

Error 404

Informatik Projektarbeit

Der grosse Dalmuti

vorgelegt am 1. November 2013

von

Elias Buff

Jonas Däster

Theresa Ehrmann

Thomas Reichert

Inhaltsverzeichnis

1. Einleitung.....	1
2. Funktionale Anforderungen	2
3. Qualitätsanforderungen.....	3
4. Use Case Diagramm.....	4
5. Use-Case-Spezifikationen	5
6. GUI Mock-up	11
7. Testplan	12
8. Architektur.....	16
9. Klassendiagramm	17
10. Client/Server Schnittstellen und Messages	18
Anhang 1: Spielablauf Textform.....	19
Anhang 2: Spielablauf Flussdiagramm	21
Anhang 3: Glossar.....	25

1. Einleitung

Die Gruppe "Error 404" ist ein Kollektiv aus der Fachhochschule Nordwestschweiz. Mitglieder sind Theresa Ehrmann, Elias Buff, Jonas Däster und Thomas Reichert vom Studiengang für Wirtschaftsinformatik an der Hochschule für Wirtschaft in Olten.

Für das Informatikprojekt haben wir den Auftrag erhalten, das Spiel "Der grosse Dalmuti" von Amigo mit der Programmiersprache Java umzusetzen. "Der grosse Dalmuti" ist ein Kartenspiel, in dem es darum geht, möglichst oft die Runde als Erster zu beenden. Der genaue Spielablauf ist im Anhang dokumentiert.

Das Spiel wird mit einer Client-Server-Architektur aufgebaut. Um am Spiel teilzunehmen müssen die Spieler den Client herunterladen und installieren. Die Systemabhängigkeiten ergeben sich ausschliesslich auf die verschiedenen Betriebssysteme. Es werden keine zusätzlichen Programme oder Systeme auf das Spiel zugreifen.

In erster Linie wollen wir die Mindestanforderungen für einen reibungslosen Spielablauf implementieren. Durch eine genaue Dokumentation soll eine Weiterentwicklung problemlos möglich sein.

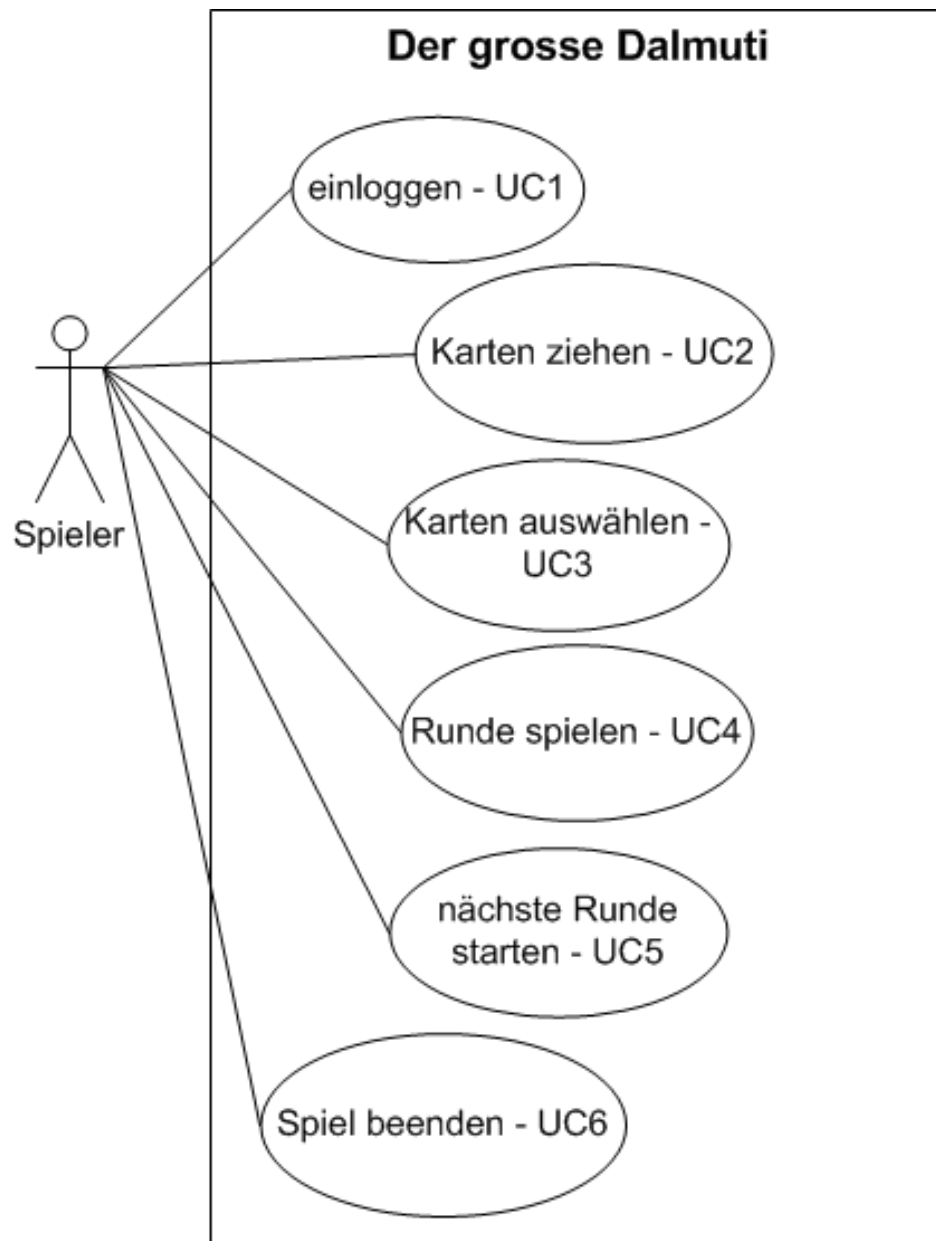
2. Funktionale Anforderungen

1. Der Spieler hat die Möglichkeit den Client aus dem Internet herunterzuladen und auf seinen Rechner zu installieren.
2. Der Spieler hat die Möglichkeit den Client zu starten.
3. Der Spieler hat die Möglichkeit einen Benutzernamen einzugeben und dem Spiel beizutreten.
4. Der Spieler hat die Möglichkeit unter Beachtung der Spielregeln:
 - a. Karten zu ziehen
 - b. Karten auszuwählen
 - c. Kartenauswahl zu bestätigen
 - d. zu passen
5. Die Spielregeln werden vom System überprüft und eingehalten.
6. Der Spieler hat die Möglichkeit das Spiel jederzeit zu beenden.
7. Der Spieler kann die eigenen Karten einsehen.
8. Der Spieler kann die ausgespielten Karten der Mitspieler auf dem GUI sehen.
9. Der Spieler kann mehrere Runden spielen.
10. Das System kann Karten austeilen, wieder einsammeln und diese Karten durchmengen.
11. Das System kann die Rolle jedes Spielers festlegen und somit auch die sich daraus ergebende Spielreihenfolge.
12. Das System kann die chronologisch richtige Reihenfolge der einzelnen Spielschritte korrekt ausführen.
13. Das System kann Ränge speichern und daraus Rollen ableiten.
14. Das System kann Spieler deaktivieren.
15. Der Client muss von der Computerplattform unabhängig installierbar und funktionsfähig sein.

3. Qualitätsanforderungen

1. Das Herunterladen des Clients darf bei den heute in der Schweiz üblichen Internetanschlüssen nicht länger als 5 Minuten dauern.
2. Die Installation des Clients darf bei den heute handelsüblichen Rechnern nicht länger als 5 Minuten dauern.
3. Das Starten des Clients darf bei den heute handelsüblichen Rechnern nicht länger als 30 Sekunden dauern.
4. Das Loginverfahren bis zum Start des Spiels, inklusive warten auf andere Spieler, darf nicht länger als 2 Minuten dauern.
5. Jedem Spieler müssen, die für die Runde oder den Spielzug relevanten Spielkarten, innert 2 Sekunden angezeigt werden.
6. Die Rollenverteilung muss innert maximal 2 Minuten abgeschlossen sein und danach für jeden Spieler sichtbar werden.
7. Ein Spielzug muss innert 60 Sekunden abgeschlossen werden.
8. Das Beenden des Spiels darf nicht länger als 5 Sekunden dauern.
9. Das Spiel muss in der Programmiersprache Java programmiert sein.
10. Der von den Spielern eingegebene Benutzername darf nicht aus mehr als 10 Zeichen bestehen, darf nur Buchstaben(A-Z) und Ganzzahlen(1-9) ohne Sonderzeichen beinhalten und muss von den Benutzernamen der bereits eingeloggtten Spieler in mindestens einem Zeichen abweichen.
11. Die GUI wird korrekt angezeigt und ist intuitiv verständlich.
12. Die Überprüfung und Einhaltung der Spielregeln erfolgt fehlerfrei.

4. Use Case Diagramm



5. Use-Case-Spezifikationen

Abschnitt	Inhalt
Use-Case-Nr.	1
Name	einloggen
Priorität	hoch
Kurzbeschreibung	Spieler muss sich mit einem beliebigen Benutzernamen einloggen können
Auslösendes Ereignis	Spieler will das Spiel spielen
Akteure	Spieler
Vorbedingung	Client heruntergeladen, installiert und gestartet
Nachbedingung	Spieler befindet sich im Spielraum
Hauptszenario	1. Spieler gibt im GUI einen beliebigen Benutzernamen ein 2. Spieler bestätigt seine Eingabe 3. System überprüft den Benutzernamen auf Einmaligkeit 4. Systemprüfung gültig -> Spieler ist eingeloggt
Alternativszenarien	-
Ausnahmeszenarien	4a. Systemprüfung ungültig -> Spieler gibt einen anderen Benutzernamen ein 5a. Spieler bestätigt seine Eingabe 6a. System überprüft erneut 7a. Systemprüfung gültig -> Spieler ist eingeloggt 7b. Systemprüfung ungültig -> weiter mit 4a
Qualitäten	QA 1, 2, 3, 4, 10

Abschnitt	Inhalt
Use-Case-Nr.	2
Name	Karten ziehen
Priorität	hoch
Kurzbeschreibung	Spieler zieht eine verdeckte Karte vom Spieltisch
Auslösendes Ereignis	Spielkarten sind auf dem Spieltisch verdeckt ausgelegt
Akteure	Spieler
Vorbedingung	1. Spieler ist eingeloggt 2. Spielkarten sind auf dem Spieltisch verdeckt ausgelegt
Nachbedingung	Rollen sind zugewiesen
Hauptszenario	<pre> graph TD Start(()) --> A[Jeder Spieler zieht nacheinander eine Karte] A --> B[Gezogene Karten werden offengelegt] B --> C{Sind ALLE offengelegten Karten unique?} C -- Ja --> D[Rollen werden anhand des Kartenwerts den Spielern zugewiesen] D --> End(()) C -- Nein --> E{Sind unique Karten vorhanden?} E -- Ja --> F[Spieler mit der unigen Karte wird Rolle zugewiesen] F --> G[Spieler mit gleichen Karten, Müssen nochmals ziehen] G --> A E -- Nein --> H[Alle Karten werden eingezogen] H --> A </pre>
Alternativszenarien	-
Ausnahmeszenarien	-
Qualitäten	QA 6, 12

Abschnitt	Inhalt
Use-Case-Nr.	3
Name	Karten auswählen
Priorität	hoch
Kurzbeschreibung	Karten auswählen, die getauscht werden sollen
Auslösendes Ereignis	Spieler will das Spiel beginnen
Akteure	Spieler
Vorbedingung	1. Die Rolle muss dem Spieler zugeteilt sein 2. Karten müssen an jeden Spieler verteilt sein 3. Karten müssen für den jeweiligen Spieler ersichtlich sein
Nachbedingung	Spieler ist bereit für dem ersten Spielzug
Hauptszenario	<pre> graph TD Start(()) --> Fork Fork --> UC1[Gro. Dalmuti wählt 2 beliebige Karten aus] Fork --> UC2[Kl. Dalmuti wählt 1 beliebige Karte aus] Fork --> UC3[Gro. Diener wählt die 2 besten Karten aus] Fork --> UC4[Kl. Diener wählt die beste Karte aus] UC1 --> Join UC2 --> Join UC3 --> Join UC4 --> Join Join --> UC5[Ausgewählte Karten werden gemäss Spielregeln getauscht] UC5 --> End((())) </pre> <p>The diagram illustrates the main scenario for selecting cards. It begins with a start node leading to a fork that splits into four parallel use cases: 'Gro. Dalmuti wählt 2 beliebige Karten aus', 'Kl. Dalmuti wählt 1 beliebige Karte aus', 'Gro. Diener wählt die 2 besten Karten aus', and 'Kl. Diener wählt die beste Karte aus'. These four use cases converge at a join node, which then leads to a final use case 'Ausgewählte Karten werden gemäss Spielregeln getauscht'. The process concludes at an end node.</p>
Alternativszenarien	-
Ausnahmeszenarien	Spieler wählt ungültige Karten aus -> Fehlermeldung -> nochmals auswählen
Qualitäten	QA 12

Abschnitt	Inhalt
Use-Case-Nr.	4
Name	Runde spielen
Priorität	hoch
Kurzbeschreibung	Spieler spielt eine Runde
Auslösendes Ereignis	Spieler ist an der Reihe seinen Spielzug durchzuführen
Akteure	Spieler
Vorbedingung	Spieler muss noch Spielkarten besitzen
Nachbedingung	Die Runde ist beendet
Hauptszenario	<pre> graph TD Start(()) --> NextPlayer[Nächster Spieler beginnt] NextPlayer --> PlayCards{Karten spielen?} PlayCards -- Ja --> SelectCards[Mögliche Karten auswählen] PlayCards -- Nein --> Pass[Passen (freiwillig oder zwingend)] Pass --> Consecutive{3x hintereinander gepasst?} Consecutive -- Ja --> StichEndet[Stich endet] StichEndet --> RemoveCards[Ausgelegte Karten werden weggenommen] RemoveCards --> NextPlayer Consecutive -- Nein --> Deactivate1[Spieler wird für aktuelle Runde deaktiviert] SelectCards --> Accept[Auswahl akzeptieren] Accept --> NoCards{Keine Karten mehr?} NoCards -- Ja --> Deactivate2[Spieler wird für aktuelle Runde deaktiviert] NoCards -- Nein --> Winner1[Gewinner der Runde] Deactivate1 --> Deactivated1{1. deaktivierter Spieler?} Deactivated1 -- Ja --> Winner1 Deactivated1 -- Nein --> Deactivated2{2. deaktivierter Spieler?} Deactivated2 -- Ja --> Winner2[2. Platzierter der Runde] Deactivated2 -- Nein --> Winner3[3. Platzierter der Runde] Winner3 --> Loser[Der übrige Spieler wird Verlierer] Loser --> End(()) Winner1 --> NextPlayer Winner2 --> NextPlayer Winner3 --> NextPlayer </pre>
Alternativszenarien	-
Ausnahmeszenarien	Sind im Diagramm des Hauptszenarios enthalten
Qualitäten	QA 5, 7, 12

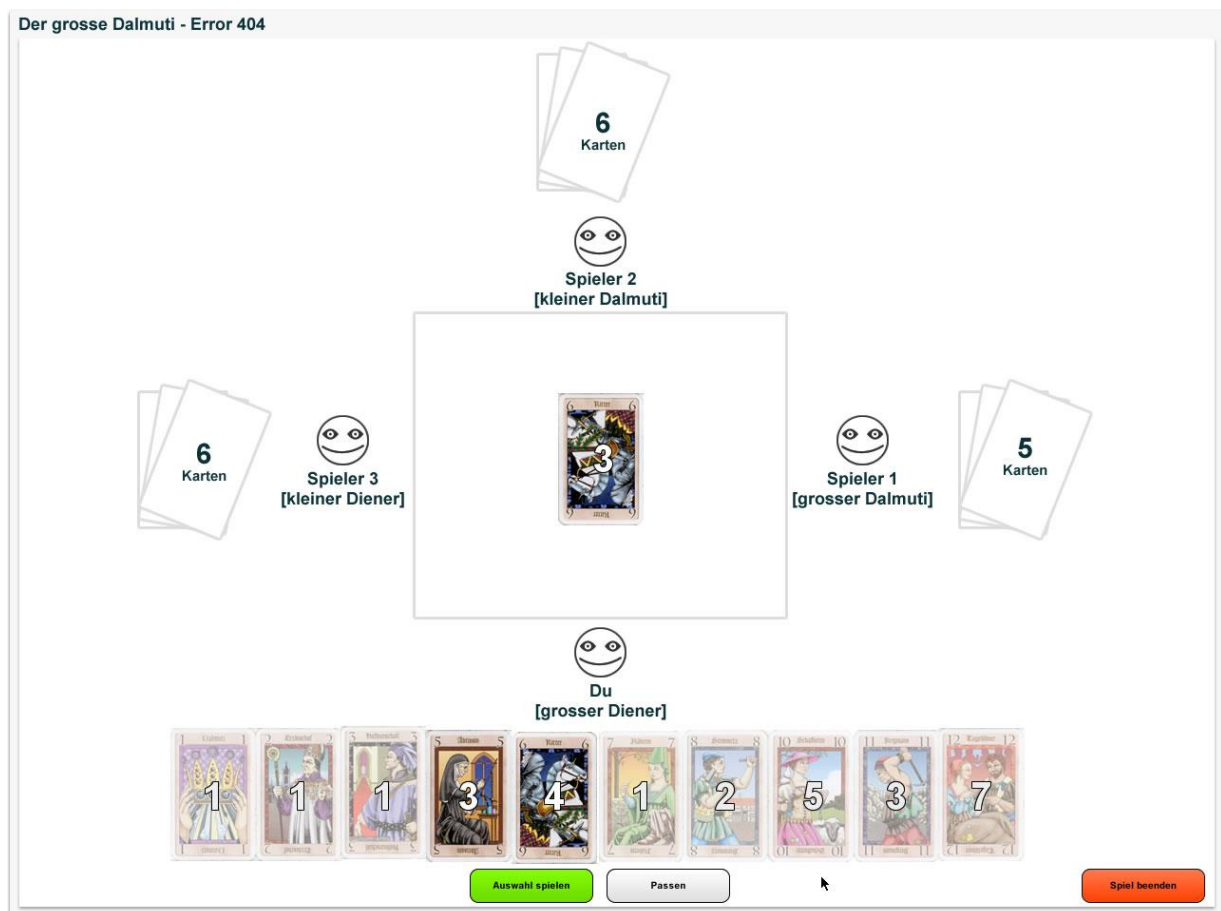
Abschnitt	Inhalt
Use-Case-Nr.	5
Name	Nächste Runde starten
Priorität	mittel
Kurzbeschreibung	Spieler hat die Möglichkeit nach jeder gespielter Runde, an neuer Runde teilzunehmen
Auslösendes Ereignis	Die Runde ist zu ende
Akteure	Spieler
Vorbedingung	Jedem Spieler ist der Rang zugewiesen
Nachbedingung	Nächste Runde startet
Hauptszenario	<ol style="list-style-type: none"> 1. Spieler klickt den Button „weitere Runde“ 2. die Rolle für die nächste Runde wird gemäss dem erzielten Rang dem Spieler zugewiesen 3. Karten werden auf die Spieler verteilt
Alternativszenarien	-
Ausnahmeszenarien	Klickt nicht jeder Spieler auf den Button „weitere Runde“ endet das Spiel
Qualitäten	-

Abschnitt	Inhalt
Use-Case-Nr.	6
Name	Spiel beenden
Priorität	hoch
Kurzbeschreibung	Spieler muss das Spiel verlassen können
Auslösendes Ereignis	Die Runde ist zu ende
Akteure	Spieler
Vorbedingung	Client gestartet
Nachbedingung	Spiel ist beendet
Hauptszenario	1. Spieler klickt den Button „Spiel beenden“ 2. Spiel ist beendet und Spieler ist ausgeloggt
Alternativszenarien	-
Ausnahmeszenarien	Schliessen des Clients während dem Spiel
Qualitäten	QA 8

vom Spieler ausgeführt

vom System ausgeführt

6. GUI Mock-up



7. Testplan

In diesem Teil der Arbeit werden verschiedene Testszenarien auf Basis der Use Cases erstellt. Diese gliedern sich in das Test Objekt (was?), die Test Methode (wie?) und das Test Ergebnis.

Test Objekt	Login
Test Methode	1. Vier Spieler loggen sich mit verschiedenen beliebigen Benutzernamen ein. 2. Vier Spieler loggen sich mit identischen Benutzernamen ein.
Test Ergebnis	1. Vier Spieler befinden sich mit korrektem Benutzernamen im Spielraum 2. Drei Spieler werden aufgefordert einen anderen Benutzernamen zu wählen.

Test Objekt	Karten ziehen
Test Methode	1. Karten werden verdeckt auf dem Spieltisch angezeigt. 2. Spieler wählen Karten aus und diese werden angezeigt.
Test Ergebnis	1. GUI Anzeige ist korrekt. 2. Rollen werden gemäss Kartenwert zugewiesen.

Test Objekt	Karten auswählen
Test Methode	1. Gemäss den Rollen wählen die Spieler Karten aus ihrer Hand, welche getauscht werden sollen. 2. Nach dem Tausch, werden die ausgewählten und ausgetauschten Karten verglichen.
Test Ergebnis	1. Karten wurden gemäss Spielregeln korrekt ausgewählt und getauscht. 2. Karten werden korrekt angezeigt.

Test Objekt	Runde spielen
Test Methode	1. Gewünschte Karten auswählen und spielen (Mehrfachauswahl möglich). 2. Passen 3. Gewünschte Karten auswählen und spielen (Mehrfachauswahl möglich). Der Spieler hat keine Karten mehr auf der Hand.
Test Ergebnis	1. Karten werden von der Hand gespielt. Nächster Spieler ist am Zug. 2. Nichts passiert. Nächster Spieler ist am Zug. 3. Spieler wird deaktiviert. Rangzuweisung erfolgt korrekt.

Test Objekt	Nächste Runde starten
Test Methode	1. Vier Spieler klicken den Button für die nächste Runde. 2. weniger als vier Spieler klicken den Button für die nächste Runde.
Test Ergebnis	1. Nächste Runde startet und die Rollen werden gemäss Rang korrekt zuge- teilt. 2. Spiel ist beendet.

Test Objekt	Spiel beenden
Test Methode	1. Spieler klickt den Button Spiel "beenden".
Test Ergebnis	1. Spiel ist beendet.

In diesem Teil der Arbeit werden verschiedene Testszenarien auf Basis der Qualitätsanforderungen erstellt.

Test Objekt	Herunterladen des Clients
Test Methode	1. Client herunterladen und Downloadzeit messen.
Test Ergebnis	1. Downloadzeit <=5 Minuten

Test Objekt	Installation des Clients
Test Methode	1. Installieren des Clients und Installationszeit messen
Test Ergebnis	1. Installationszeit <= 5 Minuten

Test Objekt	Starten des Clients
Test Methode	1. Client starten und Zeit bis zur Anzeige der GUI messen
Test Ergebnis	1. Zeit bis zur Anzeige der GUI \leq 30 Sekunden

Test Objekt	Loginverfahren
Test Methode	Login durchführen und Zeit bis Spielbeginn messen
Test Ergebnis	Zeit von Login bis Spielbeginn \leq 2 Minuten

Test Objekt	Spielkartenanzeige
Test Methode	Messung der Zeit zwischen Spiel-, Runden-, Stichstart und Anzeige der Karten auf der GUI.
Test Ergebnis	Zeit \leq 2 Sekunden

Test Objekt	Rollenverteilung
Test Methode	1. Zeit zwischen dem Ziehen der Karte und dem Beginn des Spiels messen. 2. Zeit zwischen vier Spieler klicken den Button für die Nächste Runde und dem Start der nächsten Runde messen
Test Ergebnis	1.+2. Zeit \leq 2min

Test Objekt	Dauer Spielzug
Test Methode	1. Spielzug auslaufen lassen.
Test Ergebnis	1. Timer läuft korrekt von 60 Sekunden auf 0 Sekunden runter und der Spieler passt automatisch.

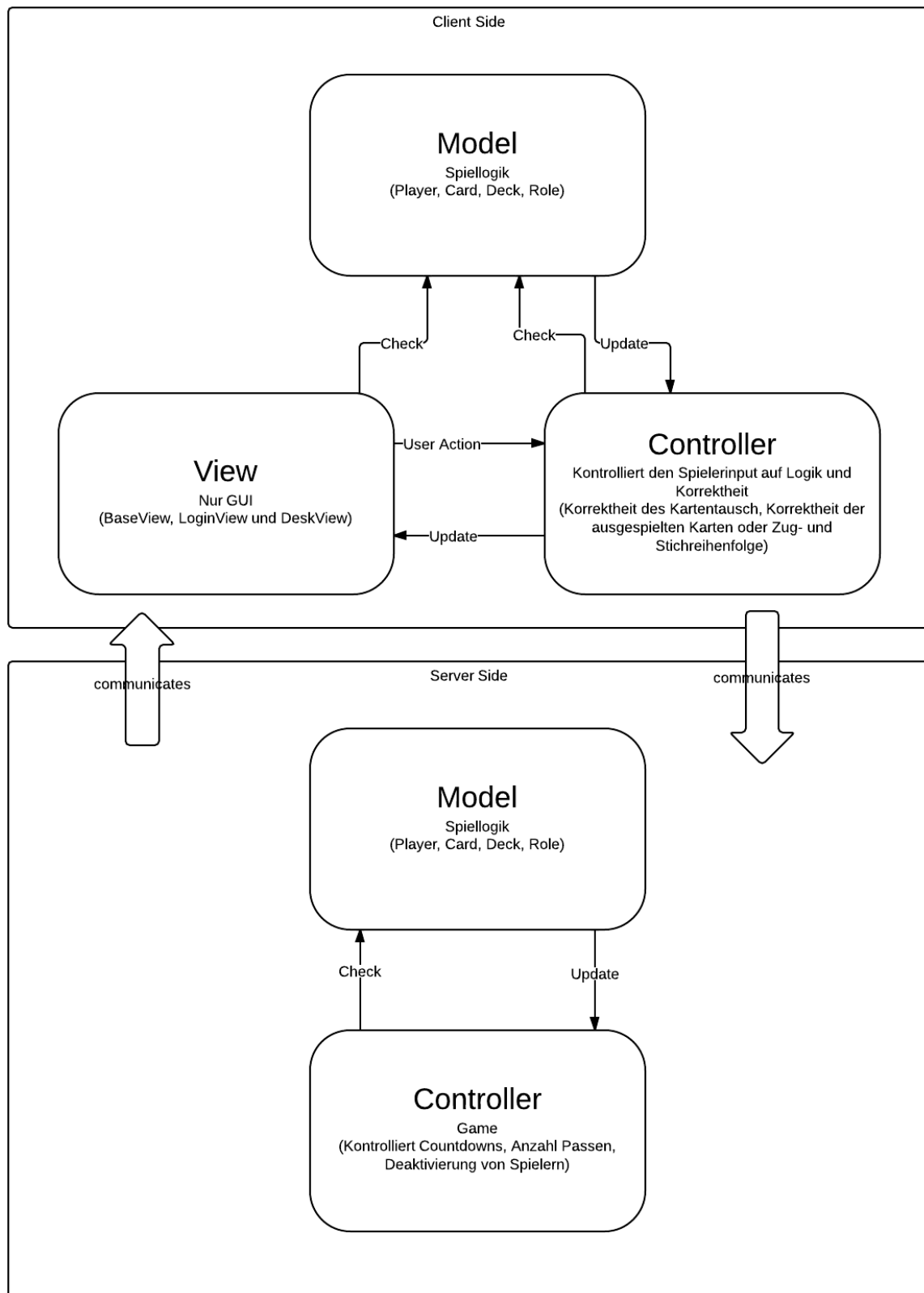
Test Objekt	Spiel beenden
Test Methode	Klicken des Buttons Spiel beenden und Messung der Zeit bis Spiel geschlossen.
Test Ergebnis	Zeit bis Spiel geschlossen \leq 5 Sekunden

Test Objekt	In Java implementiert
Test Methode	1. Installieren des Spiels auf einem Computer ohne JRE. 2. Installieren des Spiels auf einem Computer mit JRE.
Test Ergebnis	1. Installation scheitert 2. Installation erfolgreich

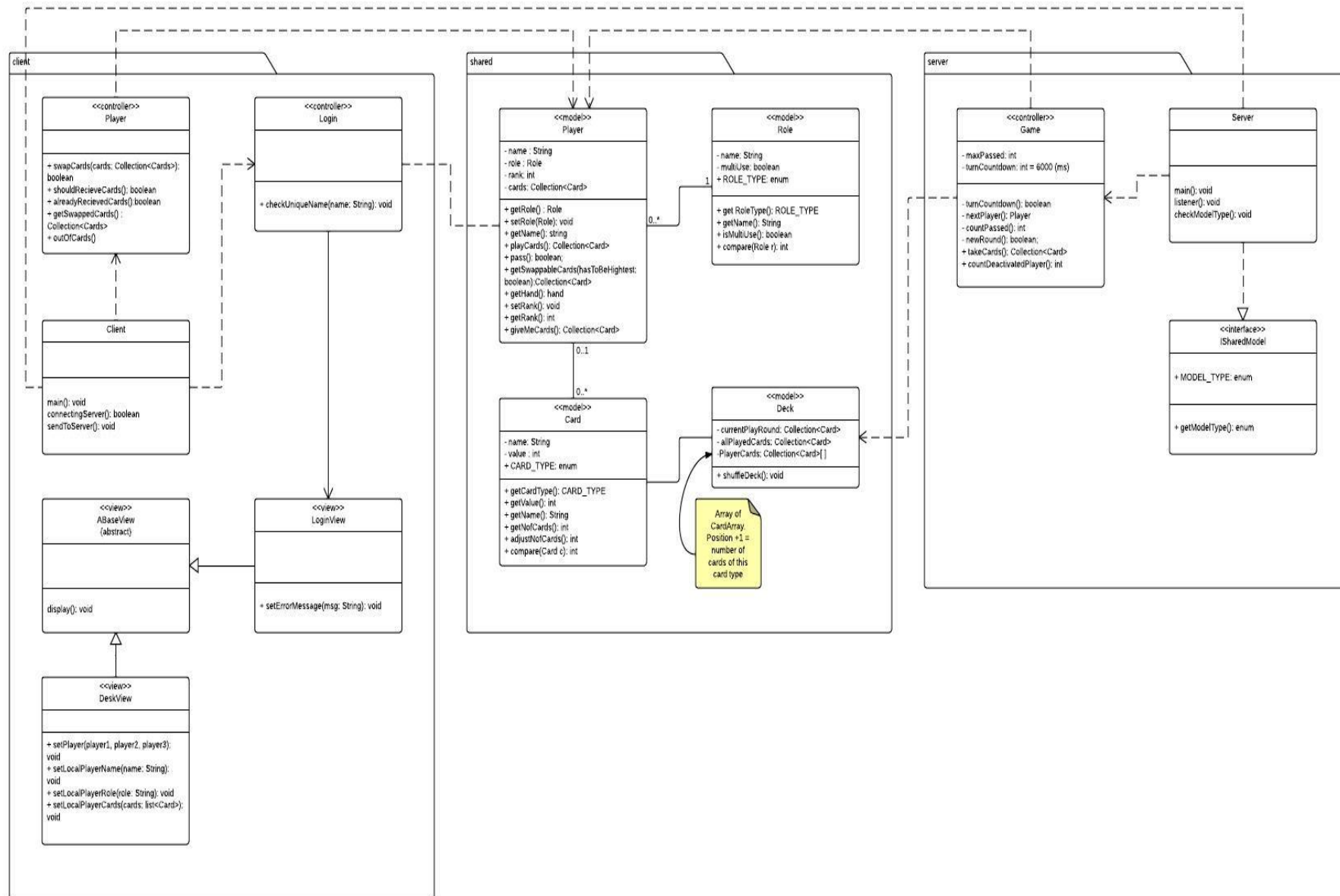
Test Objekt	Benutzername
Test Methode	1. Benutzername mit weniger als 11 Zeichen bestehend aus A-Z, a-z und 0-9 eingeben. 2. Benutzername mit mehr als 10 Zeichen oder nicht erlaubten Zeichen eingeben.
Test Ergebnis	1. Login mit Benutzername erfolgreich falls nicht schon von anderem Spieler verwendet. 2. Login nicht erfolgt und Spieler wird aufgefordert andern Benutzernamen zu benutzen.

Test Objekt	Spielregeln
Test Methode	1. Eine Runde komplett durchspielen mit Kontrolle der Spielregeln.
Test Ergebnis	1. Keine Verletzung der Spielregeln konnte festgestellt werden.

8.Architektur



9. Klassendiagramm



10. Client/Server Schnittstellen und Messages

Client/Server Schnittstelle

Der Client und Server besitzen je eine Main Methode die über das HTTP Protokoll miteinander kommunizieren. Der Server verfügt über eine *Listener*-Methode der auf Anfragen der Client *sendToServer*-Methode wartet.

Messages

Zwischen dem Client und dem Server werden die *model*-Objekte im *shared*-Packet via *Object Streams* hin und her geschickt. Je nach Objekt werden diese entweder auf dem Server oder auf dem Client instanziiert. Nach jeder Änderung eines Objekts wird dieses wieder auf die verschiedenen Client verteilt, bzw. vorgängig zum Server geschickt. Die Klassen im *shared*-Packet sind sowohl auf dem Server wie auch auf dem Client. Dadurch dass sie identisch sind, kann sowohl der Server wie auch der Client auf die Methoden der jeweiligen Klassen zugreifen um zum Beispiel ein Objekt zu aktualisieren.

Anhang 1: Spielablauf Textform

1. Vorbereitung:

Der Spieler lädt den Client auf seinen Rechner. Danach wird der Client installiert und gestartet. Nachdem der Client gestartet wurde, kann sich der Spieler mit Benutzernamen einloggen. Wenn der Spieler eingeloggt ist, muss er warten, bis sich insgesamt vier Spieler eingeloggt haben. Sobald sich vier Spieler eingeloggt haben, startet das Spiel automatisch.

2. Spielstart:

Als bald das Spiel beginnt, erscheint die GUI und alle Karten werden verdeckt ausgelegt. Danach zieht jeder Spieler eine Karte aus den ausgelegten Karten. Diese Karten werden dann für alle Spieler sichtbar aufgedeckt. Je nach Wert der aufgedeckten Karten werden die Rollen bestimmt. Ziehen Spieler gleichwertige Karten, so müssen diejenigen Spieler mit den gleichen Karten erneut ziehen, bis alle unterschiedliche Karten gezogen haben. Es gibt den grossen Dalmuti, den kleinen Dalmuti, den kleinen Diener und den grossen Diener. Der Spieler mit der besten Karte wird der grosse Dalmuti, derjenige mit der zweitbesten Karte wird der kleine Dalmuti, derjenige mit der drittbesten Karte wird der kleine Diener und derjenige Spieler mit der schlechtesten Karte wird der grosse Diener. Dies bestimmt dann ebenfalls die Reihenfolge in welcher das Spiel beginnt. Zuerst startet der grosse Dalmuti, danach folgt der kleine Dalmuti, danach der kleine Diener und als letztes spielt der grosse Diener. Sind die Rollen aufgeteilt und somit die Reihenfolge geklärt werden alle Karten wieder eingezogen und neu gemischt.

3. Schritt:

Es beginnt die erste Runde. Jetzt werden alle Karten ausgeteilt. Dieser Prozess verfährt analog zur oben beschriebenen Reihenfolge. Sind alle Karten verteilt, sieht jeder Spieler seine Karten in sortierter Reihenfolge. Nun folgt der Kartentauschprozess. Der grosse Dalmuti wählt zwei seiner Karten. Diese werden mit den beiden besten Karten des grossen Dieners getauscht. Der kleine Dalmuti wählt eine Karte aus seinem Blatt und tauscht diese mit der besten Karte des kleinen Dieners. Der grosse Dalmuti kann nun als erstes eine beliebige Anzahl gleicher Karten von seiner Hand ausspielen.

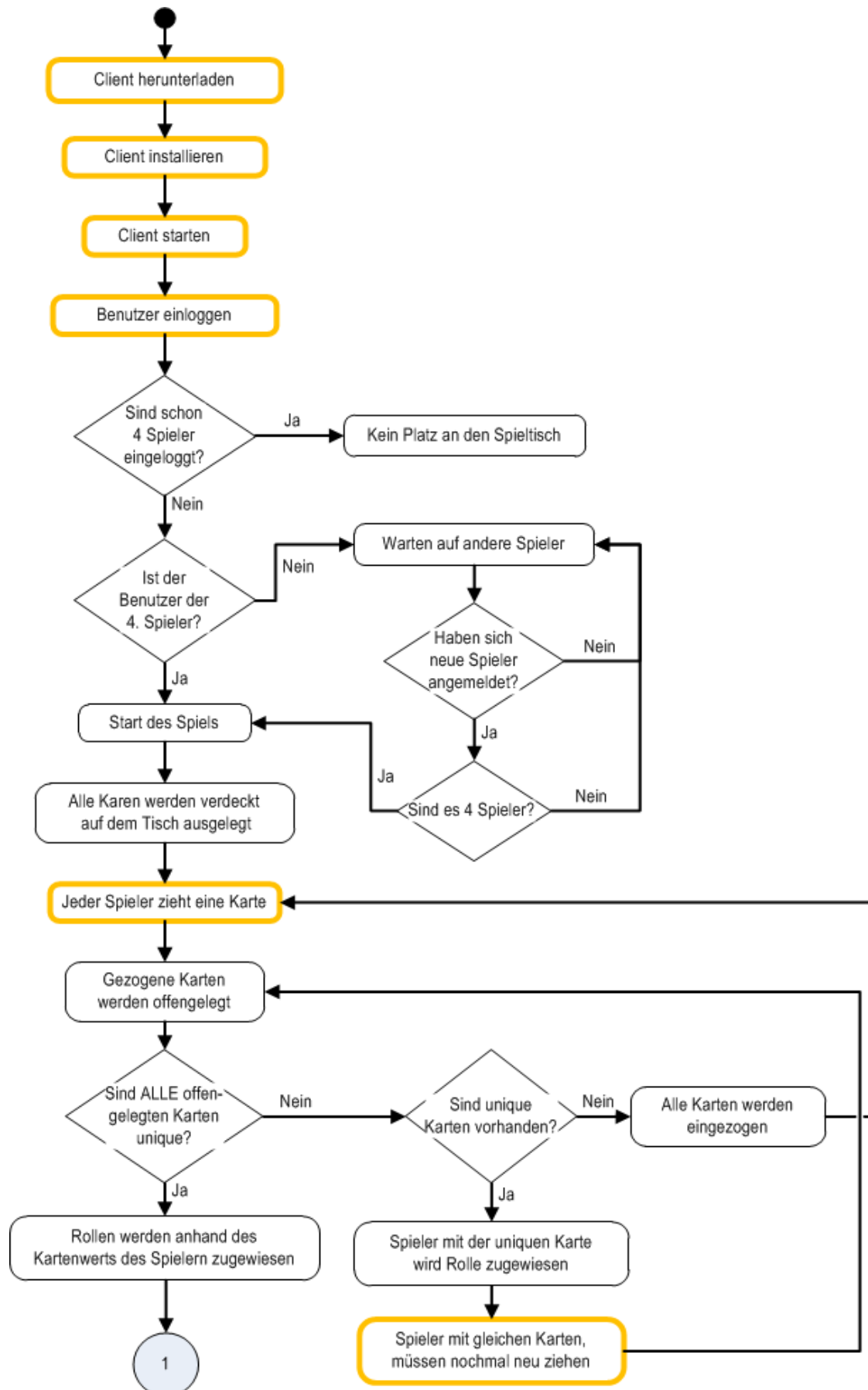
4. Schritt:

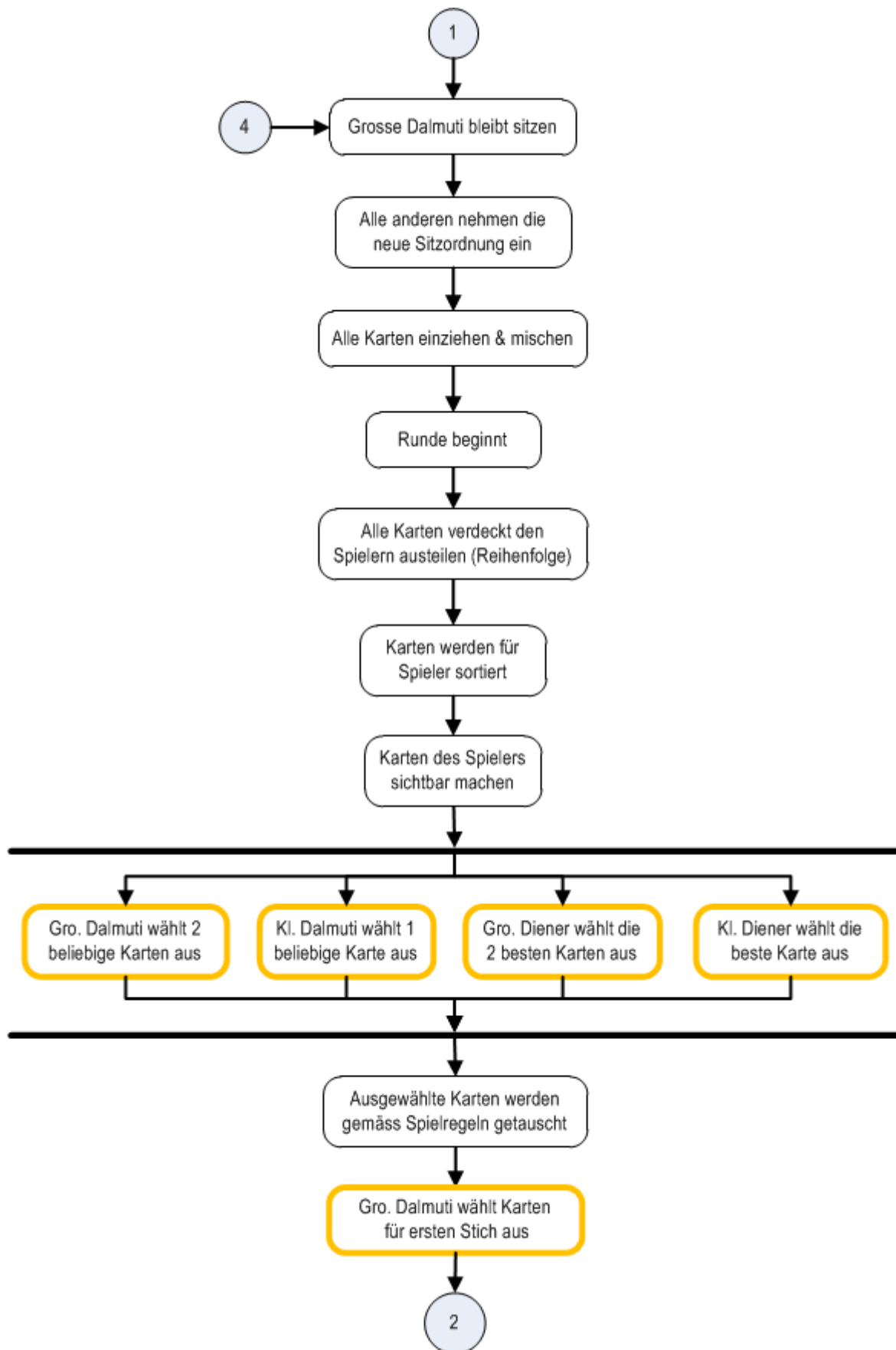
Nachdem er dies getan hat, hat der kleine Dalmuti verschiedene Optionen. Er kann Passen. Dies bedeutet, dass er keine Karten ausspielt und somit der nächste in der Reihenfolge (kleiner Diener) am Zug ist. Als zweite Option kann er versuchen mit den Karten aus seiner Hand den "Stich" zu machen. Dies bedeutet, dass er die gleiche Anzahl Karten wie der vorherige Spieler aus seiner Hand spielen kann. Diese müssen allerdings einen tieferen Wert haben, als die Karten welche der vorherige Spieler gespielt hat. Wenn er dies aufgrund seines Blatts nicht kann, so muss der Spieler passen. Hat er eine dieser Optionen wahrgenommen so folgt nun der Zug des nächsten Spielers gemäss der Reihenfolge.

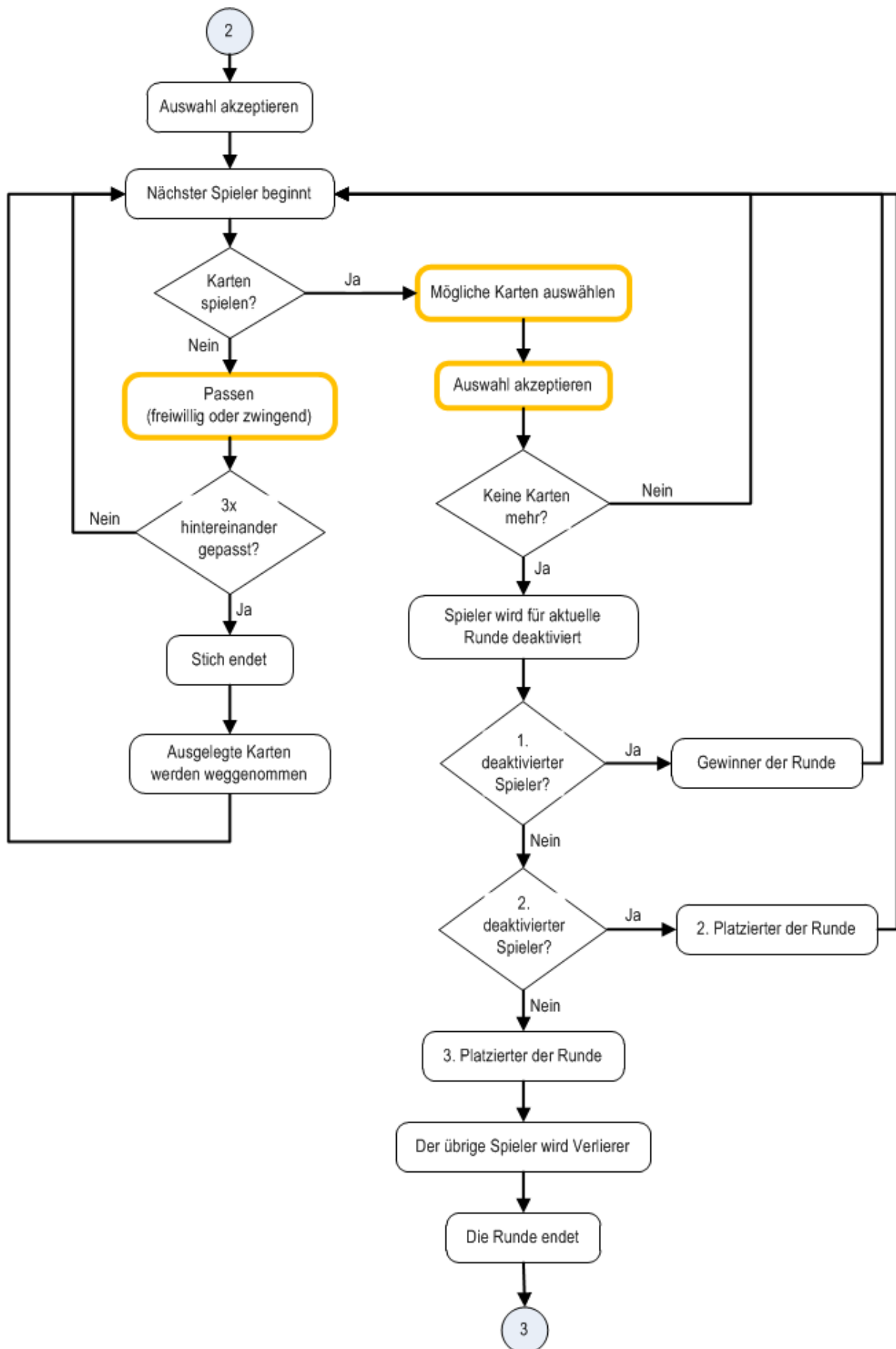
5. Rahmenbedingungen

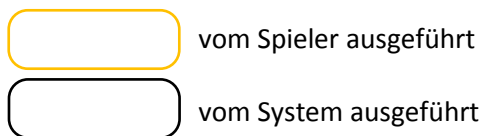
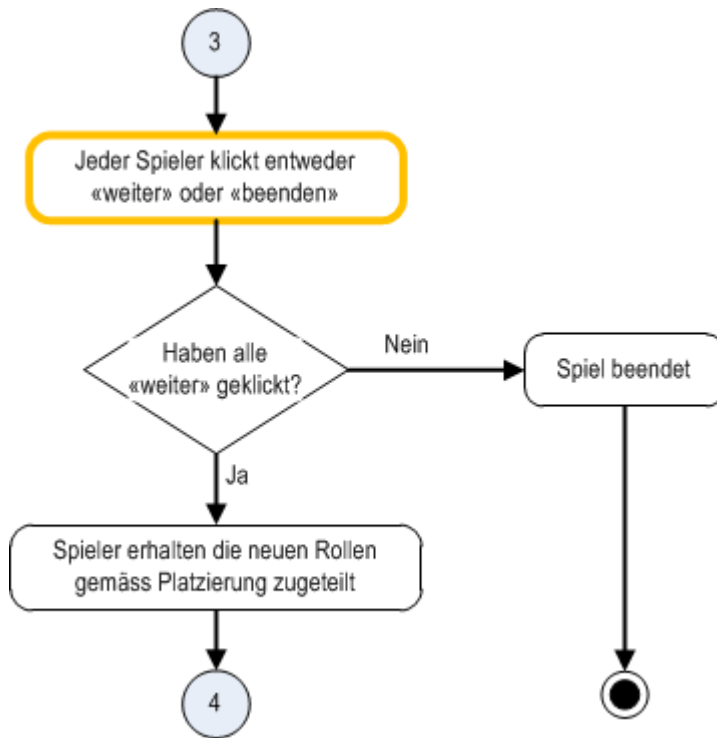
Der 4. Schritt wiederholt sich solange bis alle Spieler gepasst haben. Der Spieler welcher als letztes Karten ausgespielt hat, darf nun den 4. Schritt anstelle des grossen Dalmuti beginnen. Diese Prozedur wiederholt sich solange, bis ein Spieler keine Karten mehr auf der Hand hat. Dieser Spieler hat die Runde gewonnen. Die verbleibenden Spieler spielen nun analog nach dem 4. Schritt weiter, bis nur noch ein Spieler Karten auf seiner Hand hat. Dieser Spieler ist dann der Verlierer. Als bald dies passiert ist die Runde zu Ende und die Rollen werden entsprechend den Rängen neu verteilt. Der Gewinner ist nun der grosse Dalmuti, der Zweitplatzierte ist der kleine Dalmuti, der Drittplatzierte ist der kleine Diener und der Verlierer ist der grosse Diener. Danach startet die zweite Runde und der 3. Schritt beginnt entsprechend der Rollen erneut. Wenn dies geschehen ist, spielt der grosse Dalmuti wieder als erstes Karten aus und der 4. Schritt tritt wieder in Kraft. Dieser wiederholt sich erneut solange bis die Ränge erneut feststehen und die dritte Runde erreicht ist. Dieses gesamte Prozedere wiederholt sich ständig, bis einer oder mehrere Spieler das Spiel verlassen.

Anhang 2: Spielablauf Flussdiagramm









Anhang 3: Glossar

Begriff	Beschreibung
Spiel	1. Das Spiel "Der grosse Dalmuti", bzw. das Programm um dieses Spiel zu spielen. 2. Die Zeitspanne nach dem Beitritt in das Spiel bis zum Beenden des Spiels. Kann aus mehreren Runden bestehen.
Runde	Beginnt mit der Neuverteilung der Rollen und endet mit der Zuweisung des letzten Ranges sobald der zweitletzte spieler alle seine Karten ausgespielt hat. Kann aus mehreren Stichen bestehen
Stich	Beginnt mit dem Ausspielen der ersten Karte(n) nach dem Einsammeln der zuletzt gespielten Karten und endet mit dem dreimaligen hintereinander Passen (freiwillig oder zwingend).
Spielzug	Ist das Karten spielen oder passen der einzelnen Spielern während eines Stichts.
Rolle	Rollen die die Spieler gemäss Spielregeln einnehmen können. Es gibt folgende Rollen in absteigender Reihenfolge: grosser Dalmuti, kleiner Dalmuti, kleiner Diener und grosser Diener
Reihenfolge	Reihenfolge in welcher die Spieler zum Zug kommen. Diese ist gleich wie die Reihenfolge der Rollen. Das heisst der Grosse Dalmuti beginnt beim ersten Stich und danach geht es der Rollenreihenfolge nach weiter. Falls ein anderer als der Grosse Dalmuti den Stich gewinnt, beginnt dieser mit dem nächsten Stich und es geht von dieser Position der Rollenreihenfolge nach weiter.
Rang	Wird zugewiesen sobald der erste Spieler in einer Runde die letzten Karten spielt. Der erste Spieler der seine letzten Karten Spielt nimmt den ersten Rang ein usw. Das System speichert diese Ränge um Sie in der nachfolgenden Runde für die Rollenverteilung zu verwenden. Der erste Rang wird in der nächsten Runden der grosse Dalmuti usw.
Client	Ausführbares Programm für die Spieler welches das Spiel darstellt was durch ein GUI dargestellt wird. Der Client kommuniziert mit dem Server um die notwendigen Aktionen innerhalb des Spiels durchzuführen.
Server	Der Server berechnet, regelt und kontrolliert die ganzen Aktionen die durch die Clients gemacht werden.
GUI	Akronym für „Graphical User Interface“ – ein auf Fenstern und Dialogen basiertes, typischerweise Maus- und Keyboard-orientiertes User Interface.