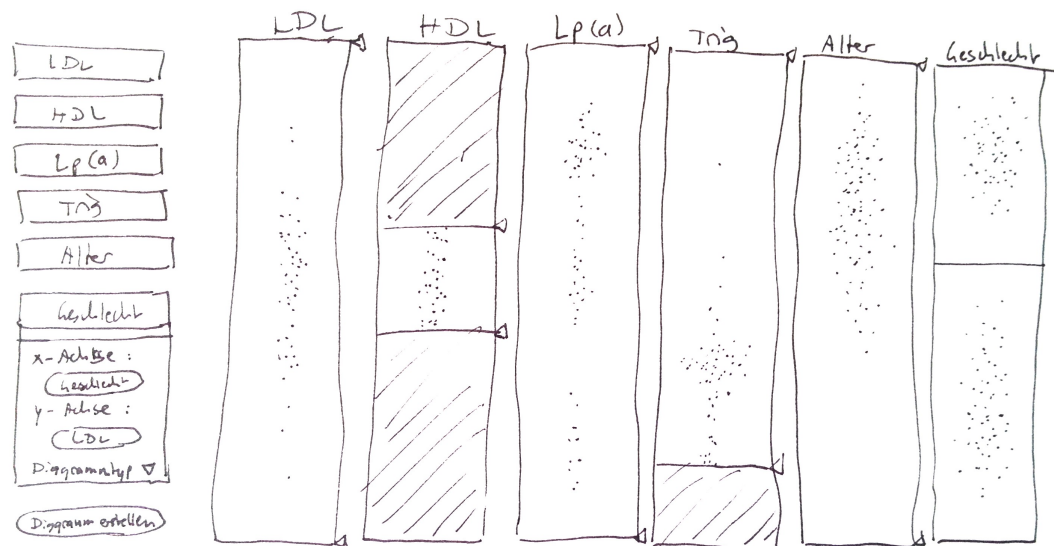


Illustration 1: Entwurfsskizze

1.4 Kundenwünsche

Um die spätere Nutzung durch andere Mitarbeiter zu erleichtern, ist der Code vernünftig zu kommentieren und soll vergleichbar mit bisherigen d3-Anwendungen aufgebaut sein. Dazu gehört die konsistente Benennung von Variablen und Methoden in Englisch, Formatierung und Einrückungen nach Firmenstandard, die Verwendung eines Konfigurationsobjektes zur leichten Anpassung an projektspezifische Anforderungen und Design und die Trennung von Berechnungen und grafischer Darstellung. Die Nutzeroberfläche ist möglichst einfach zu halten, da sie je nach Projekt sehr unterschiedlich implementiert werden wird und unter Umständen auch vollkommen entfällt, um zum Beispiel eine im Code vorkonfigurierte, nicht änderbare Darstellung zu präsentieren.

Abhängigkeiten von anderen Bibliotheken und Tools sollen möglichst vermieden werden. Aus diesem Grund wurde, abweichend vom Projektantrag, sich gegen dynamische Farbzweisungen entschieden. Die Darstellung erfolgt in Graustufen, ist jedoch leicht anpassbar und kann nach Bedarf erweitert werden.

1.5 Teilaufgaben

Folgende Teilaufgaben wurden identifiziert:

Projektplanung

Projektbeschreibung, Anforderungsanalyse, Absprachen im Unternehmen, grundlegende Architekturentscheidungen, Zeit-, Ressourcen- und Ablaufplanung

Erstellung von Beispieldaten

Auswahl sinnvoller Spalten, Identifikation von Grenzwerten und üblicher Verteilung von Werten sowie die Generierung dieser Daten unter Berücksichtigung dieser Verteilung und Grenzwerte mit Formeln, die Abhängigkeiten zwischen den Werten beinhalten.

Säulendarstellung von Daten für Zahlen und Zeichenketten

Iterative Platzierung von Datenpunkten in einem SVG-Graphen als beeswarm plot, Streuung der Punkte auf der X-Achse, Performanzoptimierung und Tests all dieser Features.

Eingrenzung des betrachteten Wertebereichs von Säulen

Gestaltung von Bedienelementen, Drag & Drop-Funktionalität von Bedienelementen, einfärben der eingegrenzten Zellen, Färben korrespondierender Zellen, Sicherstellen der Funktionalität bei Begrenzung mehrerer Spalten oder Begrenzung während des Zeichnens eines Graphen, Einrasten der Bedienelemente an Grenzwerte und Tests all dieser Features

Nutzeroberfläche erstellen

Spalten zur Darstellung auswählen, variable Datenmengen generieren, Geschwindigkeit der Darstellung anpassen, Datenstruktur und Grenzwerte eingeben, Testen

Daten zur Diagrammerstellung vorbereiten

Konfiguration erfassen, Daten gruppieren

Diagramme erstellen

Diagramme zeichnen, speichern, Design, Testen

Projektarbeit und Kundendokumentation schreiben

Projektarbeit entsprechend IHK-Bewertungsmatrix auf Grundlage von Planung, Kundendokumentation, Korrekturlesen

2. Ressourcen- und Ablaufplanung

2.1 Ablaufplanung

Die Zeitplanung erfolgt entsprechend der Teilaufgaben.

Projektplanung	12 h
Erstellung von Beispieldaten	4 h
Säulendarstellung von Daten für Zahlen und Zeichenketten	24 h
Eingrenzung des betrachteten Wertebereichs von Säulen	12 h
Nutzeroberfläche erstellen	8 h
Daten zur Diagrammerstellung vorbereiten	6 h
Diagramme erstellen	4 h
Projektarbeit und Kundendokumentation schreiben	10 h

80 h

Der Durchführungszeitraum streckt sich (entgegen den Angaben im Projektantrag) auf drei Wochen statt auf zwei, da die Projektarbeit in der Arbeitszeit des Anstellungsverhältnisses bei der BioArtProducts GmbH erstellt wird, und die vertragliche Arbeitszeit nur 25 Stunden pro Woche umfasst. Der Zeitraum selbst wurde aufgrund hoher Aufgabenlast im beantragten Zeitraum (18.-29.04.2016) und mit Rücksicht auf die notwendige Vorbereitung auf die schriftliche Abschlussprüfung am 03.05. auf die drei Wochen vor dem durch die IHK bestimmten Abgabetermin am 06.06.2016 gelegt.

Diese Aufgaben sollen sequentiell bearbeitet werden, mit Ausnahme der Projektarbeit, an der nach Abschluss jeder Teilaufgabe gearbeitet werden soll, um ein Aufschieben zu vermeiden.

2.2 Ressourcenplanung

Die Projektarbeit wird am Laptoparbeitsplatz mit Linux Ubuntu 14.04 der BioArtProducts GmbH durchgeführt. Versionskontrolle erfolgt mit git², Code und Dokumentation werden im Texteditor gedit³ geschrieben. Die Endfassung der Projektdokumentation wird mit Libre Office⁴ erstellt, Screenshots für die Kundendokumentation und andere Grafiken werden in Gimp⁵ bearbeitet. Alle diese Programme sind kostenfrei verfügbar. Für die Durchführung der Arbeit müssen weder Hard- noch Software angeschafft werden.

2.3 Kostenplanung

Die Kosten des Projekts liegen ausschließlich im Personalbereich, d.h. der Bereitstellung eines Arbeitsplatzes mit entsprechenden Arbeitsmitteln sowie Lohnzahlungen.

Zur Ermittlung der Anforderungen wurden im Vorfeld und in der Planungsphase Mitarbeitergespräche geführt. Zwischenstände sollen präsentiert und durch Kollegen getestet werden und zur Qualitätssicherung wird es Code Reviews geben. Daher schlüsseln sich die Kosten folgendermaßen auf:

Posten	Menge	Personalkosten	Ressourcenkosten	Gesamt
Entwicklungs	80 h	2000 €	1200 €	3200 €
Mitarbeitergespräche	4 h	100 €	60 €	160 €
Code-Review/Tests	4 h	100 €	60 €	160 €
	88 h	2200 €	1320 €	3520 €

Für einen Mitarbeiter wird ein Stundensatz von 25 € angenommen. Ressourcenkosten für den Arbeitsplatz liegen bei schätzungsweise 15 € pro Stunde.

2 <https://git-scm.com/>

3 <https://wiki.gnome.org/Apps/Gedit>

4 <https://www.libreoffice.org/>

5 <https://www.gimp.org/>

3. Durchführung und Auftragsbearbeitung

Für das Projekt wurde ein Git-Repository angelegt und ein Name gewählt, unter dem die Funktionen und Objekte aufgerufen werden können, die es zur Verfügung stellen soll. Dafür wurde sich für "Correlated Interactive Data Organisation", kurz "Corridor", entschieden, da sich dies als Kurzform gut merken und aussprechen lässt und die Langform das Kernziel des Projektes gut beschreibt. Im ersten Schritt wurden HTML- und CSS-Datei angelegt, Abhängigkeiten (d3) eingebunden und zwei Javascript-Dateien angelegt. Eine (data.js) beinhaltet alle Skripte, die für die Datenerzeugung notwendig sind und präsentationsspezifische Funktionen bereitstellen, die andere (main.js) alle übrigen Funktionen. Wird das Programm in anderen Projekten eingesetzt, genügt es zur Verwendung, die main.js einzubinden. Aus diesem Grund wurde auf eine weitere Spaltung in verschiedene Dateien verzichtet.

Im Folgenden wird die allgemeine Funktionsweise des Programms beschrieben. Implementierungsdetails können dem kommentierten Quellcode entnommen werden.

3.1 Beispieldaten mit Abhängigkeiten

Um für Testzwecke gleichbleibende Daten zu erhalten, wurde eine Zufallszahlenimplementierung, die mit einem Salz funktioniert, gewählt, also konsistent die immer gleichen Zahlen produziert, sofern das Salz, das der Funktion als Parameter übergeben wird, das gleiche ist. Ebenfalls benötigt wurde eine Funktion zur Erzeugung von Zahlen in Gaußscher Normalverteilung. Hier wurde die Java-Implementierung von nextGauss() umgesetzt und erweitert um die Möglichkeit, den gewünschten Mittelwert, die Standardabweichung und eine Funktion zur weiteren Beeinflussung des erzeugten Wertes zu übergeben.

Als Beispieldaten wurden Alter, Geschlecht, Medikamente, Risikofaktoren, Diagnosen und die Blutfettwerte Lp(a) (Lipoprotein a), LDL (Low Density Lipid) und HDL (High Density Lipid) gewählt. Diese Werte werden für verschiedene Studien, die von der BioArtProducts GmbH betreut werden, erhoben, so dass Häufigkeit von Erkrankungen und Verteilung von Laborwerten von Patienten, die für Apheresebehandlung infrage kommen, aus diesen Datenbeständen erhoben und näherungsweise nachgebildet werden können. Um zu Demonstrationszwecken Abhängigkeiten aufzeigen zu können, wurden Korrelationen für diese Daten in die Formeln zur

Generierung eingebaut; zum Beispiel wird der Betablocker Bisoprolol hauptsächlich von älteren Menschen eingenommen, Raucher sind häufiger männlich und Adipositas oder periphere arterielle Verschlusskrankheit erhöhen die Wahrscheinlichkeit eines hohen LDL-Wertes. Diese stellen keine realen Zusammenhänge dar, sondern dienen der Veranschaulichung, wie das Projekt Korrelationen sichtbar machen kann. Reale Daten können aufgrund von Datenschutzbestimmungen nicht verwendet werden.

3.2 Spaltendarstellung

Die Erzeugung eines Graphen zur Darstellung dieser Werte wurde mit der Visualisierungsbibliothek d3 realisiert. In einem SVG (Scalable Vector Graphic) fester Breite und von der Browserfensterhöhe abhängiger Höhe werden Datenpunkte auf y-Achse entsprechend ihres Wertes in der darzustellenden Spalte platziert. Die entsprechende Skala wird initial für den ersten Wert erstellt und mit den folgenden Daten nach Bedarf angepasst, und aktualisiert bei Veränderungen alle vorher platzierten Punkte. Für Zahlenwerte wird eine lineare Skala verwendet, entgegen dem Projektantrag wird hierbei nicht zwischen Fließkomma- und Ganzzahlen unterschieden, da in den Anforderungen an die Darstellung keine sinnvollen Unterschiede zwischen beiden bestehen. Zeichenketten wie z.B. Geschlecht oder Medikamente erhalten einen entsprechend ihrer Häufigkeit großen Bereich, in dem sie in Gaußscher Normalverteilung zufällig platziert werden. Auf diese Weise entstehen für letztere Werte anschauliche Punktwolken, deren Größe die Häufigkeit des Auftretens des jeweiligen Wertes widerspiegelt. Für die Verteilung auf der x-Achse wird jeder Punkt zunächst in der Mitte der Spalte platziert und im Folgenden mithilfe einfacher Kollisionserkennung (die jeden Punkt der Einfachheit halber als Quadrat auffasst) so lange im wachsenden Abstand wechselnd links oder rechts von dieser Mitte bewegt, bis keine Überschneidung mit anderen Punkten mehr vorliegt. Auf diese Weise wird sichergestellt, dass jeder Punkt einzeln erkannt und ausgewählt werden kann. Hat ein Punkt keinen Platz mehr, wird der Radius aller Punkte verkleinert und seine Position auf der x-Achse neu berechnet. Auf diese Weise ist das Programm in der Lage, auf Bildschirmen mit einer Auflösung von 1080 Pixeln in der Höhe je nach Verteilung der Werte mehrere tausend Punkte darzustellen. Da in dieser Größenordnung auch die Generierung der Graphen einige Sekunden in Anspruch nimmt, und auf dem zum Testen verwendeten Arbeitsplatz Verzögerungen bei der Bedienung der interaktiven Elemente auftreten, wird davon abgeraten, das Programm in seiner jetzigen Form auf größere Datenbestände anzuwenden und die Darstellung für diese nicht weiter optimiert.

Punkte werden iterativ platziert, um Daten mit wenig Anpassungen auch z.B. per Ajax entgegennehmen zu können, ohne dass der Gesamtdatenbestand und mit ihm Extremwerte für die Skalenerzeugung vorher bekannt sein müssen.

3.3 Interaktive Wertebereichsbegrenzung

Am oberen und unteren Ende jeder Spalte sind Elemente, die als Schieberegler fungieren, platziert. Klickt der Anwender mit der Maus und bewegt sie, folgen sie der Höhe des Mauszeigers. Auf diese Weise können schnell und einfach Grenzwerte bestimmt werden, die aus der Betrachtung entfallen. Die auf diese Weise ausgeschlossenen Punkte werden blass gefärbt, so dass sie deutlich als deaktiviert erkennbar sind. Ebenfalls entfärbt werden korrespondierende Zellen anderer Spalten. Jede Zelle speichert als Attribut, ob sie in ihrer Spalte begrenzt wurde. Zellen werden dann entfärbt, wenn mindestens eine korrespondierende Zelle als deaktiviert markiert ist, so dass mehrere Schieberegler auf mehreren Spalten benutzt werden können. Auf allen Säulen können also Eingrenzungen vorgenommen werden. Übrig bleiben in der Anzeige in ihrer Standardfarbe nur diejenigen Punkte, deren Datensatz auf keiner einzigen Säule begrenzt wurde. Angepasst werden musste dafür ebenfalls die Erzeugung von Zellen, da bereits beim Platzieren determiniert werden muss, ob eine Eingrenzung vorliegt oder nicht. Handelt es sich beim eingegrenzten Wert nicht um Zahlenwerte, sondern Zeichenketten, rastet der Schieberegler nur an Grenzen zwischen den einzelnen Werteblocks ein, anstatt frei beweglich zu sein. Ändert sich die Skala einer Spalte, weil noch Datenpunkte platziert werden, werden die Regler entsprechend ihrer Position auf der Skala, nicht ihrer absoluten Position, neu platziert.

3.4 Nutzeroberfläche

Bei Öffnen der HTML-Datei wird zunächst die Menge an Beispieldaten, die generiert werden soll, abgefragt. Empfohlen werden hier Werte von 50 bis 1000. Generieren von Daten entfernt diese Eingabemöglichkeit, an ihre Stelle tritt ein Auswahlfeld aller verfügbarer Spalten. Wird dort eine ausgewählt, wird diese wie bereits beschrieben dargestellt. Die Geschwindigkeit, mit der Punkte platziert werden, ist ebenfalls einstellbar, in Millisekunden je Punkt. Welche Größenordnung hierbei sinnvoll ist, richtet sich nach der Gesamtmenge an Daten und der Leistungsfähigkeit des verwendeten Computers. Je mehr Zellen bereits platziert wurden, desto länger kann die Berechnung der Position des nächsten Punktes dauern. Eine gute Faustregel ist, mindestens ein Hunderstel der Gesamtdatenmenge anzugeben. Hier besteht offensichtlich noch Optimierungsbedarf der Performanz.

Sind Spalten platziert, können diese ausgewählt werden, indem der Spaltentitel angeklickt wird. Dadurch erhält der Nutzer die Möglichkeit, Titel, Einheit und Grenzwerte zu verändern.

3.5 Diagrammerstellung vorbereiten und Diagramme erstellen

Bei der Begutachtung bislang zur Anwendung gekommener Diagrammfunktionen, wie sie in unter dem Punkt Prozessschnittstellen beschrieben wurden, wurde erheblicher Überarbeitungsbedarf festgestellt. Größtes Problem stellen Redundanzen und Uneinheitlichkeiten dar - ähnliche Funktionen nehmen zum Teil gänzlich unterschiedlich vorbereitete Daten entgegen. Einzelne Diagrammtypen sind nicht dynamisch konfigurierbar, sondern stellen spezifische Einzellösungen dar, die nicht wiederverwendet werden können. Manche Konfigurationsmöglichkeiten bestehen nicht für alle Diagrammtypen, und viele visuelle Aspekte sind nicht automatisch gelöst, erfordern zum Beispiel die manuelle Übergabe von Zeilenumbrüchen für Beschriftungen. Aus diesem Grunde konnte dieses Feature nicht wie geplant eingebaut werden. Diese Projektarbeit wird jedoch auch nach Abgabe fortgeführt, verbessert und erweitert. Dabei wird auch der alte Funktionsbestand grundlegend überarbeitet und vereinheitlicht, so dass eine Ankopplung später möglich wird. Die für diese Aufgabe veranschlagte Zeit wurde in die dynamische Anpassung der Skalen und Punktradien gesteckt, um ein visuell ansprechendes Ergebnis präsentieren zu können.

4. Projektergebnisse

4.1 Anpassungen

Umgesetzt wurde ein Visualisierungstool, das aus Daten eine Säulenansicht generiert, die iterativ Datenpunkte platziert und ihre Skala und die Punktgröße nach Bedarf anpasst. Durch Eingrenzung des Wertebereichs mit Schiebereglern können Zusammenhänge aufgezeigt werden. Größte Abweichung zum Projektantrag stellt der vollständige Verzicht auf die Möglichkeit dar, Spalten zur Diagrammerstellung auszuwählen und Diagramme zu generieren. Ursache hierfür findet sich in der ungenügenden Qualität und Einheitlichkeit des Codes, der dafür eingebunden werden sollte.

4.2 Soll-Ist-Vergleich

	Soll	Ist
Projektplanung	12 h	9 h
Erstellung von Beispieldaten	4 h	5 h
Säulendarstellung von Daten für Zahlen und Zeichenketten	24 h	12 h
Säulendarstellung von Zeichenketten und Listen von Zeichenketten		9 h
Dynamische Skalen und dynamischer Zellradius	–	14 h
Eingrenzung des betrachteten Wertebereichs von Säulen	12 h	16 h
Nutzeroberfläche erstellen	8 h	8 h
Daten zur Diagrammerstellung vorbereiten	6 h	–
Cleanup (Code kommentieren, strukturieren, optimieren)	–	4 h
Diagramme erstellen	4 h	–
Projektarbeit und Kundendokumentation schreiben	10 h	8 h
	80 h	85 h

Im Zeitaufwand sind einige Abweichungen feststellbar. Die Planung nahm deutlich weniger Zeit in Anspruch als veranschlagt, da Teilaufgaben aus dem Projektantrag und Absprachen im Unternehmen klar hervorgingen. Es konnte daher schnell zur Entwicklung übergegangen werden. Die Vorbereitung von Beispieldaten dauerte eine Stunde länger als erwartet. Verschätzt wurde sich auch mit dem Aufwand der Spaltendarstellung. Während die grundlegende Verteilung schneller als gedacht umgesetzt werden konnte (12h), erforderten die dynamischen Skalen und Zellradien mehrfache Umstrukturierung des Aufbaus (14h). Auch die Verteilung von Punkten, die Zeichenkettenwerte darstellen, zeigte sich aufgrund der veränderlichen Skalen als umfangreiche Aufgabe (9 h). Keine Abweichung gab es im Bereich der Nutzeroberfläche, zu der in der Ist-Rechnung auch Design der Graphen, z.B. Labels und dynamische Höhe, gezählt werden.

Die Eingrenzung des Wertebereichs stellte sich als Herausforderung heraus. Da die Anpassung der Darstellung in Echtzeit erfolgen sollte, musste hier nicht nur eine effektive, sondern auch effiziente Lösung gefunden werden. Auf Performanz lag ein besonderes Augenmerk.

Im Schreiben der Projektarbeit konnten zwei Stunden Zeit eingespart werden. Aufgrund konsistenter Verwendung von Versionskontrolle und gründlicher Kommentierung des Codes ließen sich Arbeitsweise, Entscheidungsprozesse und Zeitaufwand leicht nachvollziehen und zusammenfassen.

Insgesamt wurde die veranschlagte Zeit leicht überschritten. Dies konnte durch wenige Überstunden ausgeglichen werden, so dass eine pünktliche Abgabe erfolgen kann.

4.3 Qualitätskontrolle

Zur Sicherstellung der Lesbar- und Nachvollziehbarkeit des Codes wurde dieser ausführlich kommentiert. Änderungen wurden mittels Versionskontrolle per git festgehalten, so dass sie nachvollziehbar sind und jederzeit rückgängig gemacht werden können. Am Ende der Implementierung jedes Features stand das Testen desselbigen mit verschiedenen Datenbeständen und Bildschirmgrößen, um volle Funktionstüchtigkeit und konsistente Darstellungsqualität zu gewährleisten. Relevante Zwischenschritte wurden Mitarbeitern vorgeführt und diesen Gelegenheit gegeben, den Code zu studieren und eigenständig zu testen. Nach Abgabe dieser Arbeit wird eine engere Zusammenarbeit angestrebt, da der Verfasser für weitergehende Anpassungen nicht mehr gewährleisten muss, diese eigenständig und ohne fremde Hilfe durchzuführen.

4.4 Ausblick

Das Programm soll für die BioArtProductsGmbH zum Einsatz kommen und wird daher weiterentwickelt. Dafür wird es zunächst mit realen Studiendaten getestet und im Unternehmen präsentiert, um Änderungswünsche und Verbesserungsvorschläge zu sammeln und die Weiterentwicklung inhaltlich und zeitlich zu planen. Dazu gehört auch die Überarbeitung der Funktionen, die zur Diagrammerstellung entwickelt wurden und an die dieses Programm ursprünglich angeschlossen werden sollte. Ebenfalls notwendig ist die Verbesserung der Performanz; insbesondere die Neuberechnung von Zellpositionen bei Veränderung einer Skala kann bei großen Datenmengen zu Einfrieren der Webseite führen und muss, wo sie sich nicht beschleunigen lässt, ebenfalls iterativ erfolgen.

Zu den von Verfasserseite vorgeschlagenen Verbesserungen zählen weiterhin:

- Kopplung von allen ressourcenintensiven Selektionen an Timer oder AnimationFrames, um Einfrieren zu verhindern
- Lässt sich der Radius einer Zelle nicht weiter verkleinern, wenn die nächste Zelle keinen Platz mehr findet, soll sich seine Transparenz verringern und Überlappungen erlaubt werden
- Spalten mit Zeichenkettenwerten könnten Werte mit geringer Häufigkeit zu einem "Sonstige"-Block addieren
- Beschriftungen dürfen sich nicht überlappen oder aus dem Zeichenbereich hinausragen
- Farbmanagement für Grenzwerte und Zeichenkettenwertegruppen
- Spalten mit Datumswerten ermöglichen
- Weitere Darstellungsmöglichkeit von Spalten mit mehreren Werten pro Datensatz als Menge der Werte (z.B. Darstellung der Anzahl der Medikamente pro Patient, statt eine Aufschlüsselung)
- Überdenken des Wertebereichsbegrenzungskonzepts: unterbrochene Wertebereiche ermöglichen und Zeichenkettenwerte einzeln an- und abwählen, statt Schieberegler von oben und unten
- Beschriftung der prozentualen Verteilung aktiver Zellen (anstatt aller) von Zeichenkettenwerten bei Eingrenzung eines Wertebereichs
- Anzeige des exakten Grenzwertes auf numerischen Skalen bei Bewegen der Schieberegler
- Darstellung von Schnittmengen ermöglichen (z.B. Summe aus zwei Spalten oder Unterschiede zu Vorwerten)

5. Gestaltung des Projektes

Da das Programm in verschiedene Projekte eingebunden werden soll, die sich im Design stark voneinander unterscheiden können, wurde die Gestaltung möglichst neutral gehalten. Farben, Schriften und Graphenmaße können über CSS beziehungsweise Konfigurationsvariablen leicht angepasst werden. Für die Präsentation wurde ein schlichtes Design in unaufdringlichen Farben gewählt, das die Visualisierung für sich sprechen lässt.

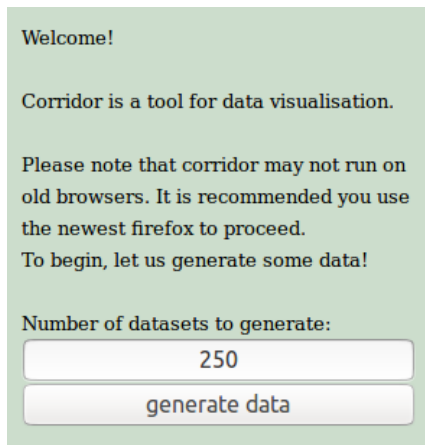
Besonderes Augenmerk wurde auf die Verständlichkeit der Säulendarstellung gelegt. Viele technische Entscheidungen sind mit dem Ziel einer ansprechenden und anschaulichen Visualisierung getroffen worden. Inhalt und Gestaltung lassen sich aus diesem Grund schwer trennen. So beginnt zum Beispiel die Darstellung mit Punkten großen Radiuses, die, sobald sie nicht mehr in eine Reihe passen, schrittweise verkleinert werden, um auch kleine Datensätze (und auch große Datensätze zu Beginn des Ablaufs des Säulenerstellungsprozesses) gut erfassen zu können. Die Platzierung von Zellen erfolgt iterativ, um keine Wartezeit entstehen zu lassen – Bedienelemente können sofort genutzt werden und ein Überblick ist schnell möglich, selbst wenn noch nicht alle Berechnungen abgeschlossen sind. Diese und weitere die Funktionsweise betreffende Gestaltungsfragen wurden im Abschnitt “Durchführung und Auftragsbearbeitung” bereits erläutert.

Die Gestaltung und Formatierung der Projektdokumentation wurde nach den IHK-Richtlinien⁶ vorgenommen. Auf ein Literaturverzeichnis wurde verzichtet, da keine Printmedien zur Erstellung verwendet wurden. Digitale Quellen wurden in Fußnoten beschrieben.

6 https://www.rostock.ihk24.de/aus_und_weiterbildung/Pruefungen/abschlusspruefung/Dokumentation_Projektarbeit/2646884

6. Kundendokumentation

Zum Start des Programms wird die index.html aufgerufen. Getestet wurde das Projekt in Google Chrome 48, Mozilla Firefox 46 und Internet Explorer 11, in anderen oder älteren Browsern kann es zu Darstellungsabweichungen oder Fehlern kommen. Es wird die Verwendung von Firefox empfohlen.



Welcome!

Corridor is a tool for data visualisation.

Please note that corridor may not run on old browsers. It is recommended you use the newest firefox to proceed.
To begin, let us generate some data!

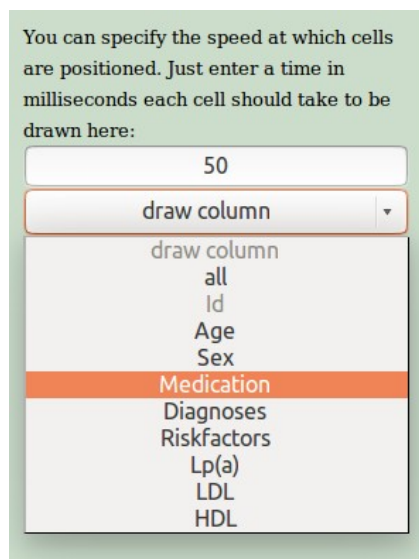
Number of datasets to generate:

250

generate data

Illustration 2: Eingabefelder zur Datengenerierung

Zunächst benötigt das Programm Daten, um sie darstellen zu können. Das erste Eingabefeld betrifft die Anzahl der generierten Datensätze. Die vorgegebene Standardmenge beträgt 250, empfohlen werden Eingaben zwischen 10 und 1000. Bei höheren Werten kann die Darstellung einige Zeit in Anspruch nehmen oder es zum Einfrieren des Browsers kommen.



You can specify the speed at which cells are positioned. Just enter a time in milliseconds each cell should take to be drawn here:

50

draw column

draw column
all
Id
Age
Sex
Medication
Diagnoses
Riskfactors
Lp(a)
LDL
HDL

Illustration 3: Eingabefelder zur Säulendarstellung

Ausgangszustand ist eine überwiegend leere Seite, auf der Titel und Untertitel sowie ein farbig hervorgehobener Bereich mit Text und Eingabefeldern zu sehen sind. In diesem Bereich finden sich Hinweise zur Bedienung und Formularelemente zur Konfiguration der Visualisierung. Die Hinweise sind in Englisch formuliert, da die BioArtProducts GmbH internationale Kunden hat, denen das Programm vorgestellt werden soll.

Mit Klicken auf den Button mit der Aufschrift "generate data" werden diese Daten generiert.

Im nächsten Schritt kann im Zahlenfeld die Geschwindigkeit der Darstellung in Millisekunden pro Datenpunkt angepasst werden. Als Standard ist hier 50 vorgegeben, empfohlen wird mindestens ein Hunderstel der im vorigen Schritt eingegebenen Datensatzmenge. Im Auswahlfeld mit der Aufschrift "draw column" kann anhand der Spaltennamen⁷ nun ausgewählt werden, welche Werte dargestellt werden sollen, oder auch alle auf einmal mit der Option "all".

7 Die Spalte "id" kann nicht ausgewählt werden, da sie nur der eindeutigen Identifikation eines Datensatzes dient. Ihr Inhalt wird jedoch angezeigt, wenn über eine Zelle mit der Maus gefahren wird.

Nun erscheint im weißen, leeren Bereich der Seite eine Säule (oder mehrere), in der in der vorher angegebenen Geschwindigkeit Punkte platziert werden. Weitere Säulen können auf die gleiche Weise hinzugefügt werden, es kann jedoch jede Spalte nur einmal dargestellt werden. Jede Säule hat eine Skala an ihrem linken Rand, auf der die Werte der Punkte abgelesen werden können. Bei numerischen Werten ist das eine lineare Skala, bei sonstigen Werten sind es Blöcke, in denen gleiche Werte zusammengefasst sind. Die Position innerhalb dieser Blöcke hat keinen Informationsgehalt - jeder Punkt ist gleichwertig zu allen anderen im gleichen Block. Die Größe, also Höhe, eines Blockes gibt Aufschluss über den Anteil der Datensätze mit einem solchen Wert, wobei ein Datensatz auch mehrere verschiedene Werte gleichzeitig haben kann. Die Anzahl der Datensätze ohne einen Wert für die betreffende Spalte wird unter dem Spaltennamen angezeigt.

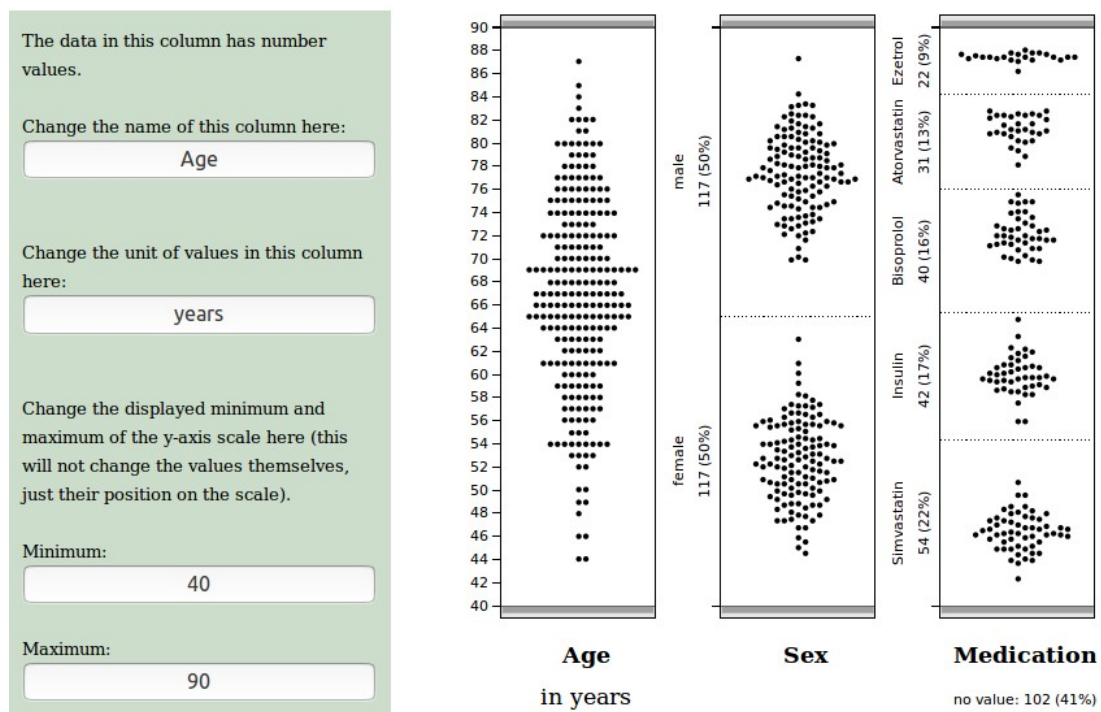


Illustration 4: Säulendarstellung und Säulenkonfiguration

Klickt man auf den Titel einer Spalte, der direkt unter jeder Säule steht, erhält man weitere Konfigurationsmöglichkeiten. Über ein Eingabefeld lässt sich der Name der Spalte ändern (diese Änderung wird sichtbar, sobald der Fokus das Feld verlässt, der Nutzer also z.B. an eine andere Stelle klickt). Spalten mit numerischen Werten haben darüber hinaus noch weitere Felder, über die sich die Einheit, in der besagte Werte angegeben werden, anpassen lässt, sowie die Minimal- und Maximalwerte der Skala.

Wichtigste interaktive Funktion stellen jedoch die Schieberegler dar, mit denen sich der Wertebereich einzelner Säulen eingrenzen lässt. Hierzu klickt man mit der Maus auf einen der flachen grauen Balken am oberen oder unteren Rand einer Säule und schiebt sie mit gedrückter Maustaste über die Säule. Schiebt man den Regler über Zellen, werden diese ausgegraut, ebenso wie alle anderen Zellen, die zum selben Datensatz gehören. Auf diese Weise lässt sich erkennen, welchen Einfluss die Werte dieser Spalte auf andere Werte des gleichen Datensatzes haben. Wird zum Beispiel in der Säule für Medikamente der Bereich "Insulin" ausgegraut, sind auch alle Zellen im Bereich "Diabetes" bei den Diagnosen betroffen. Daraus lässt sich schließen, dass im Datenbestand alle Patienten mit Diabetes Insulin nehmen und umgekehrt alle Patienten, die Insulin nehmen, Diabetes haben. Grenzt man hingegen die Spalte für das Patientenalter dahingehend ein, dass Patienten unter 60 Jahren ausgeblendet werden, betrifft dies mehr männliche als weibliche Patienten, und überproportional viele, die einen niedrigen HDL-Wert haben. Solche Eingrenzungen lassen sich auch auf mehreren Spalten gleichzeitig vornehmen.

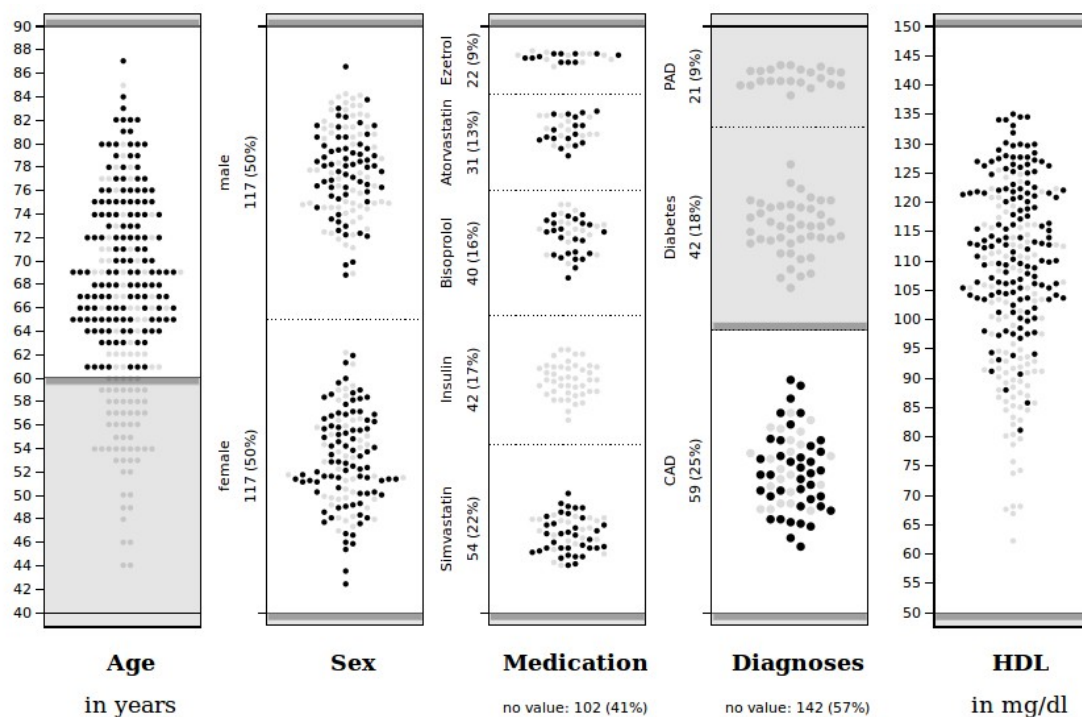


Illustration 5: Wertebereichsbegrenzung durch Schieberegler