

Projekt AGB-10

Entwurfsbeschreibung

Inhaltsverzeichnis

1	Allgemeines	1
2	Produktübersicht	2
3	Grundsätzliche Struktur und Entwurfsprinzipien für das Gesamtsystem	2
4	Grundsätzliche Struktur und Entwurfsprinzipien der einzelnen Pakete	3
4.1	Das Paket model	3
4.1.1	Die Klasse manager.GraphManager	4
4.2	Das Paket view	4
4.3	Das Paket data	5
4.3.1	Die Klasse ICDGraph	6
4.3.2	Die Klasse Disease	6
4.3.3	Die Klasse DiseasesList	7
4.4	Zustandsdiagramm aus Nutzersicht	7
4.5	Datenflussdiagramm	8

1 Allgemeines

Das vorliegende Projekt hatte zum Ziel, eine Desktopanwendung und ein dazugehöriges Applet zu entwickeln, welches Komorbiditätsnetzwerke grafisch darstellt. Als Datenbasis dienten dabei die Datensätze des Human Disease Network (Hudine <http://hudine.neu.edu>).

Wichtigste und folgenschwerste Entwurfsentscheidung war dabei die Quelldatensätze zu indexieren um mit den großen Datenmengen umzugehen. Dies hat sich unter anderem darauf ausgewirkt, dass in der fertigen Version kein Applet angeboten wird, da zu diesem Zweck entweder die Datensätze lokal bei jedem Benutzer vorhanden sein müssen (was dem Sinn eines Applets widersprechen würde) oder, dass die Datensätze über das Internet zum Anwender übertragen werden müssen, welches wegen der schier Größe der Daten nahezu jede Internetverbindung nicht schaffen würde.

2 Produktübersicht

Der fertige Comorbidity Graph Browser ist eine einfache Desktopanwendung. Der Nutzer kann Daten einlesen bzw. die zur Verfügung gestellten Daten nutzen, um sich Zusammenhänge grafisch darstellen zu lassen.

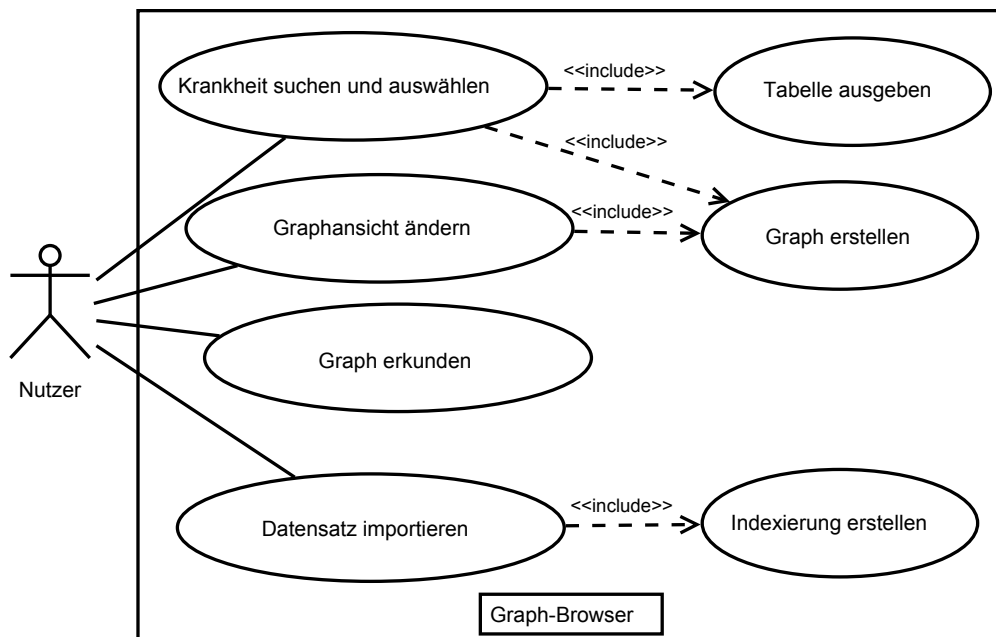


Abb. 1: Use-Case-Diagramm

3 Grundsätzliche Struktur und Entwurfsprinzipien für das Gesamtsystem

Das Projekt ist mit einem leicht modifiziertem MVC-Pattern entworfen worden. Prinzipiell erfolgt eine Trennung von Klassen, die mit der Ein- und Ausgabe von Daten zu tun haben (das Paket data), Klassen, die die Anwendungslogik enthalten (das Paket model) und Klassen für Nutzeroberfläche und Nutzerinteraktion (das Paket view). Das bedeutet, dass die Actions bzw. die ActionListener immer als interne Klassen in den grafischen Elementen realisiert sind.

Ein wichtiges Element ist dabei die Klasse GraphBrowserManager über die die Klassen Zugriff auf die weiteren Managerklassen und somit z.B. auf die Ergebnisse der Dialoge

haben. Dabei existiert immer nur eine Instanz des GraphBrowserManagers und auch die weiteren Manager existieren immer nur einmal im Programm.

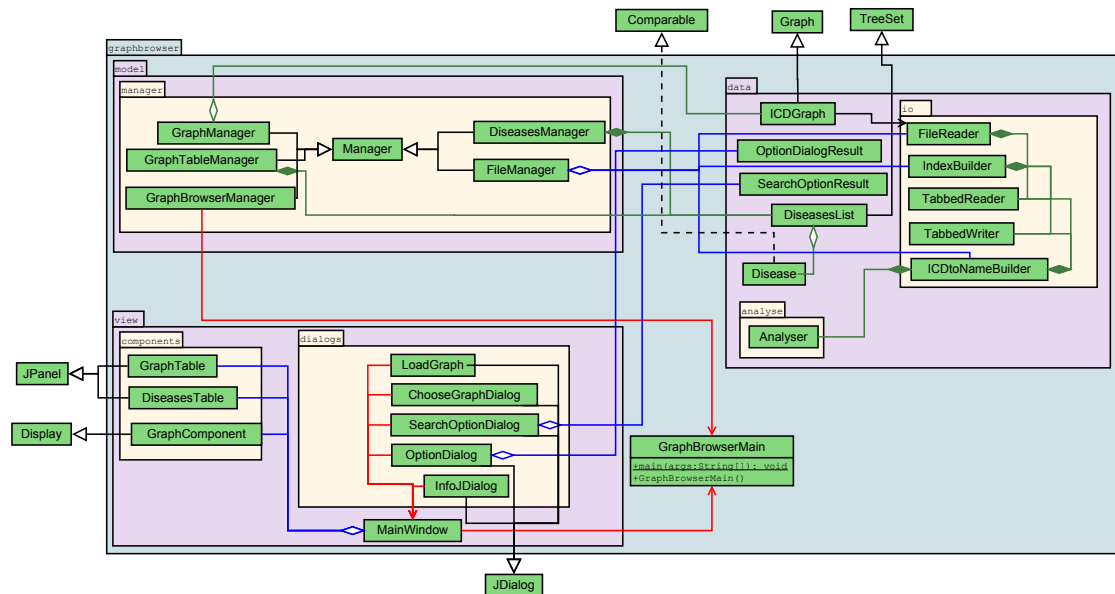


Abb. 2: Klassendiagramm des Gesamtsystems

Die öffentlichen Methoden der einzelnen Klassen folgen in den Klassendiagrammen für die einzelnen Pakete. In diesen fehlen allerdings aus Gründen der Übersichtlichkeit die Pfeile für Assoziationen und Kompositionen.

4 Grundsätzliche Struktur und Entwurfsprinzipien der einzelnen Pakete

4.1 Das Paket model

Das Paket model enthält verschiedene Managerklassen, welche die Programmlogik realisieren. Die Klassen in den anderen Paketen erhalten nur eine Instanz der Klasse GraphBrowserManager, welche das Singleton Pattern realisiert und die einzelnen Manager bereitstellt. Somit müssen die Klassen nur jeweils einen Manager im Konstruktor erhalten, wodurch die Wartung des Programms erheblich erleichtert wird.

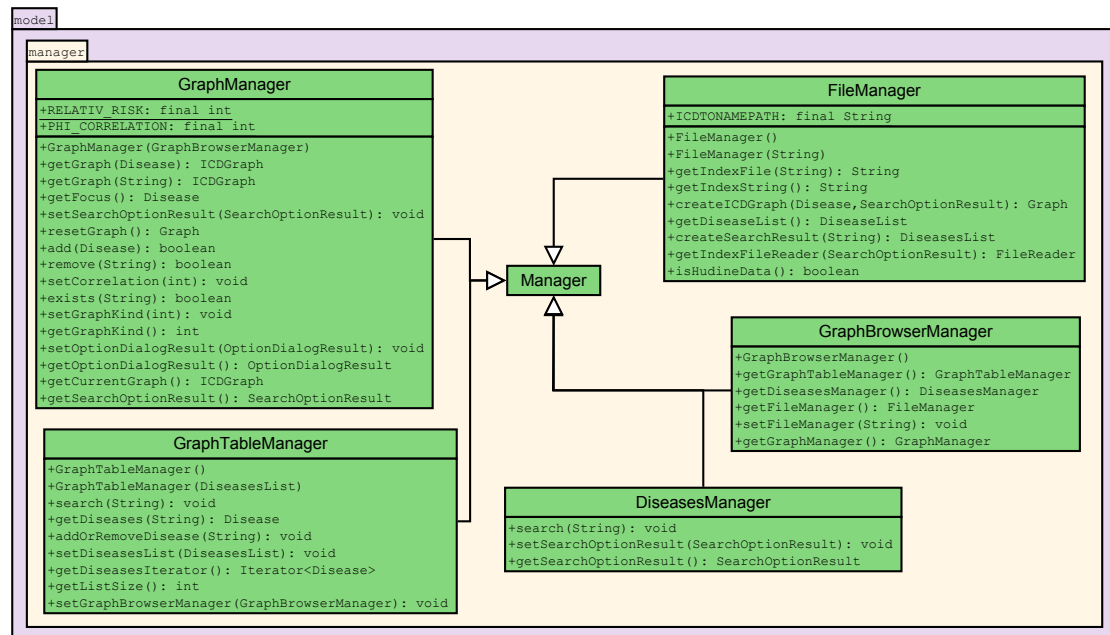


Abb. 3: Klassendiagramm des Pakets model

4.1.1 Die Klasse manager.GraphManager

Diese Klasse führt alle Berechnungen aus, die die Darstellung des Graphen betreffen. Dazu gehört vor allem das Entfernen und Hinzufügen von Krankheiten in den Graph.

4.2 Das Paket view

In diesem Paket befinden sich, wie bereits erwähnt, die Klassen zur Darstellung der Daten sowie die dazugehörigen Listener als interne Klassen. Das Paket ist wieder in die beiden Pakete components und dialogs aufgeteilt, welche, wie der Name bereits nahelegt, die Komponenten für die Darstellung sowie die Nutzerdialoge enthalten. Hauptklasse des Pakets ist die Klasse MainWindow, welche die einzelnen Komponenten und Dialoge zusammenführt.

Beim Erstellen der Oberfläche des Hauptfensters und der Dialoge haben wir den GUI-Builder Jigloo verwendet, der ein kostenloses Plugin für die IDE Eclipse ist.

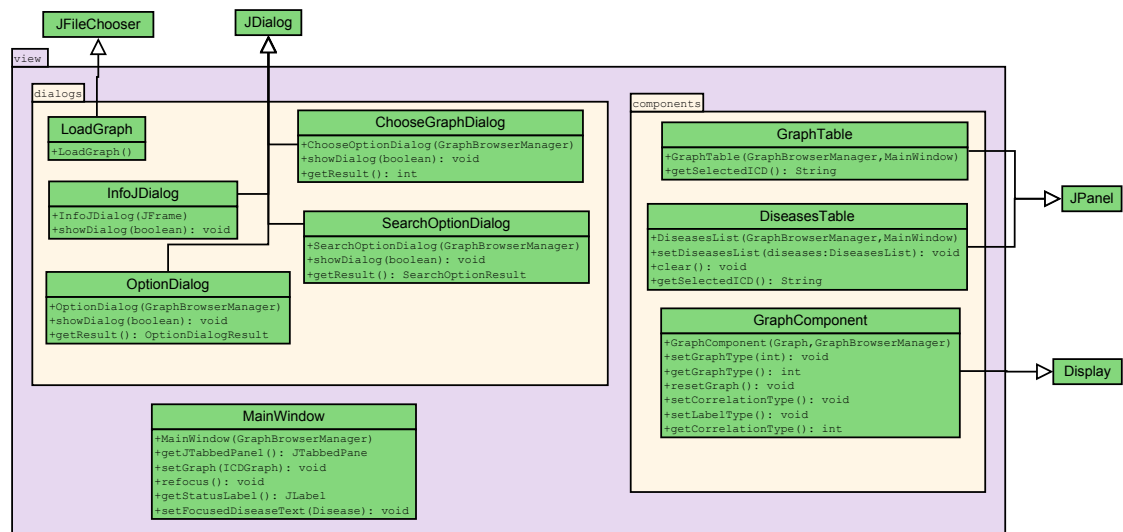


Abb. 4: Klassendiagramm des Pakets view

4.3 Das Paket data

Das Paket enthält einige Klassen sowie 2 weitere Pakete. Die Pakete sind io, welches Klassen für die Ein- und Ausgabe von Dateien enthält, und analyse, welches dafür sorgt, dass den ICD-Codes aus dem Quelldatensatz der Krankheitsname und weitere Informationen zugewiesen werden können.

4.3.3 Die Klasse DiseasesList

Die Klasse DiseasesList ist eine Datenstruktur für Objekte der Klasse Disease. Sie wird an mehreren Stellen des Programms verwendet: es werden die aktuell angezeigten Krankheiten gespeichert, außerdem wird sie benutzt, um die Ergebnisse der Suchfunktion zu speichern.

4.4 Zustandsdiagramm aus Nutzersicht

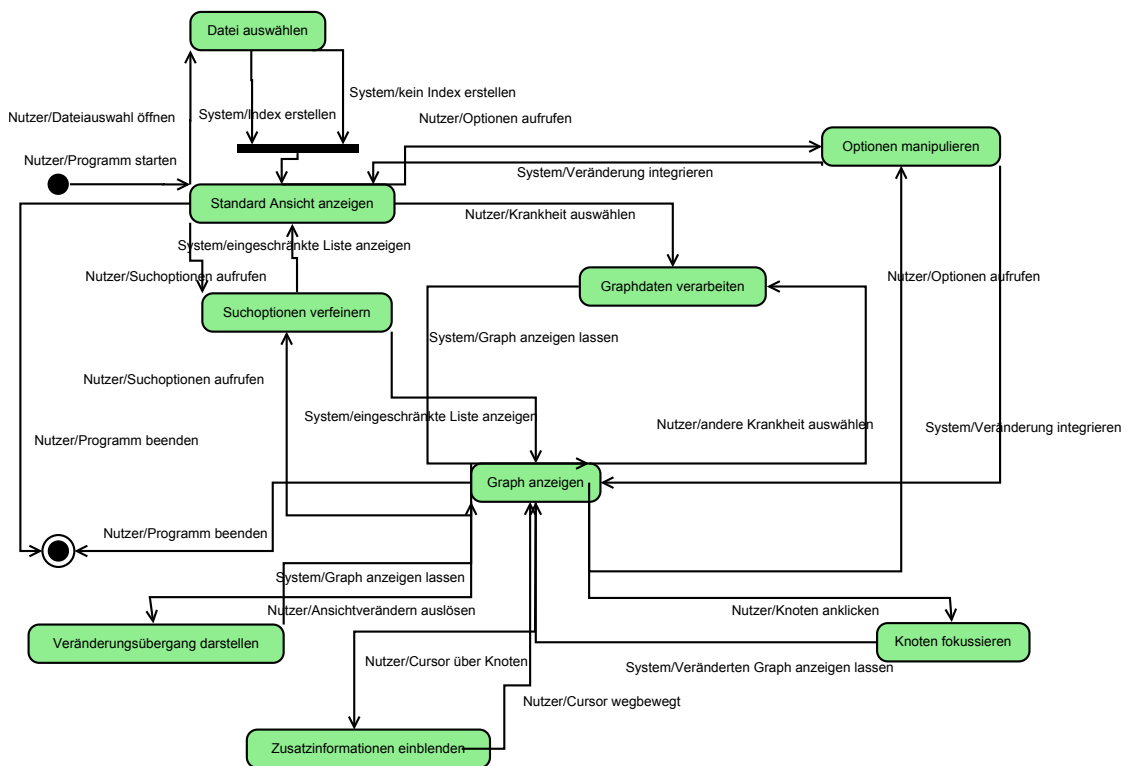


Abb. 6: Zustandsdiagramm

4.5 Datenflussdiagramm

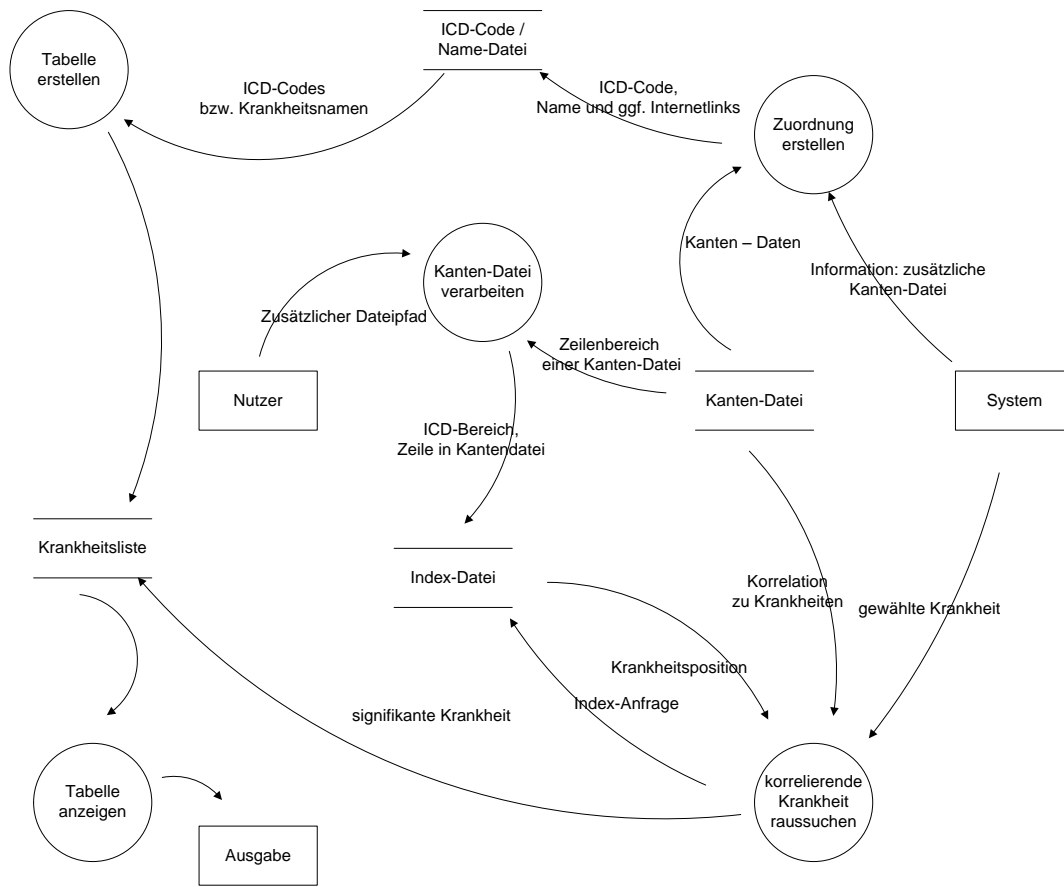


Abb. 7: Datenflussdiagramm