

XYZ

Gliederung

1. Team
2. Projekt Idee
3. User Interface
4. Use Cases
5. Schnittstellen
6. Workflow
7. Verwendete Pattern
8. Technologien
9. ASP.NET Core
10. Probleme
11. Live Demo
12. Blick in die Zukunft

1 Team

X

- Reports, WalkThroughs

Y

- CardBoxes, Cards, Css

Z

- Statistics, Client Validierung


2 Projekt Idee

- Easy Studying ist eine Web-Anwendung die Studenten beim Lernen unterstützen soll
- Ein Karteikastensystem soll es dem Studenten erlauben, Wissensfragen zu beantworten
- Am Ende werden diese ausgewertet und ein Ergebnis wird geliefert
- Punkte System für beantworten von Fragen -> Ranking
- Melden von Karteikästen und Karten

3 User Interface

- Startseite

[EasyStudying](#) [Karteikästen](#) [Meine Karteikästen](#) [Reports](#) [Statistiken](#) Hallo Zeus! [Abmelden](#)



EasyStudying - Be part, learn smart [Registrieren](#)

Neu hinzugefügt		
Name	Kategorie	
Latein Für Anfänger	Latein	▶
Mathe 1	Mathematik	▶

Beliebteste Karteikästen		
Name	Kategorie	
Mathe 1	Mathematik	▶
Latein Für Anfänger	Latein	▶

Zuletzt bearbeitet		
Name	Kategorie	
Mathe 1	Mathematik	▶

© 2016 - EasyStudying

3 User Interface





- Karteikästen

[EasyStudying](#) [Karteikästen](#) [Meine Karteikästen](#) [Reports](#) [Statistiken](#) Hallo Zeus! [Abmelden](#)

Übersicht Karteikästen

Hauptkategorie ▼ Unterkategorie ▼ Suchen

Anzahl der Karteikästen 2

Name	Beschreibung	Hauptkategorie	Unterkategorie	Ersteller	Anzahl Klicks	Erstellungsdatum	
Mathe 1	Karteikasten für Mathe	Mathematik	Algebra	Zeus	2	09.07.2016	 
Latein Für Anfänger	Einstieg in eine tote Sprache.	Latein	Anfänger	Zeus	0	17.07.2016	 

« < 1 > »

© 2016 - EasyStudying

3 User Interface

- Statistik

EasyStudying

Karteikästen

Meine Karteikästen

Reports

Statistiken

Hallo Apollo!

Abmelden

Statistik

Suchen

Benutzer	Anzahl der Antworten	Anzahl der durchgearbeiteten Karteikästen	Anzahl der richtigen Antworten	Punkte	Zuletzt durchgearbeitet
Apollo	1	1	1	6	17.07.2016
Zeus	45	11	21	76	16.07.2016
Apollo	1	1	1	6	17.07.2016

«

<

1

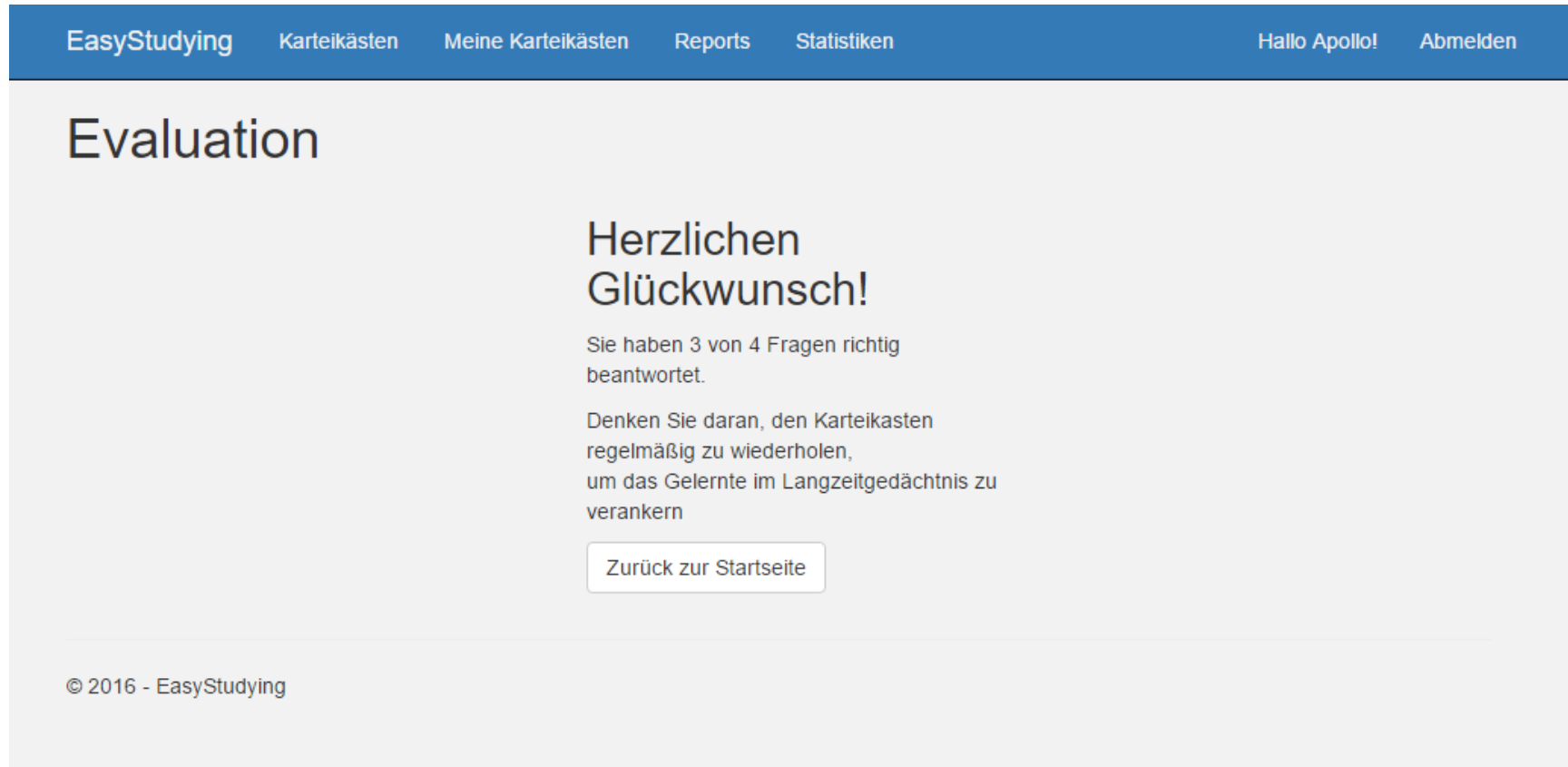
>

»

© 2016 - EasyStudying

3 User Interface

- Evaluation



3 User Interface

- Karteikasten durcharbeiten

EasyStudying

Karteikasten

Meine Karteikasten

Reports

Statistiken

Hallo Apollo!

Abmelden

Kategorie: Mathematik

Unterkategorie: Algebra

Karteikasten: Mathe 1

Karteikasten schließen

Karte melden

Tipp

Frage:

Wieviel ist 5 x 5?

Antwort:

☐ 10

☒ 25

☐ 20

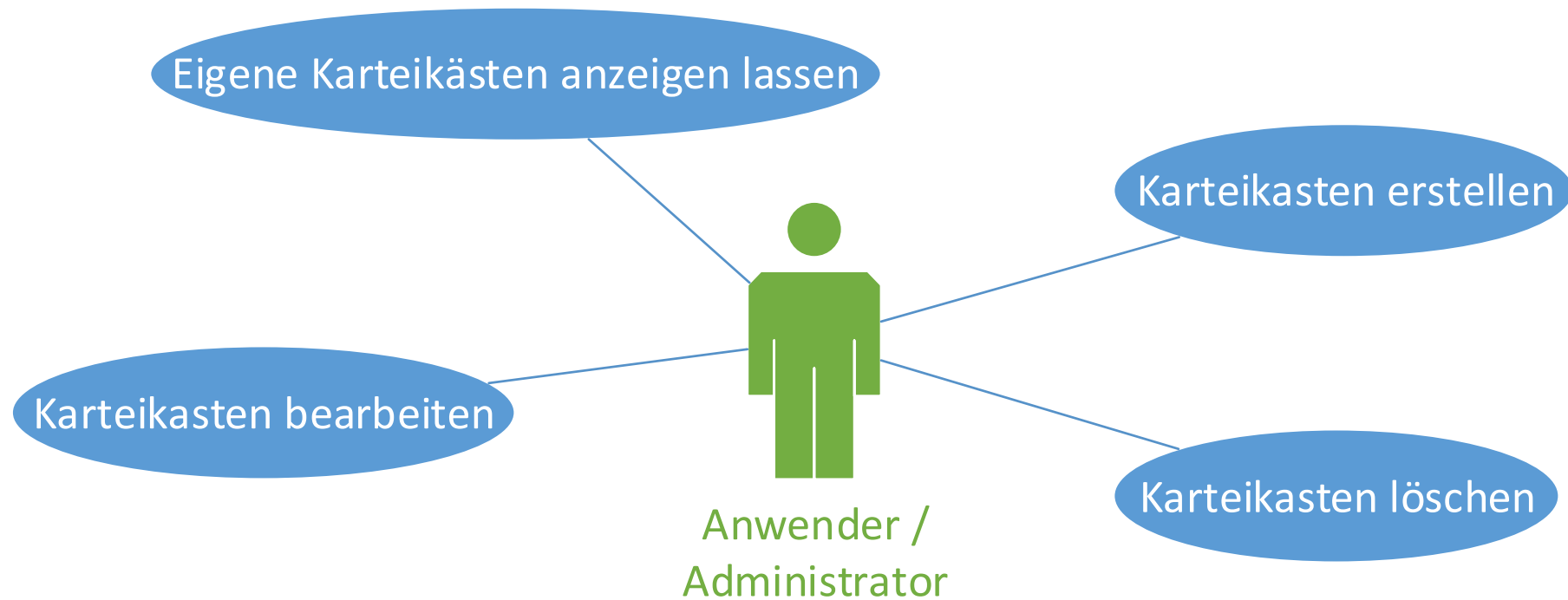
☐ 15

Prüfen

1 / 4

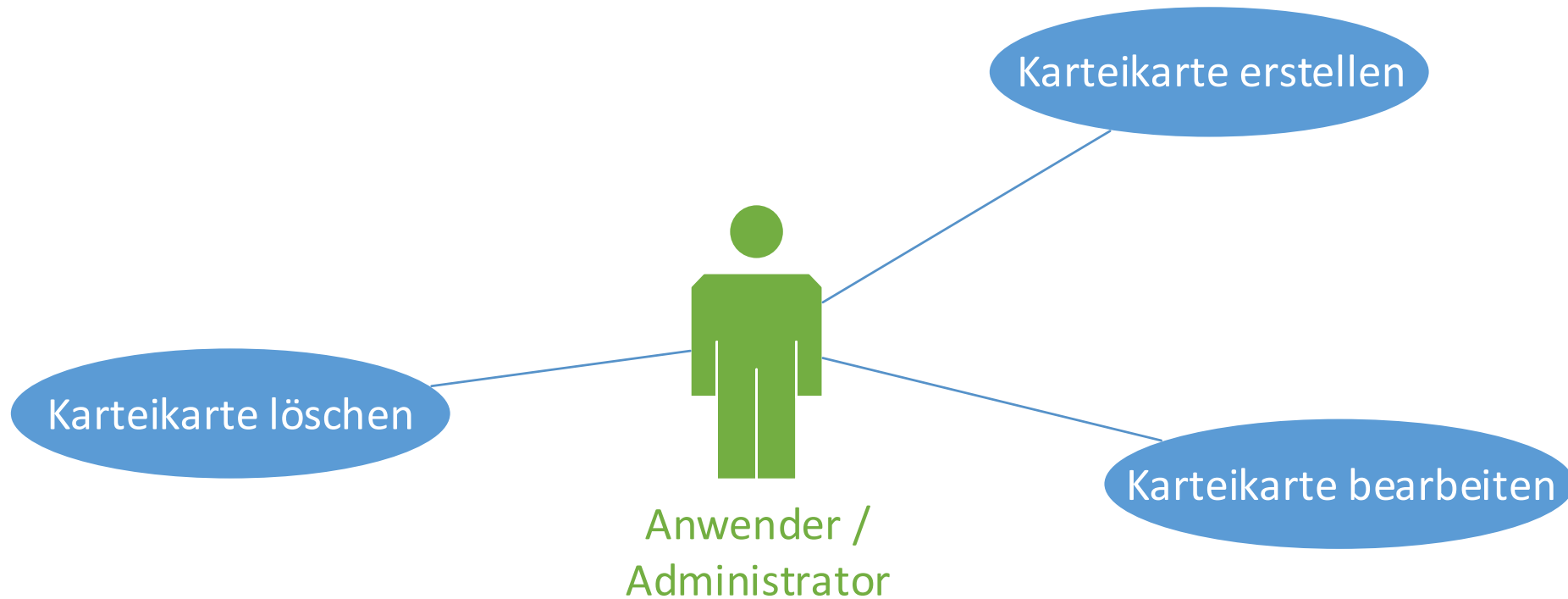
4 Use Cases

- Meine Karteikästen



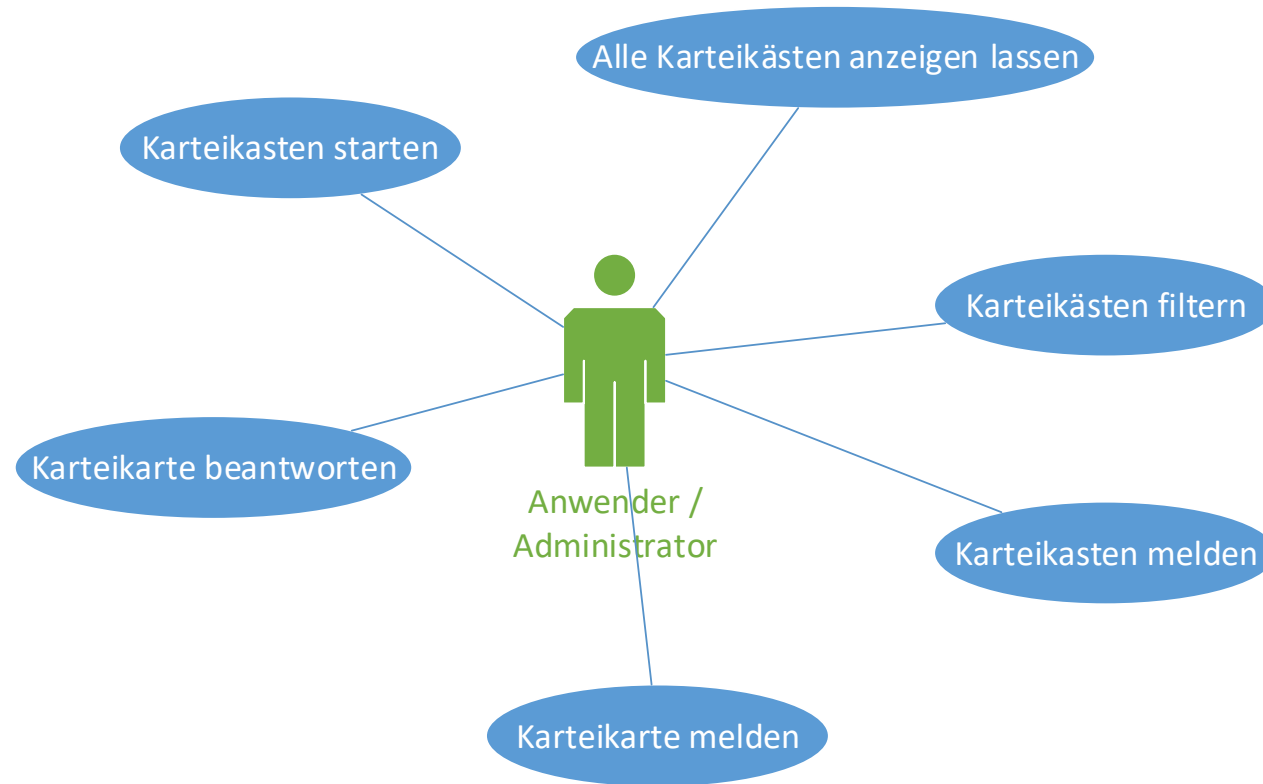
4 Use Cases

- Karteikarten



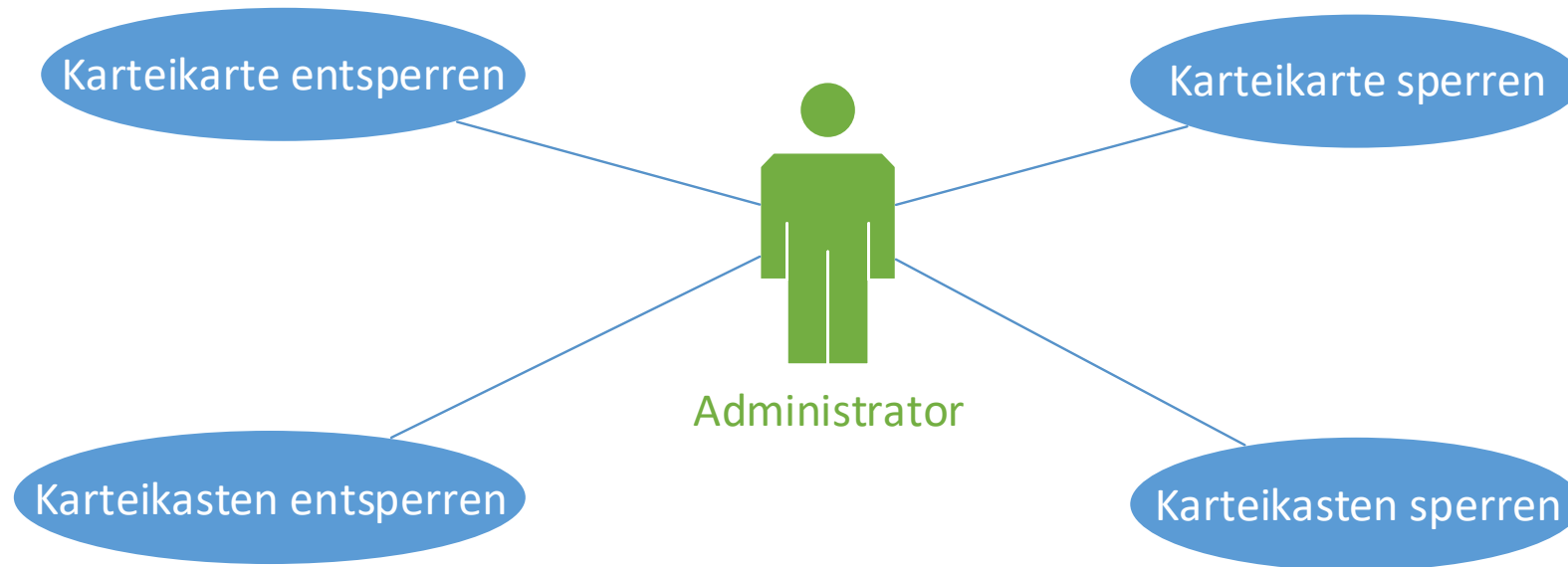
4 Use Cases

- Karteikästen (Allgemein)



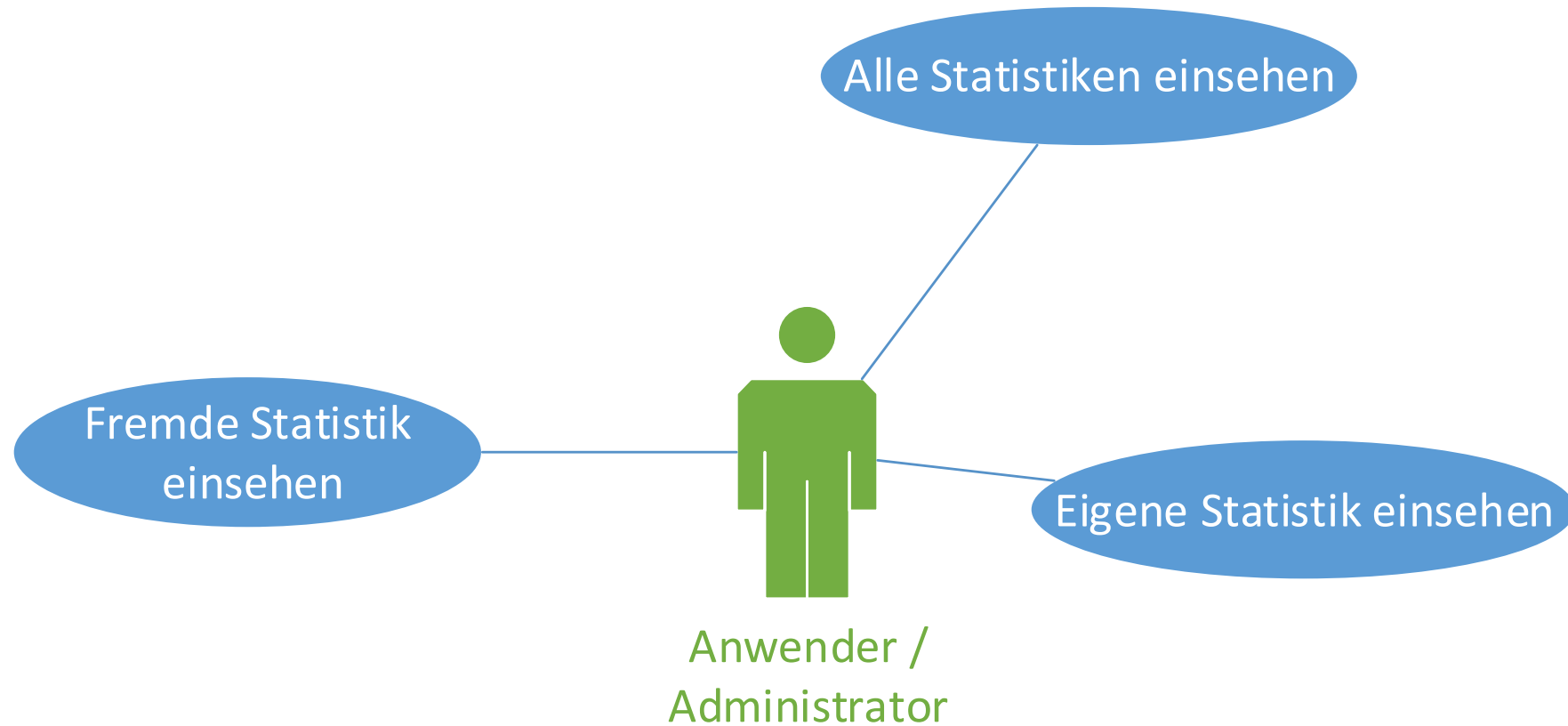
4 Use Cases

- Sperrfunktionen



4 Use Cases

- Statistiken



5 Schnittstellen

- CardBoxesController

CardBoxesController	
Get	CardBoxes/Index {int: page}
Post	CardBoxes/Index {string: searchString, string: mainCategories, string: subCategories, int: page}
Get	CardBoxes/MyCardBoxes {int: page}
Get	CardBoxes/Create
Post	CardBoxes/Create{ CardBox: cardBox}
Get	CardBoxes/Edit{Guid: id}
Post	CardBoxes/Edit{Guid: id, CardBox: cardBox}
Get	CardBoxes/Delete{Guid: id}
Get	CardBoxes/ChangeLock{ Guid: id}
Post	CardBoxes/ChangeLock{ CardboxModel: cardbox}

5 Schnittstellen

- CardController

CardsController	
Get	Cards/Index{Guid: cardBoxID, int: page}
Get	Cards/Create{Guid: id}
Post	Cards/Create{Guid: id, CardModel: card}
Get	Cards/Edit{Guid: id}
Post	Cards/Edit{Guid: id, CardModel card}
Get	Cards/Delete{Guid: id}
Post	Cards/DeleteConfirmed{Guid: id}

5 Schnittstellen

- StatisticController

StatisticsController

Get

Statistics/Index {int: page}

Post

Statistics/Index {string searchString, int: page}

5 Schnittstellen

- ReportsController

ReportsController	
Get	Reports/Index {int: page}
Get	Reports/ReportCardBox{Guid: cardBoxID}
Post	Reports/ReportCardBox{ReportModel: report}
Get	Reports/ReportCard{Guid: cardBoxID, Guid: cardId}
Post	Reports/ReportCard{ReportModel: report}
Get	Reports/ChangeStatus{Guid: id}
Post	Reports/ChangeStatus{Report: report}

5 Schnittstellen

- WalkThroughController

WalkThroughController

Get

WorkThrough/Index {Guid: id}

Post

WorkThrough/Index{Guid: id,
CardQuestionViewModel: model}

Post

WorkThrough/ValidateCard{Guid: id,
CardQuestionViewModel: model}

Get

WorkThrough/Evaluation

6 Workflow

Startseite

1. User kann einen der beliebtesten, neuesten oder zuletzt bearbeiteten Karteikästen auswählen
2. User kann über Navbar auf Karteikästen, eigene Karteikästen, Reports und Ranking gehen

Reports

1. User sieht alle Reports, die von ihm erstellte Karteikästen bzw. Karteikarten betreffen
2. Admin sieht alle Reports und kann deren Status ändern bzw. die Karteikästen/-karten sperren und entsperren

6 Workflow

Eigene Karteikästen

1. User sieht nur selbst erstellte Karteikästen
2. User kann neuen Karteikasten erstellen
3. User kann beliebigen gelisteten Karteikasten bearbeiten, löschen und sich sämtliche Fragen eines beliebigen gelisteten Karteikasten ansehen

Karteikästen

1. User sieht alle verfügbaren Karteikästen
2. User kann diese Karteikästen filtern nach: Haupt- /Unterkategorie oder Name(Volltextsuche)
3. User kann beliebigen gelisteten Karteikasten melden
4. User kann beliebigen gelisteten Karteikasten wählen und somit durcharbeiten

6 Workflow

Karten

1. User kommt auf diese Ansicht nur über „myCardBoxes“ → Karteikarten ansehen
2. User sieht alle Karteikarten des zuvor gewählten Karteikastens
3. User kann neue Karteikarten hinzufügen
4. User kann beliebige gelistete Karte bearbeiten und löschen

Karteikasten erstellen

1. User gibt Daten zu Karteikasten ein
2. User kann als nächstes die Nächste Karteikarten erstellen
3. Anschließend hat er die Möglichkeit, eine nächste Karteikarte zu erstellen oder den Karteikasten zu speichern

6 Workflow

Durcharbeiten

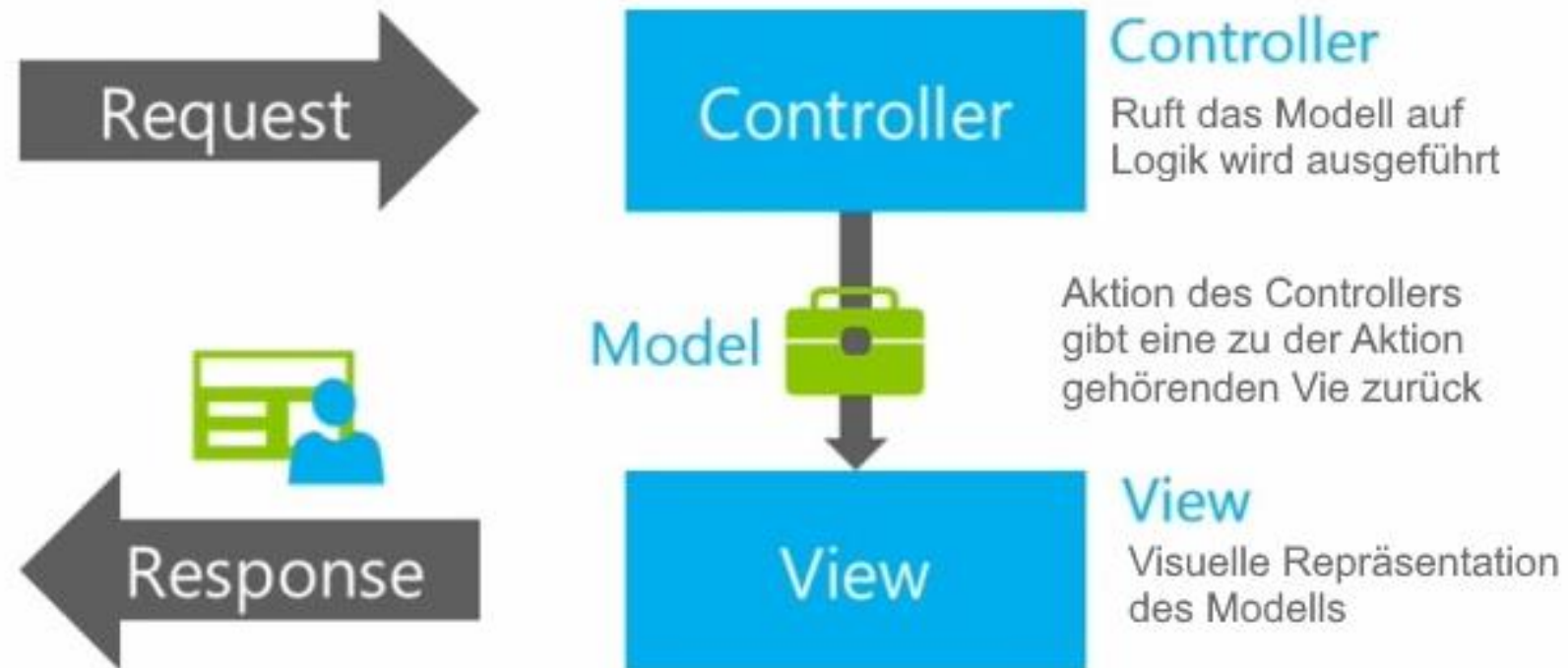
1. User bekommt Fragen des zuvor gewählten Karteikastens nacheinander angezeigt
 - Darauf folgt eine Auswertung

Statistik

1. User kann eigene Statistik und Statistik aller User einsehen
2. User kann nach User Statistiken suchen

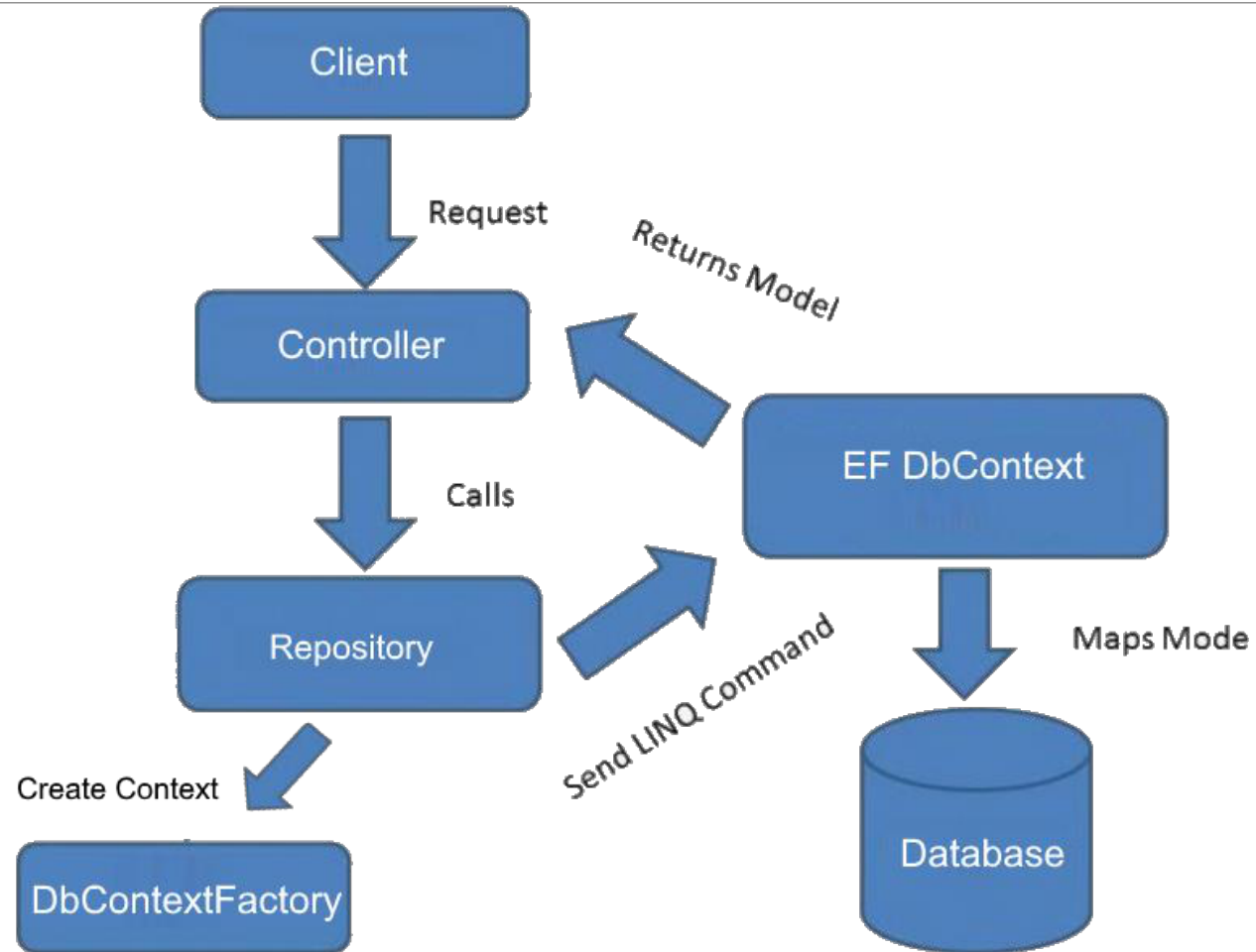
6 Verwendete Pattern

- MVC

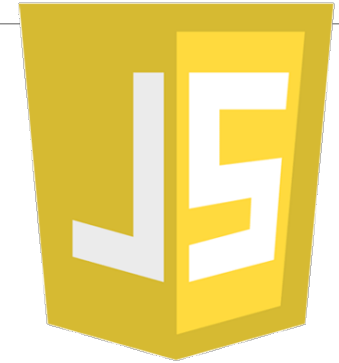


7 Verwendete Pattern

- Repository & Factory



8 Technologien



ASP.NET Core



CSS



9 ASP.NET Core

- Open source
- Entwicklung und Lauffähigkeit ist Plattform übergreifend
- Microsoft entwickelt ASP.NET Core über GitHub: <https://github.com/aspnet/home>
- Entwickeln von Cloud basierten Webapplikationen, IoT Applicationen und Moblie Backend
- 1.0.0 Release an 27.06.2016
- Vorteile von ASP.NET Core
 - Leichte Integration von modernen Client-Side Frameworks (Gulp, Grunt, Bootsrap, jQuery, usw.)
 - Eingebaute dependency injection
 - Baut auf modernen Technologien auf
 - Community steht im Mittelpunkt (Open source, contributions usw.)

10 Probleme

- Umstieg von RC2 -> 1.0.0
- Neue Technologie
 - Viele Packages von ASP.NET nicht verwendbar
- Zeit
- Zu Beginn Migration Probleme der Datenbank

11 Live Demo



12 Blick in die Zukunft

- Community Features
- Optimierung der Admin Tools
- Meldungen Benutzer freundlicher machen
- Workflows optimieren
- Frontend verschönern – Responsive
- Claims integrieren
- SignalR für Meldungen
- Überarbeitung Punktevergabe