|  |  |  |
| --- | --- | --- |
| Start | Ende | Aktivität |
| 11.01 | 11.01 | Beschreibungen der Model Schnittstellen |
| 12.01 | 14.01 | Implementierung der DataModel Klassen |
| 15.01 | 18.01 | Anbindung an das SQLite – Framework (EntityDatabase) |
| 19.01 | 19.01 | Implementierung der EntityFactory und der ModelFacade |
| 20.01 | 21.01 | Grundgerüst des ParserModels, dateiformartunabhängiger Teil |
| 22-01 | 28.01 | Implementierung des Hl7Parsers, Anbindung an Everest-Framework |
| 29.01 | 01.02 | Darstellung von Enums in einem Picker |
|  |  |  |
|  |  |  |

Änderungen:

EntityObserver Schnittstellen entfernt, Grund: Beobachter Prinzip nicht mit SQLite-Extensions möglich, stattdessen manuelles Updaten über ModelFacade, intuitiverer Vorgang für ViewModel

DataModel: DoctorsLetterGroupDoctorsLetter Klasse hinzugefügt: Um Many-To-Many Relationen darstellen zu können, benötigt das SQLite-net-extensions Framework eine separate Klasse für die Relation

Hl7ParserHelper hinzugefügt: Bestimmte Klassen im Everest-Framework, die zum Parsen von Hl7Dateien benutzt werden, implementieren die ICloneable Schnittstelle, die in einem Xamarin-Cross-Platform Projekt nicht verfügbar ist. Operationen mit diesen Klassen werden nun über den Hl7ParserHelper plattformspezifisch ausgeführt

Enum (mit Extension) für die verfügbaren Dateiformate hinzugefügt (erlaubt es einfacher später Unterstützung für weitere Dateiformate hinzuzufügen)

FileToDatabaseParser: Init(string filename) Methode hinzugefügt, ParseMethoden sind dafür parameterlos. Erlaubt flexibler aus einer Datei lesen zu können, beim vorherigen Modell musste für jeden Parsingschritt neu aus der Datei gelesen werden, so können gemeinsame Vorbereitungen getroffen werden (z.B. beim Hl7Parser: Parsen des gesamten Dokuments mit dem Everest in frameworkeigenen Datentyp der dann mit den restlichen Methoden in die DataModel Datentypen geparst wird)

Utilities für Enum-Picker im ProfileTabViewModel: Um Elemente aus einem Enum darstellen zu können wird ein Konvertierer benötigt, da die dafür von Xamarin angebotene Funktionalität eine Liste von Elementen erwartet. Über den Konvertiere kann jedem Listeintrag ein Enum-Wert zugewiesen werden.

ModelFacade: Methode zum Abfragen des aktuell aktiven Profils hinzugefügt

FileHelper: Methode zum Löschen von Dateien hinzugefügt, Wird nun nichtmehr von auschließlich von der ModelFacade, sondern immer dort wo auch der Dateizugriff stattfindet verwendet (sonst werden aufrufe auf solche Methoden von der ModelFacade unnötig kompliziert)