

Dokumentation

Wahllokal-Software für Landtagswahlen in Schleswig-Holstein

Johann Mantler

26.06.2011, Stand: Integration / Test

Inhaltsverzeichnis

1. Aufgabenstellung.....	2
2. Wahlhandlung aus Benutzersicht.....	3
3. Grobentwurf.....	4
4. Klassendiagramm	5
4.1 presentation layer	6
4.2 business layer	8
5. Architektur.....	10
5.1 Entwurfsmuster:.....	10
5.2 Besondere Datenstrukturen:.....	10
6. Ergänzungen	10

1. Aufgabenstellung

Wahlen 2011

Demnächst sollen die Landtagswahlen in Schleswig-Holstein durch Rechner unterstützt werden. Unterstützen Sie die Landesregierung. Erstellen Sie ein System zur Unterstützung der Wahl. Vielleicht kann es sogar einmal zur Bundestagswahl weiterentwickelt werden.

Das gesamte System besteht aus mehreren kooperierenden Teilsystemen, die getrennt voneinander entwickelt werden.

1. WaehlerVerzeichnis
2. WahlListen
3. Briefwahl
- 4. WahlLokal**
5. WahlErgebnis
6. SitzImParlament

Wahllokal:

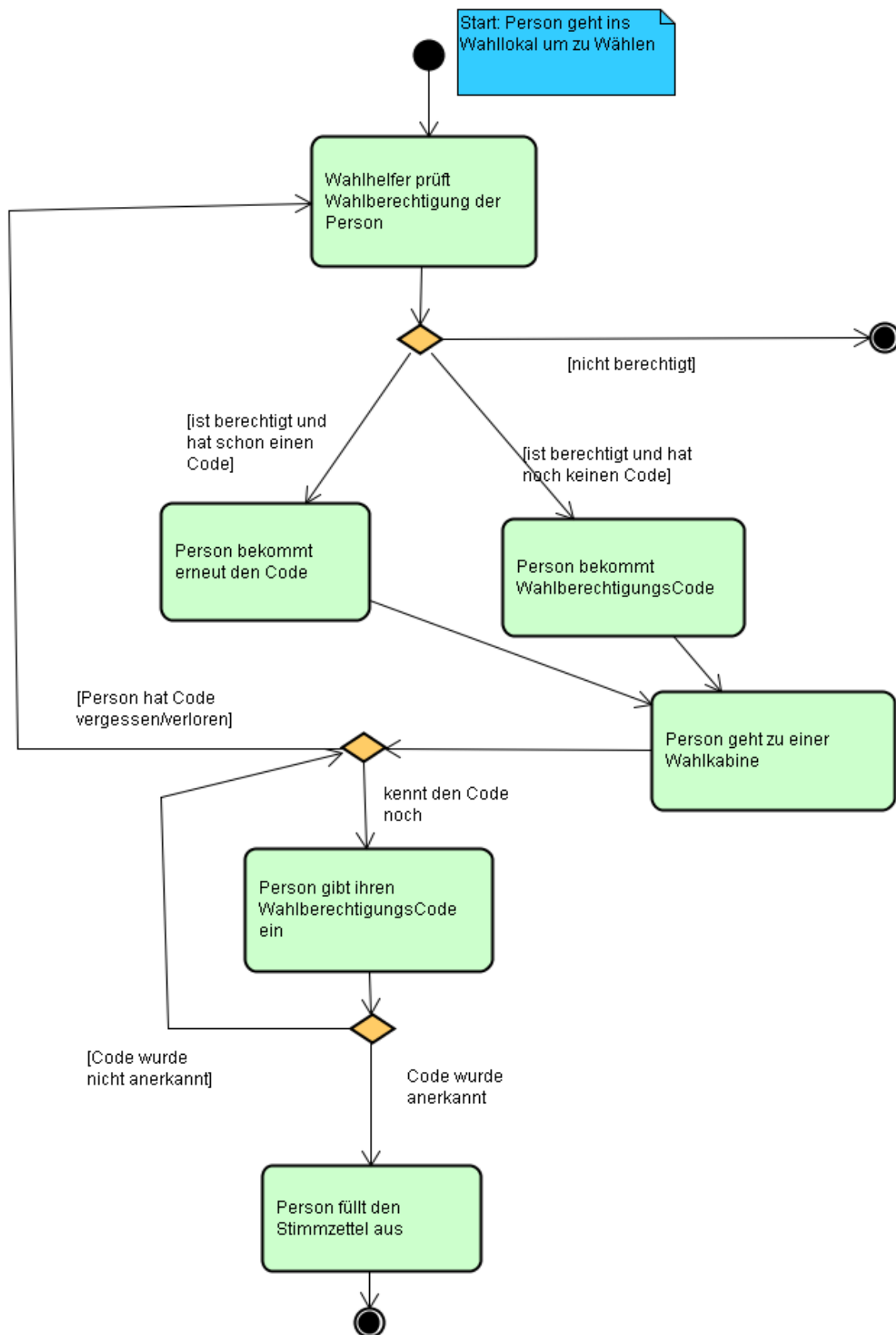
Am Wahltag wird die Wahl im Wahllokal unterstützt:

1. Abprüfen, ob eine Wahlperson auf der Liste der Wahlberechtigten des Wahllokals steht
2. Ausgeben "elektronischer Wahlberechtigung"
3. Eingabe Wahlberechtigung am Wahlrechner
4. Ausfüllen des elektronischen Stimmzettels am Wahlrechner
5. Ausdruck des Stimmzettels
6. Nach Schließung des Wahllokals wird die elektronische Zählung am Wahlrechner durchgeführt und das Ergebnis an die WahlleiterIn weitergegeben (Papier und elektronisch).

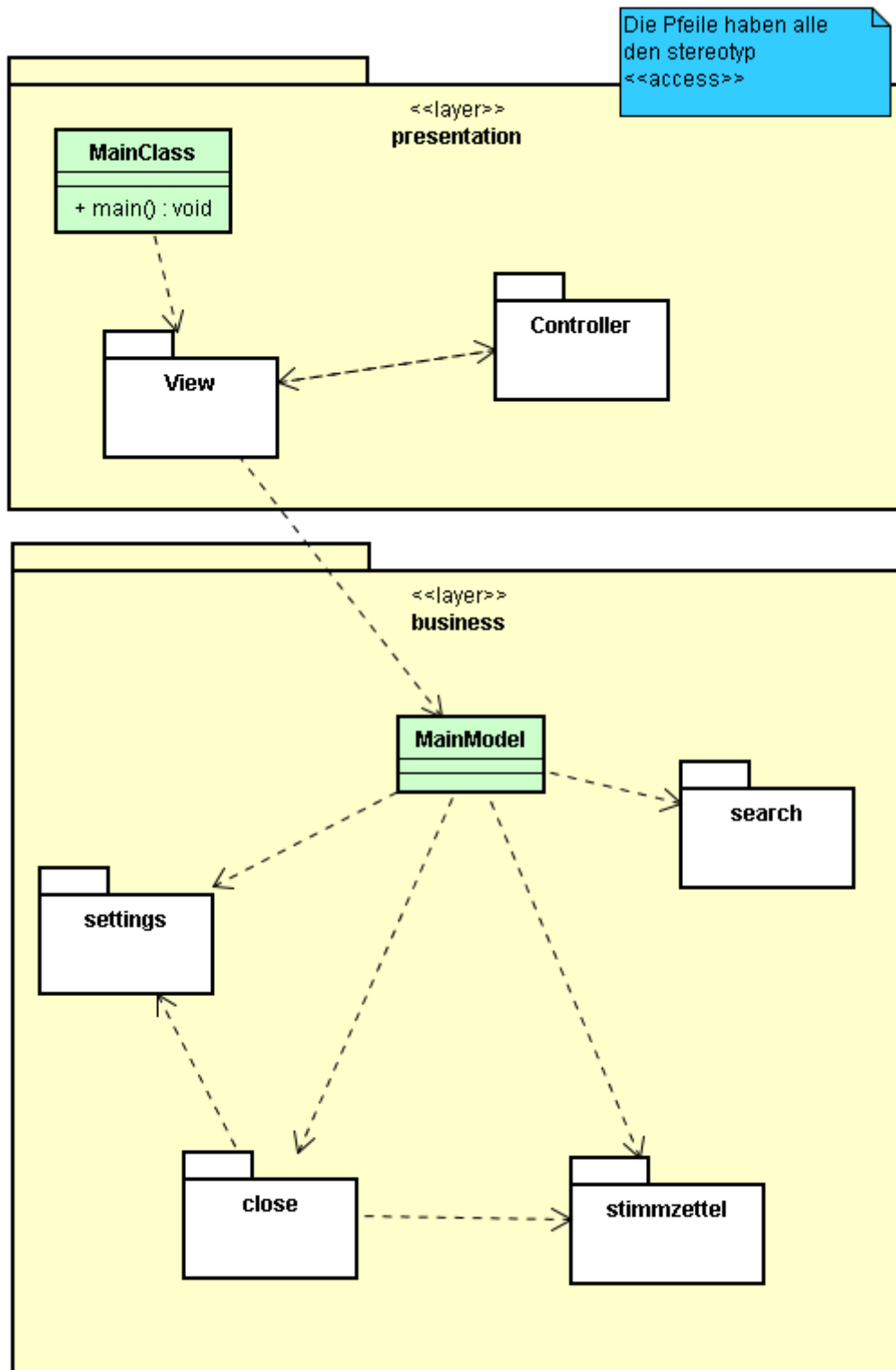
Anmerkung:

Aufgrund des Ausfalles eines Teammitglieds wurde der Umfang der Aufgabe reduziert. Dazu siehe Pflichtenheft.

2. Wahlhandlung aus Benutzersicht



3. Grobentwurf



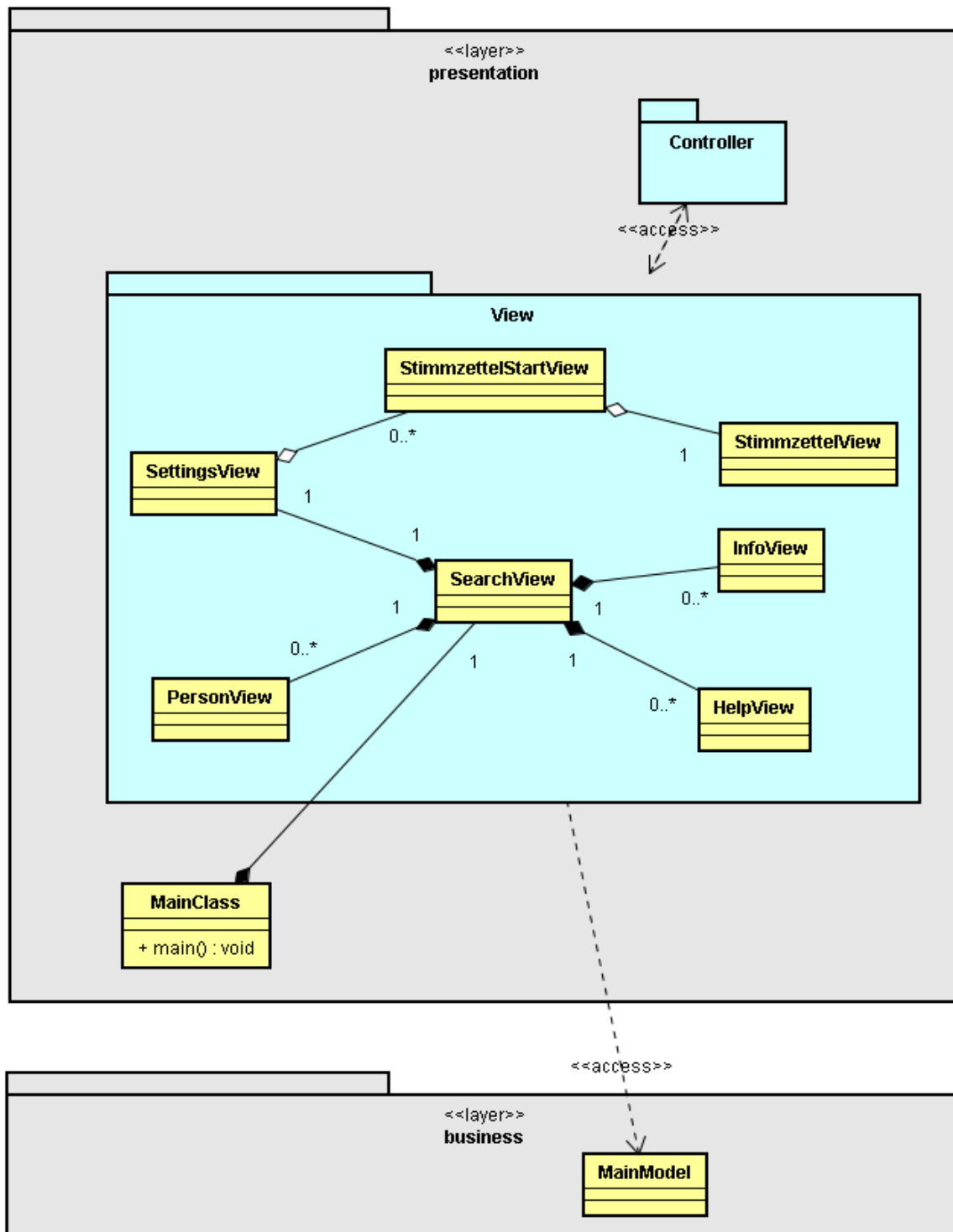
4. Klassendiagramm

Anmerkung:

Der übersichthalber wurde darauf verzichtet die Attribute und Methoden mit in das Klassendiagramm aufzunehmen, dafür sie Quellcode oder Javadoc Dokument.

4.1 presentation layer

4.1.1. Ansichten (Views)



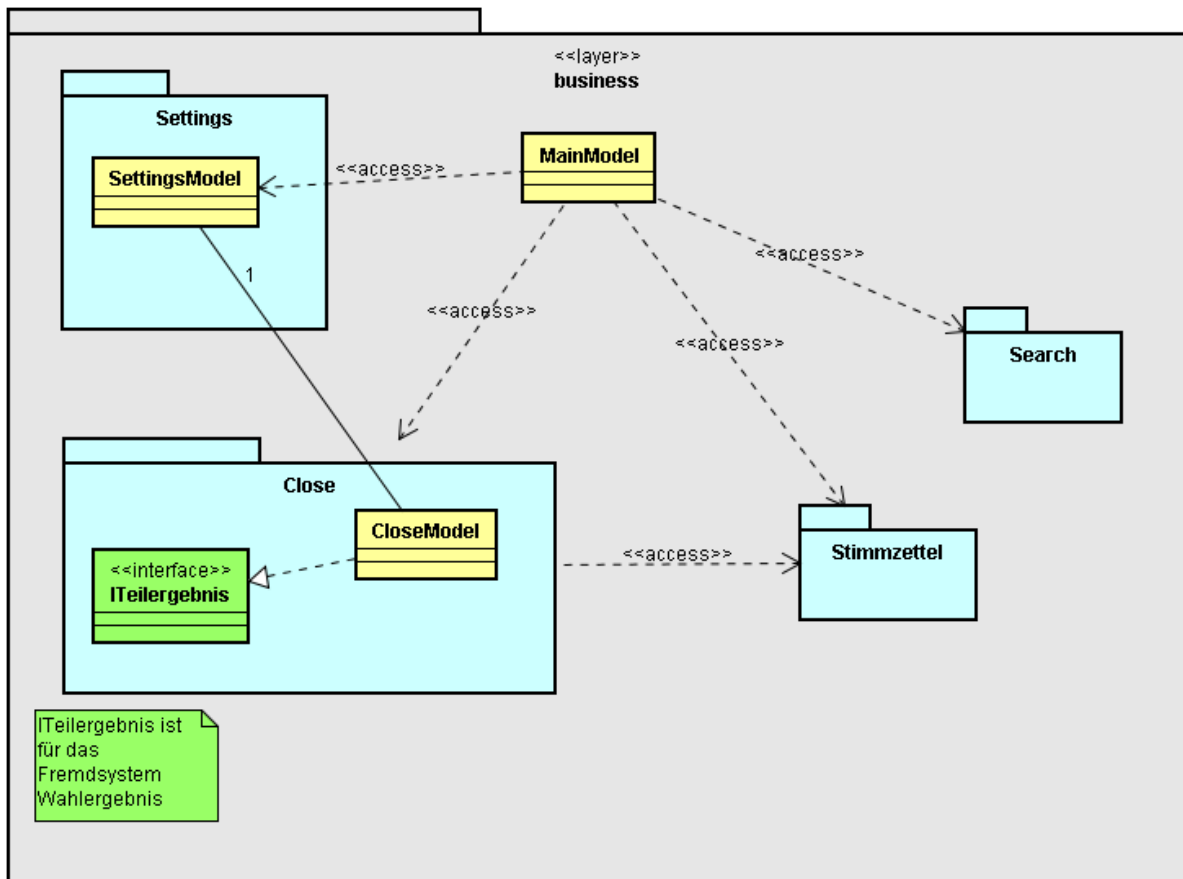
4.1.2 Kurze Erläuterung

Gestartet wird die Anwendung mit Hilfe der Klasse `MainClass` die als einzige Methode die Methode `main()` enthält. In dieser `main()` wird genau ein Objekt des `SearchView` erstellt.

Die Controller-Klassen kennen sich untereinander nicht, sondern nur ihre View. Die Klassenzuordnung von View zu Controller ist leicht durch den Namen ersichtlich z.B.: Ist die View vom PersonController die PersonView usw. .

4.2 business layer

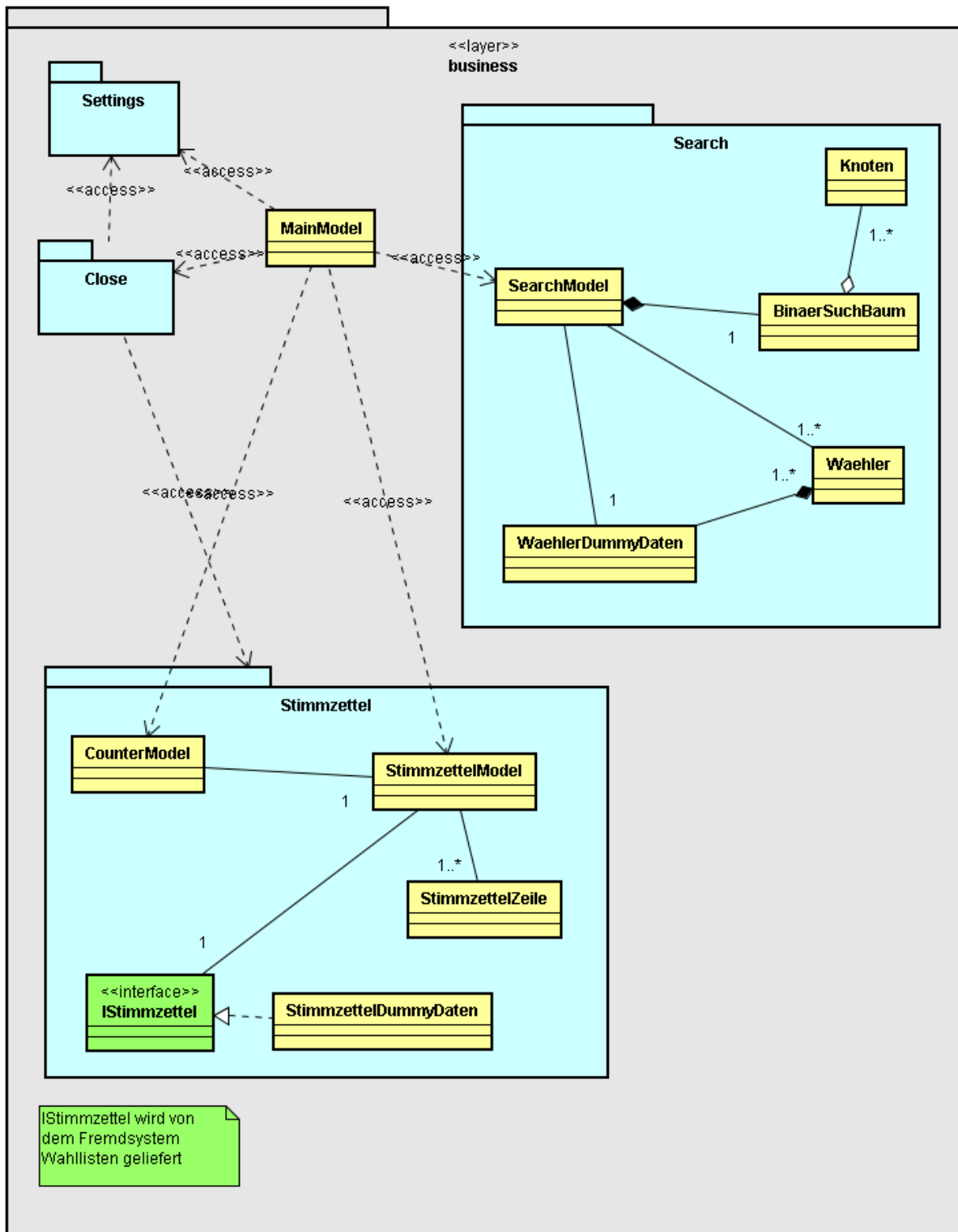
4.2.1 SettingsModel und CloseModel



Kurze Erläuterung

Das CloseModel kommt dann zum Einsatz, wenn das Wahllokal geschlossen wird. Darum hat es auch Zugriff auf die Stimmzettel- und Zählzeiten (Paket Stimmzettel) und auf die Einstellungen (Paket Settings). Das CloseModel fasst das Zählergebnis von Erst- und Zweitstimme zusammen und übergibt diese, dem Fremdsystem Wahlergebnis. Für weitere Details siehe Javadoc Dokumentation.

4.2.2 SearchModel und StimmzettelModel



Kurze Erläuterung

Im Paket Stimmzettel wird die eigentliche Zählung der Stimmzettelabgaben durchgeführt, sowie auch die Daten die zur Aufbereitung eines Stimmzettel nötig sind, vom Fremdsystem Wahllisten geholt. (Solange dieses System noch eigenständig läuft, wird die StimmzettelDummyDaten Klasse benutzt)

Im Paket Search wird nach einer Person gesucht, um herauszufinden ob diese auch wahlberechtigt ist oder nicht. Dazu müssen zuerst die Wähler vom Fremdsystem Wählerverzeichnis geholt werden. Aber da die Schnittstelle noch nicht von diesem Fremdsystem bereitgestellt worden ist, konnte kein Interface implementiert werden. Eine spätere Anpassung ist jedoch mit dem Entwurfsmuster Adapter möglich...

5. Architektur

5.1 Entwurfsmuster:

1. Um die Präsentation von der Geschäftslogik zu trennen, wurde das Modell-View-Controller – Prinzip herangezogen. Die Klassen sind bereits anhand ihrer Namensgebung als entweder Model, View oder Controller identifizierbar.
2. Das System basiert derzeit auf einem 2-Schichten-Modell, presentation und business. Die Datenhaltungsschicht mit dem Entwurfsmuster DAO war geplant, konnte aber aufgrund des Ausfalls eines Teammitgliedes nicht mehr realisiert werden.
3. Die Klassen MainModel, SearchModel, SettingsModel, CounterModel, StimmzettelModel, SettingsView und SearchView sind als Singleton-Klassen definiert. Warum diese Klassen Singleton sind, steht im Quellcode und in der Javadoc Dokumentation.
4. Das Muster Delegation wurde in der Klasse MainModel benutzt. Diese Klasse fungiert als Delegator, der Aufgaben von der obersten Schicht, der Präsentationsschicht bekommt, und diese an die jeweiligen Fachmodels weiterleitet. Dadurch wird erreicht, das die Präsentationsschicht lediglich die Klasse MainModel kennen muss.

5.2 Besondere Datenstrukturen:

1. Im Paket Search (business layer) wird ein Binärer Suchbaum benutzt um die Wählerobjekte darin zu speichern und effizient zu suchen. Das ist besonders dann sinnvoll wenn man bedenkt das die Suchzeit so auf $O(\log n)$ beschränkt ist und es in einem Wahllokal eventuell Mehrere Tausend Wähler gibt.

6. Ergänzungen

Aufgrund von anfänglichen Schwierigkeiten mit Maven sind im Repository 2 Ordner zu sehen:

- WahlLokal
- wahllokal

Wobei das letztere das richtige ist.