Elaboration

## Projektmanagement

#### Inception

|  |  |  |  |
| --- | --- | --- | --- |
|  | Benjamin Hohl | BH |  |
|  | Florian Bosshard | FB |  |
|  | Nadri Mamuti | NM |  |
|  | Sebastian Sprenger | SS |  |
|  |  |  |  |
| Mitarbeiter | Bezeichnung | Soll [h] | Ist [h] |
| Gruppe | Projektskizze | 8 | 8 |
| Gruppe | Projektskizze Präsentation | 3 | 4 |
| NM | UML für Eclipse einrichten | 1 | 2 |
| SS | Github für Eclipse einrichten | 2 | 2 |
| FB & BH | Eclipse Projekt aufsetzen | 1 | 1 |
| SS | Java Framework Slik anschauen | 2 | 1 |
|  |  |  |  |
| Total |  | 17 | 18 |

#### Elaboration

|  |  |  |  |
| --- | --- | --- | --- |
| Mitarbeiter | Bezeichnung | Soll [h] | Ist [h] |
| NM | Projektmanagement erstellen | 1 | 1 |
| NM | Anwendungsfälle schreiben | 4 | 3 |
| SS | eine erste Architektur | 2 | 2 |
| BH | zusätzliche Spezifikationen | 2 | 2 |
| NM | Anwendugsfalldiagramm | 1 | 1 |
| FB | System-Sequenzdiagramm | 2 | 2 |
| FB | Systemoperationen | 0.5 | 1 |
| SS | Domänenmodell | 4 | 3 |
| FB | Glossar | 1 | 1 |
| SS | Software Codierung | 4 | 4 |
|  |  |  |  |
| Total |  | 21.5 | 20 |

## Andwendugsfälle

### Fully dressed Anwendugsfall

#### „Zug ausführen“

**Hauptakteur:**

Spieler / Computer

**Stakeholders und ihre Interessen:**

* necaREx-Benuter: Möchte eine intuitive Benutzeroberfläche. Möchte seinen Zug nach den geltenden Schachregeln tätigen können. Benutzer welche mit den Schachregeln nicht sehr vertraut sind möchten das unzulässige Züge als solche angezeigt werden.
* Kunde: Möchte das seine Schüler mit necaREx ihre Schachfähigkeiten verbessern können. Möchte das die Schüler auf einfache Art Züge tätigen können.

**Vorbedingungen:**

Der Benutzer muss eine Figur ausgewählt haben und diese verschieben.

**Nachbedingungen:**

Der Benutzer hat den Zug getätigt und Figur verschoben.

**Haupterfolgsszenario:**

|  |  |
| --- | --- |
| Akteur:   1. Er wählt eine Figur aus   4. Er verschiebt die Figur | System:   1. Prüft welche Züge zulässig sind. 2. Stösst Regel-Engine an   5. Abfrage: Wird gegnerische Figur   geschlagen? 6. Allenfalls gegnerische Figur   entfernen. |

Dieses Szenario kann so oft wiederholt werden bis ein König geschlagen wurde und das Spiel beendet wird.

**Alternativer Ablauf:**

|  |  |
| --- | --- |
| Akteur:  6.b Akteur darf Zug nicht ausführen | System:  6.a Zug verstösst gegen Regeln |

**Häufigkeit des Auftretens:**

Dieser Anwendungsfall tritt auf solang das Spiel läuft.

### Casual Anwendungsfälle

#### „Regeln durchsetzten“

**Haupterfolgsszenario:**

Wählt der Benutzer eine Figur aus, wird die Regel-Engine angestossen. Diese überprüft welche Züge mit der gewählten Figur erlaubt sind. Zieht der Benutzer eine Figur auf ein unerlaubtes Feld, so wird die Figur auf das vorherige Feld zurückgesetzt.

**Alternatives Szenario:**

Wird einer der zwei Könige geschlagen, so wird das Spiel beendet.

#### „Figur wählen“

**Haupterfolgsszenario:**

Der Benutzer wählt eine Figur aus, mit der er den nächsten Zug tätigen möchte. Das System prüft ob die ausgewählte Figur dem Spieler gehört, welcher die Figur ausgewählt hat. Das Feld auf dem die ausgewählte Figur steht wird eingefärbt. Die möglichen Zielfelder werden ebenfalls eingefärbt.

**Alternatives Szenario:**

Falls die ausgewählte Figur nicht dem Spieler gehört, welcher die Figur ausgewählt hat wird die Figur wieder abgewählt.

#### „Neues Spiel starten“

**Haupterfolgsszenario:**

Sobald der Benutzer ein neues Spiel startet wird das Schachfeld initialisiert und die Figuren werden in Anfangsstellung bereitgestellt.

**Alternatives Szenario:**

Falls das Spiel nicht neu gestartet werden kann, wird vom System eine Fehlermeldung erzeugt.

### Brief Anwendungsfälle

#### „Gegenspieler selektieren“

Der Benutzer wählt seinen Gegenspieler, entweder menschlicher Gegenspieler oder Computerspieler aus.

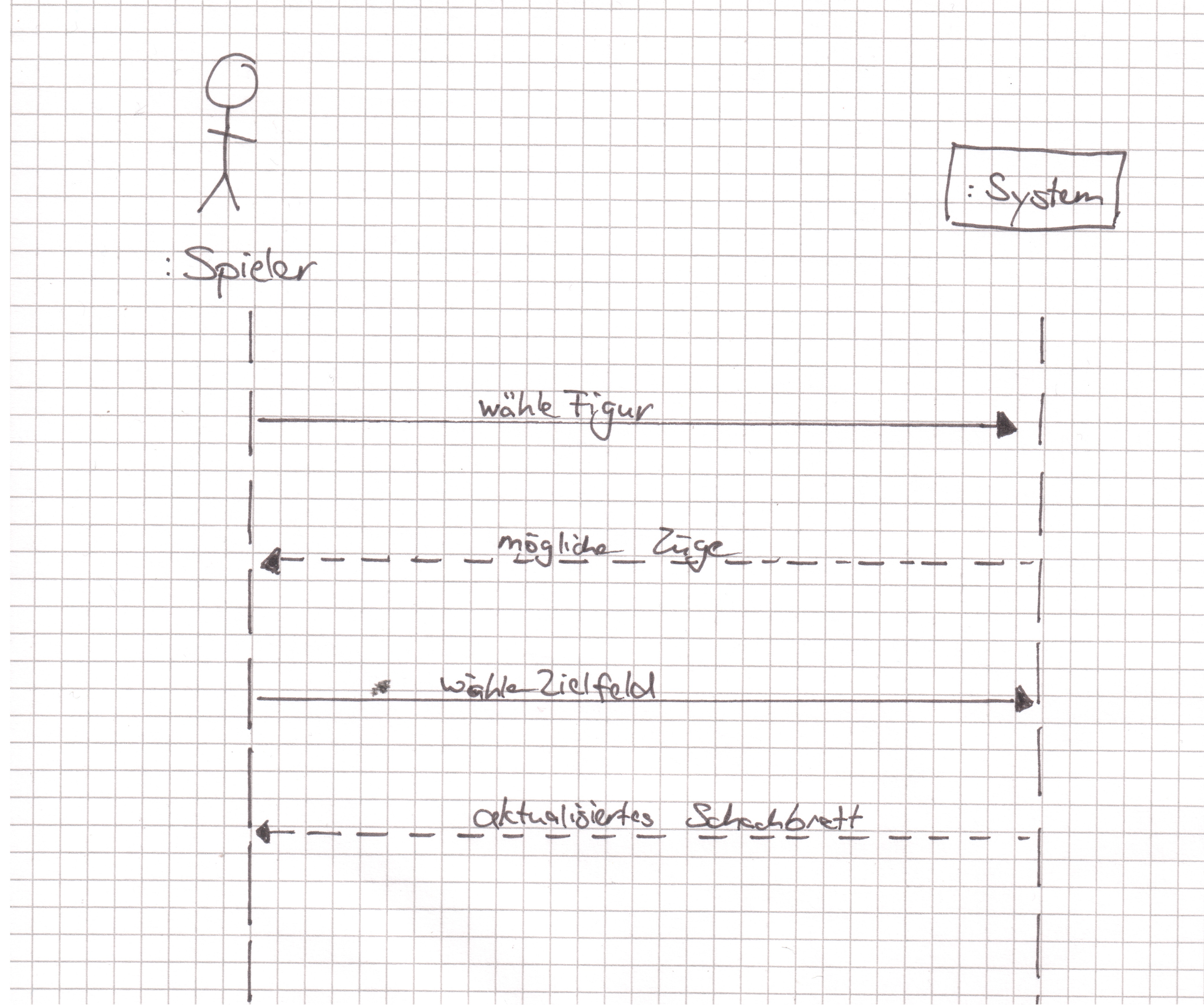
## Eine erste Architektur

## Zusätzliche Spezifikationen

## Anwendungsfalldiagramm



## System-Sequenzdiagramm



## Systemoperationen

## Domänenmodell

## Glossar