

Embedded Systems

Block 3: Systemarchitektur Nachbereitung

Projekt: Unihockey-App

Inhaltsverzeichnis

1	Auftrag	2
1.1	Gemäss Moodle	2
1.2	Gemäss Mail vom 19.04.2015	2
2	Softwareentwurf	3
2.1	Klassenmodelle	3
2.1.1	System	3
2.1.2	UnihockeyDataService	3
2.2	Datenmodell	5
2.3	Dynamische Modelle	6
2.3.1	Sequenzdiagramm	6
2.3.2	Sequenzdiagramm UnihockeyDataService	7

1 Auftrag

1.1 Gemäss Moodle

Überarbeiten des Architektur-Entwurfs unter Berücksichtigung:

- der OO-Design Prinzipien
- Architektur- und Entwurfsmustern
- Implementieren der Komponenten (zumindest des Gerüsts)
- Unit-Test erstellen
- Testplan ergänzen

Stellen Sie die Architektur grafisch dar und erstellen Sie ein Codegerüst. Veröffentlichen Sie die Resultate im dafür vorgesehenen Forum.

1.2 Gemäss Mail vom 19.04.2015

Aus der Dokumentation zur Architektur muss ersichtlich sein, aus welchen

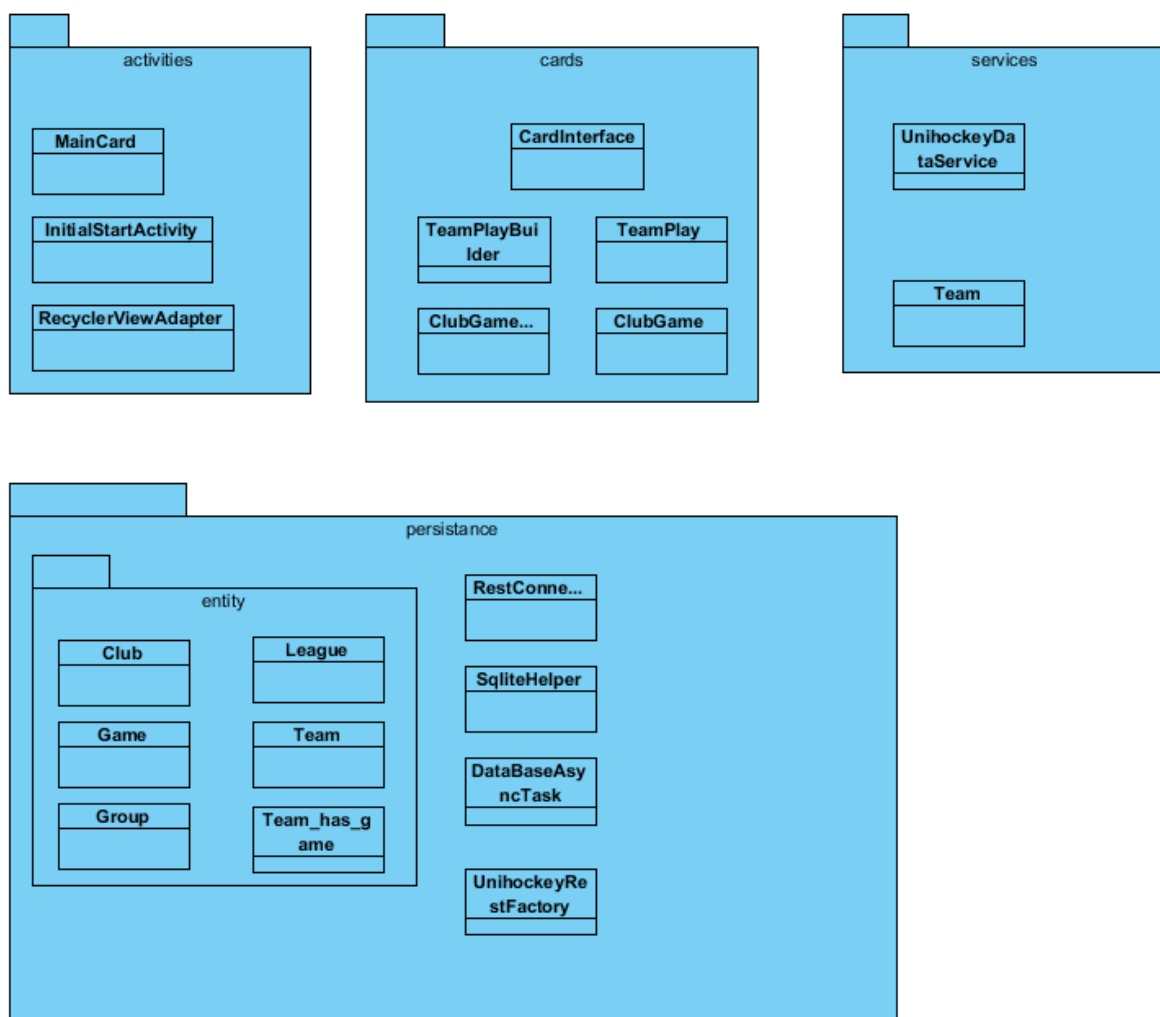
- Komponenten, oder Klassen das System besteht (-> Klassendiagramm, evtl. Verteilungsdiagramm)
- Welches die aktiven Komponenten oder Klassen sind (Activities, Services, Threads)
- Wie diese Komponenten oder Klassen miteinander interagieren. (Sequenzdiagramme, evtl. Zustands- oder Aktivitätsdiagramme)

2 Softwareentwurf

2.1 Klassenmodelle

2.1.1 System

Das folgende Diagramm zeigt unsere Applikationsarchitektur zum momentanen Zeitpunkt aufgeteilt in die Packages.

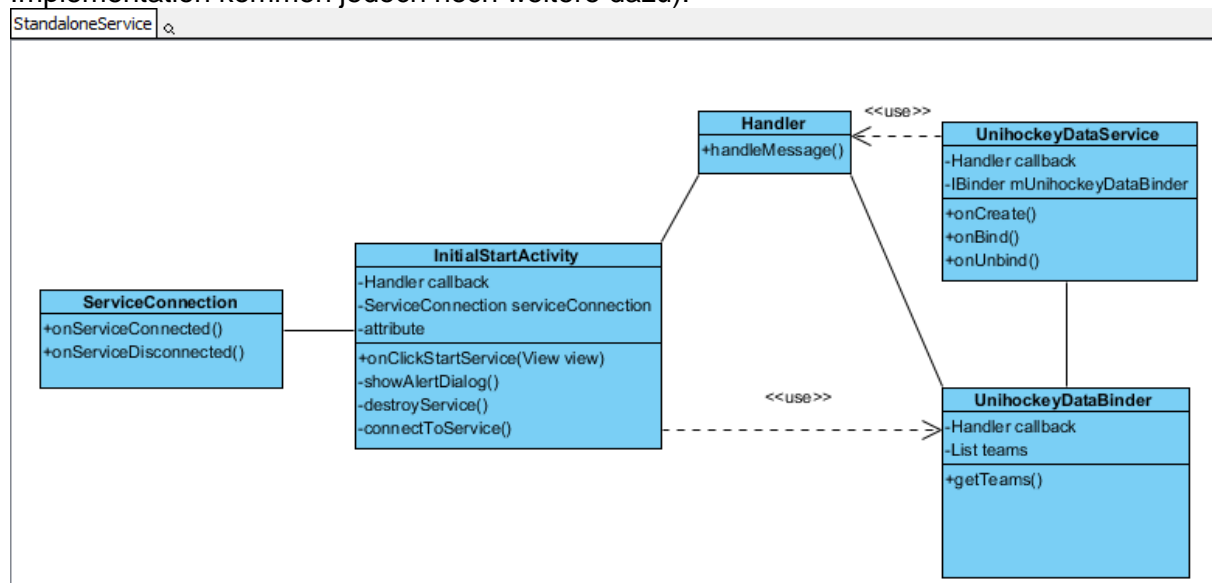


2.1.2 UnihockeyDataService

In unserer Applikation müssen Daten von einer REST Schnittstelle bezogen werden. Diese Daten werden an unterschiedlichen Punkten der Applikation beantragt, heruntergeladen und in die interne DB gespeichert. Dieser Prozess läuft in einem Service den wir `UnihockeyDataService` nennen.

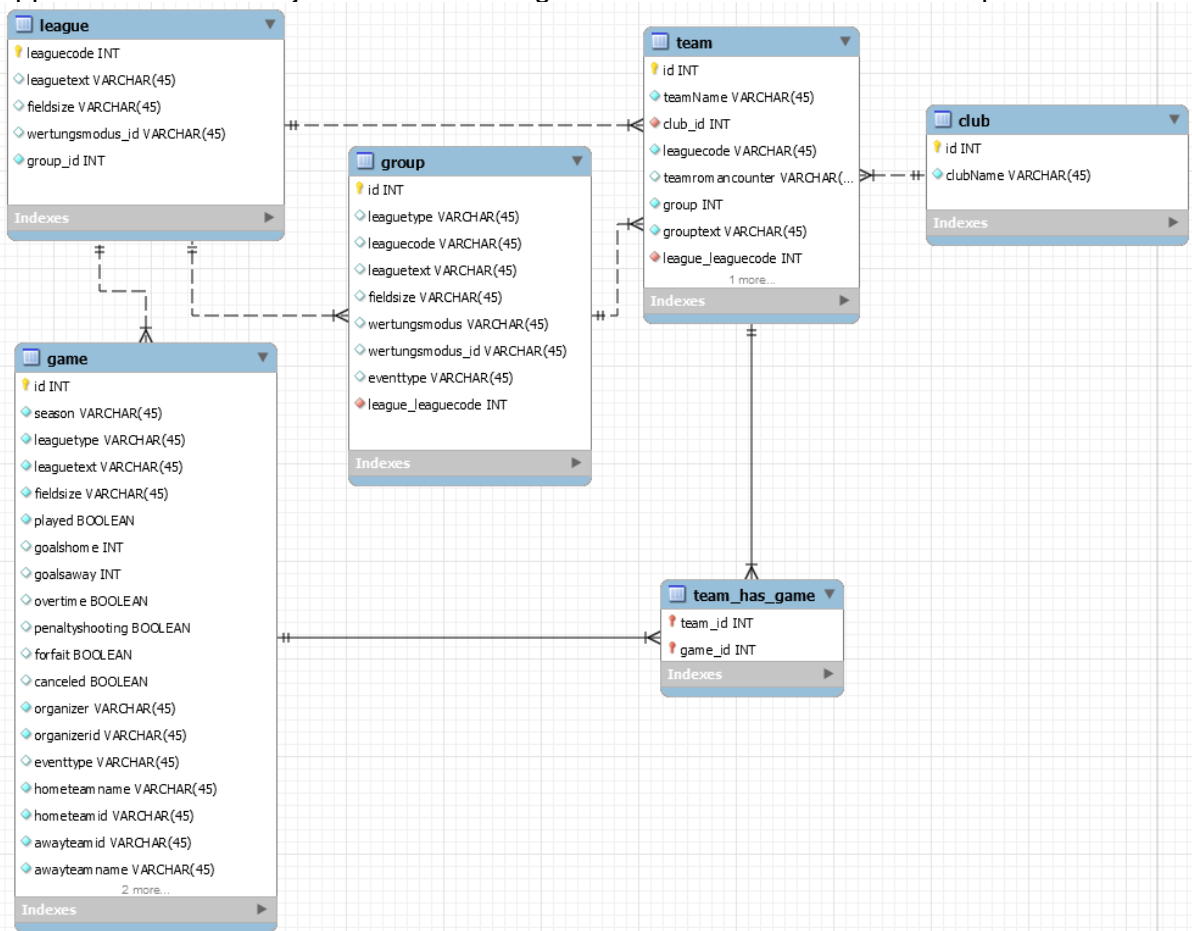
Das folgende Diagramm stellt die Komponente Service dar wie wir sie in unserer Applikation verwenden.

(Das Diagramm zeigt die Klassen und Methoden nicht vollständig dar. Z.B. wurde zur Illustration im UnihockeyDataBinder lediglich die Methode getTeams erwähnt. In der Implementation kommen jedoch noch weitere dazu).



2.2 Datenmodell

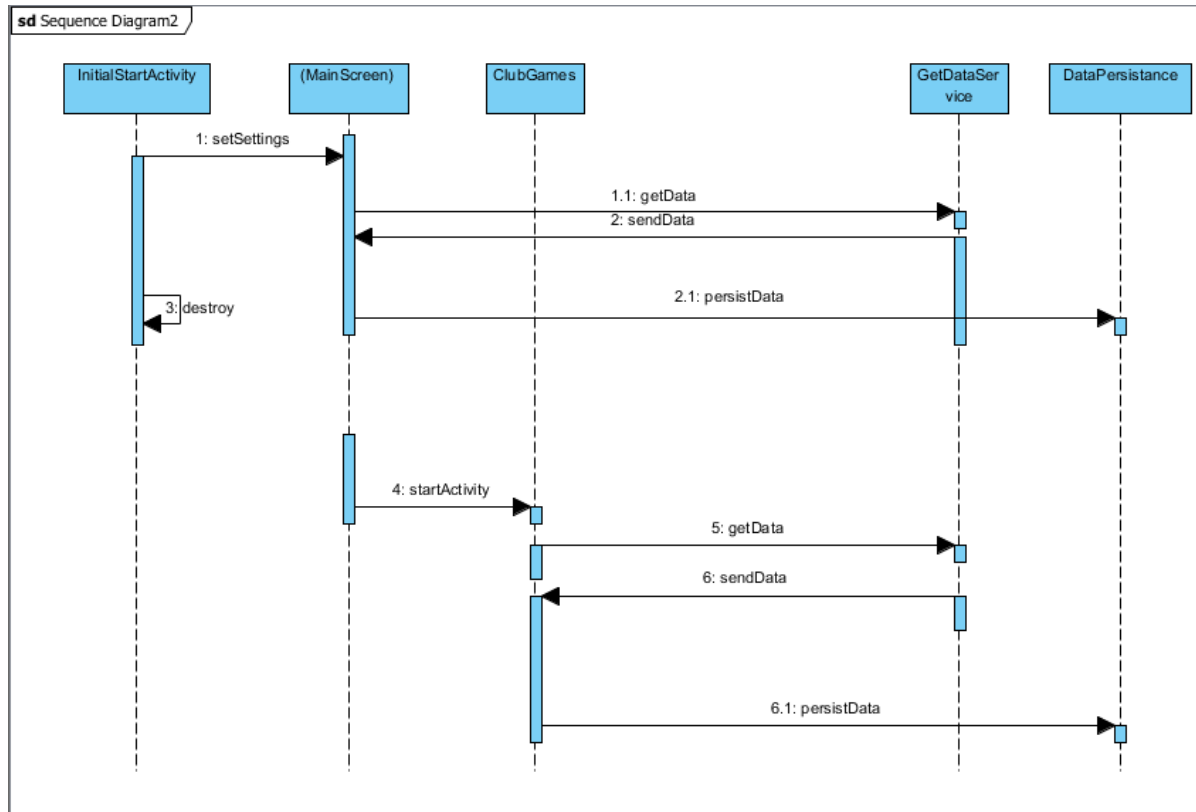
Das folgende Diagramm zeigt die Tabellen unserer in der Applikation integrierten Datenbank. Die Tabellen sind so definiert, dass sie ohne grosse Veränderungen von der REST Schnittstelle abgeholt und in der Applikation gespeichert werden können. Die Applikation kann dann jederzeit darauf zugreifen und bei Bedarf die Daten updaten.



2.3 Dynamische Modelle

2.3.1 Sequenzdiagramm

Das folgende Sequenzdiagramm zeigt wie die Applikation mit internen Komponenten kommunizieren soll.



2.3.2 Sequenzdiagramm UnihockeyDataService

Das folgende Diagramm versucht zu veranschaulichen wie die Daten mittels dem UnihockeyDataService von dem Webservice bezogen werden sollen.

