

Softwarepraktikum 2004

Kino

Projektdokumentation (Teilbeleg 1)

Teamleiter: Thomas Emmerich

Mitglieder des Projektteams:

Thomas Seidel
Marko Prokot
Markus Schubert
Marcus Zelend
Robert Meier

Praktikumsbetreuer:

H. Müller-Dieckert

Inhaltsverzeichnis

1. SPEZIFIKATION DER FUNKTIONELLEN ANFORDERUNGEN.....	3
1.1. Produktbeschreibung.....	3
1.1.1. Leistungsumfang.....	3
1.1.2. Nutzungsumgebung.....	3
1.1.3. Nutzerklassen.....	3
1.2. Funktionelle Anforderungen.....	3
1.2.1. Umgebungsmodell.....	3
1.2.2. Verhaltensmodell.....	5
1.3. Definition der Nutzerschnittstelle.....	16
1.3.1. Webbasiertes System zur Kartenreservierung.....	16
1.3.2. Kassen- und Adminsystem.....	16
2. SPEZIFIKATION DER OPERATIONELLEN ANFORDERUNGEN.....	20
2.1. Operationelle Anforderungen an die Daten, Datenbasen und Datenströme.....	20
2.1.1. Datenkatalog mit Speichergrößen.....	20
2.1.2. Abschätzungen.....	22
2.1.3. Genauigkeit.....	23
2.2. Operationelle Anforderungen an die Prozesse.....	23
3. SPEZIFIKATION WICHTIGER QUALITÄTSANFORDERUNGEN.....	23
3.1. Benutzbarkeit.....	23
3.2. Zuverlässigkeit	23
3.3. Integrität.....	23
3.4. Flexibilität.....	24
3.5. Portabilität.....	24
4. BASISMASCHINE UND ENTWICKLUNGSUMGEBUNG.....	24
4.1. Basismaschine.....	24
4.1.1. Hardware.....	24
4.1.2. Betriebssystem.....	24
4.1.3. Sonstige Basissoftware.....	24
4.2. Entwicklungsumgebung.....	24
5. PLANUNG.....	25

1. Spezifikation der funktionellen Anforderungen

1.1. Produktbeschreibung

1.1.1. Leistungsumfang

Das System besteht aus zwei Teilsystemen. Eines soll dem potentiellen Kinogast per Internet ermöglichen, Belegungen der verschiedenen Vorstellungen anzuzeigen und Karten vorzubestellen.

Das zweite System soll im Kino an den Kassen eingesetzt werden und zum Verkauf von Karten genutzt werden. Es soll weiterhin Wechselgeld berechnen können und aktuelle Belegungen der Vorstellungen anzeigen.

1.1.2. Nutzungsumgebung

Das System für den Kinogast soll an jedem an das Internet angeschlossenen Computer verfügbar sein.

Das Kassensystem ist nur auf den Kassensystemen vorhanden und verfügbar.

Weitere Administrationsaufgaben können von den Kassensystemen oder von einem zusätzlichen Rechner im Kino aus ausgeführt werden.

1.1.3. Nutzerklassen

Es sind folgende drei Nutzerklassen geplant

G – Kinogast

K – Kassierer/in

A – Administrator/in (Programmmchef/in vom Kino)

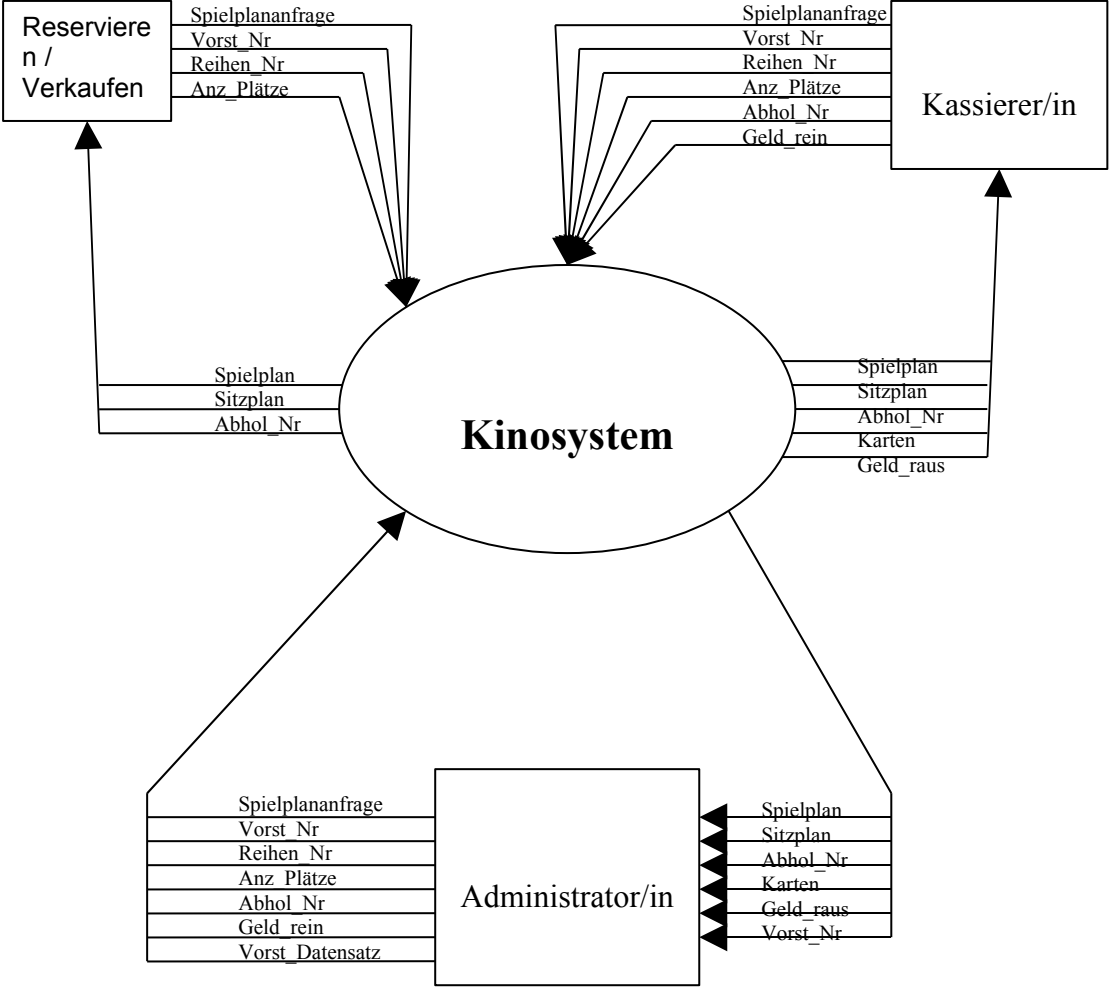
1.2. Funktionelle Anforderungen

1.2.1. Umgebungsmodell

1.2.1.1. Ereignistabelle

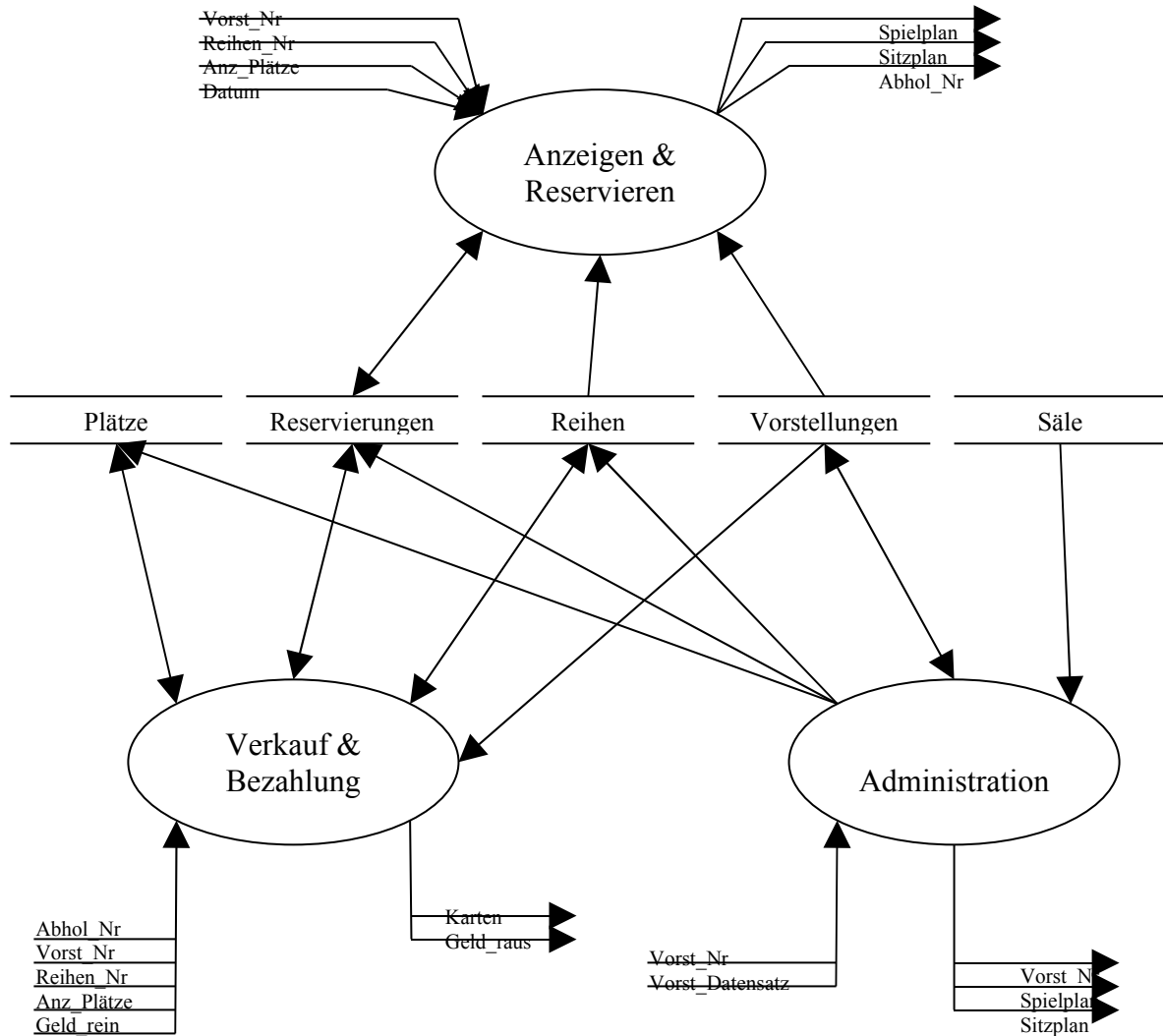
Nr.	Event	Auslöser	Data in	Data out
01	Spielplan anzeigen	G,K,A	Spielplananfrage	Spielplan
02	Sitzplan anzeigen	G,K,A	Vorst_Nr	Sitzplan
03	Plätze reservieren	G,K,A	Vorst_Nr + Reihen_Nr + Anz_Plätze	Abhol_Nr
04	Karten verkaufen	K,A	Vorst_Nr + Reihen_Nr + Anz_Plätze + Geld_rein	Karten + Geld_raus
05	Reservierte Karten kaufen	K,A	Abhol_Nr + Geld_rein	Karten + Geld_raus
06	Alle Reservierungen löschen	A	Vorst_Nr	Sitzplan
07	Vorstellung hinzufügen	A	Vorst_Datensatz	Vorst_Nr + Spielplan
08	Vorstellung bearbeiten	A	Vorst_Nr + Vorst_Datensatz	Spielplan
09	Vorstellung löschen	A	Vorst_Nr	Spielplan

1.2.1.2. Kontextdiagramm



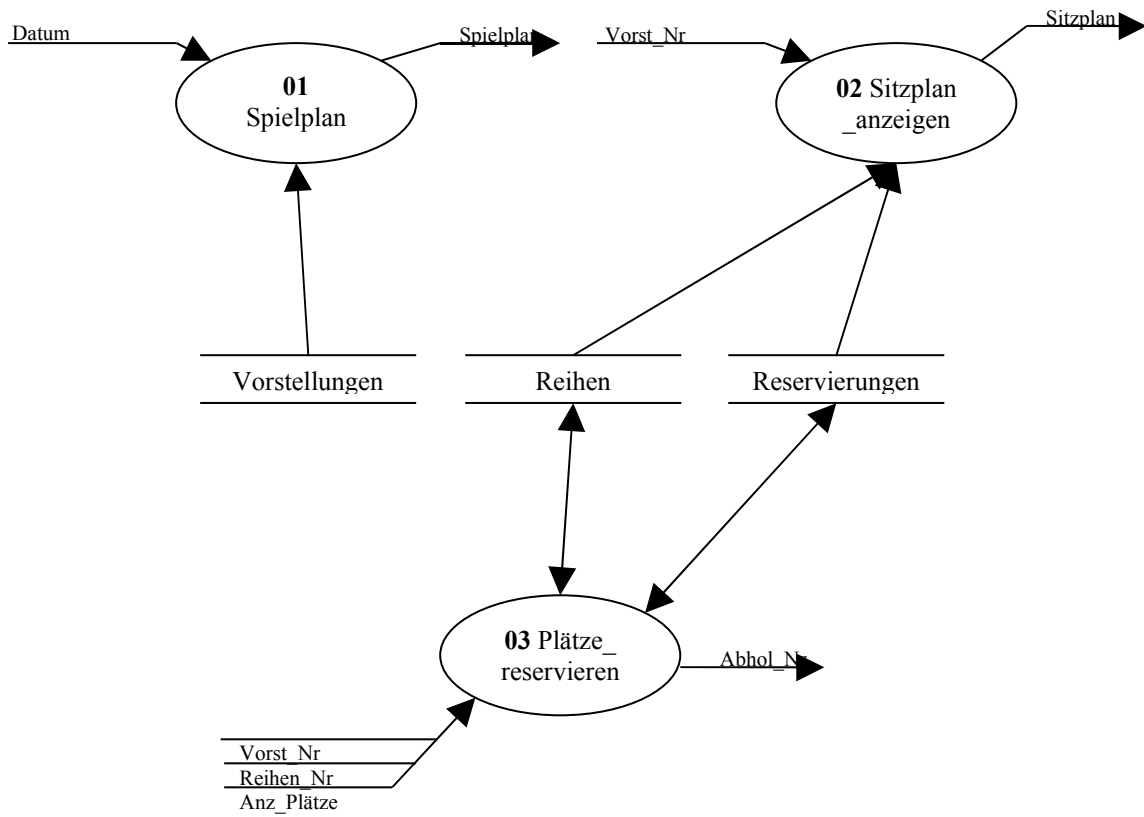
1.2.2. Verhaltensmodell

1.2.2.1. Grobes Verhaltensmodell

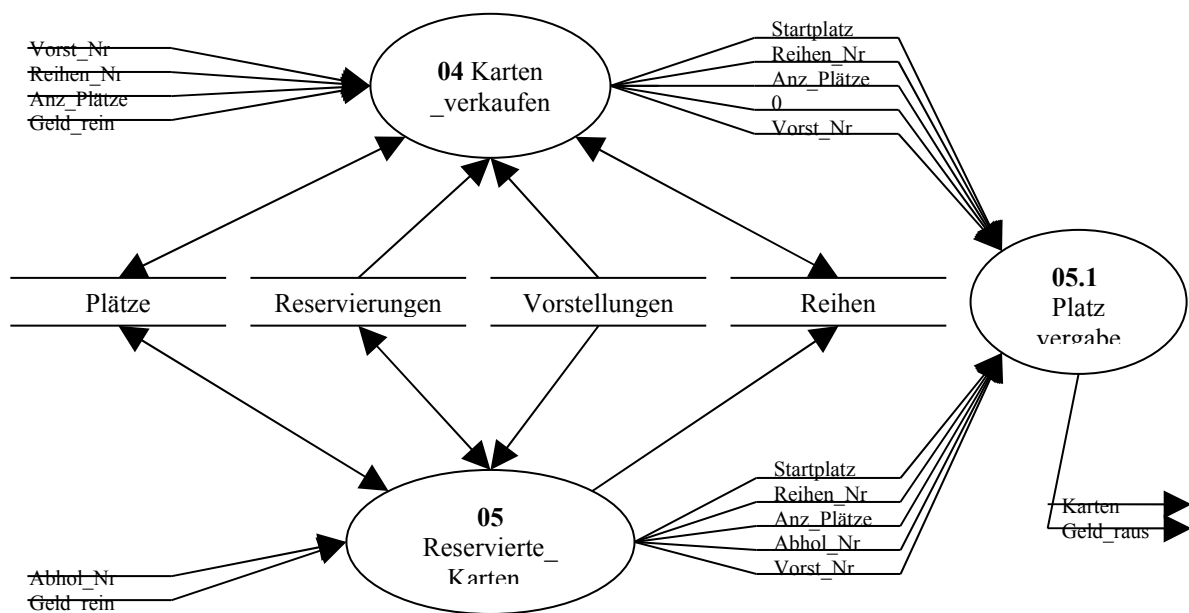


1.2.2.2. Primäres Verhaltensmodell

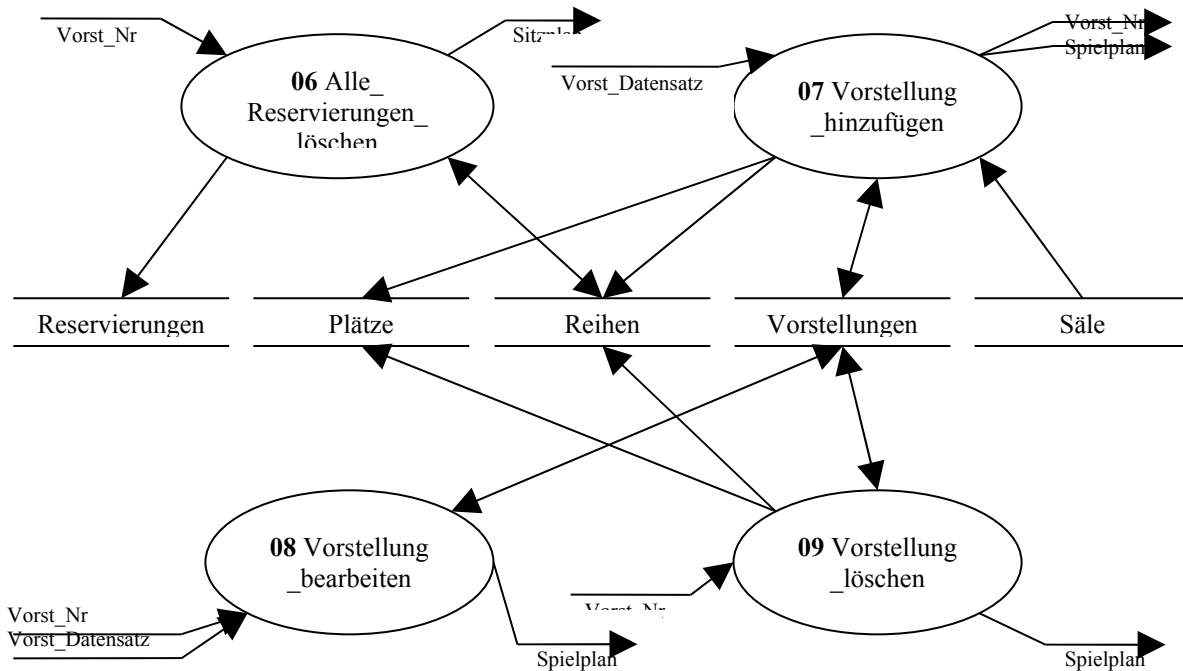
1.2.2.2.1. Teilmodell Anzeigen & Reservieren



1.2.2.2.2. Teilmodell Verkauf & Bezahlung



1.2.2.2.3. Teilmodell Administration

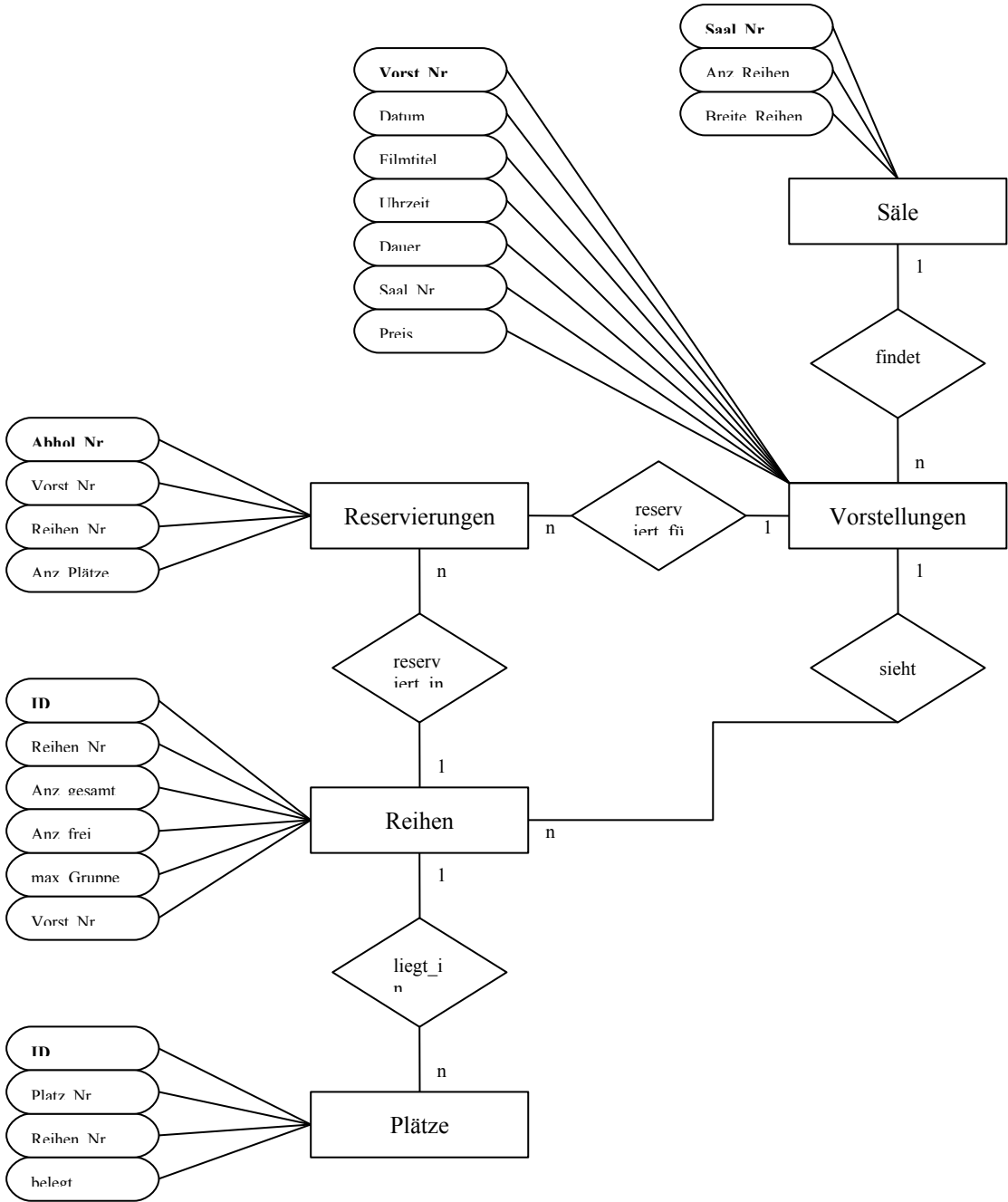


1.2.2.4. Datenkatalog

<i>Element</i>	<i>Strukturbeschreibung</i>
Vorstellungen	{Vorstellung}
Vorstellung	@Vorst Nr + Datum + Filmtitel + Uhrzeit + Dauer + Saal Nr + Preis
Vorst Nr	gZahl *3-stellig, positiv*
Datum	Tag + Monat + Jahr
Filmtitel	Zeichenkette50
Uhrzeit	Stunden + Minuten
Dauer	gZahl *3-stellig, 0 <= Dauer <=300*
Saal Nr	gZahl *2-stellig, 1 <= Saal Nr <= 20*
Preis	Geldbetrag
Tag	gZahl *2-stellig, 1 <= Tag <= 31*
Monat	gZahl *2-stellig, 1 <= Monat <= 12*
Jahr	gZahl *4-stellig, 2004 <= Jahr <= 2100*
Stunden	gZahl *2-stellig, 00 <= Stunden <= 23*
Minuten	gZahl *2-stellig, 00 <= Minuten <= 59*
Geldbetrag	rZahl *6-stellig, davon 2 Nachkommastellen*
Säle	{Saal}
Saal	@Saal Nr + Anz Reihen + Breite Reihen
Anz Reihen	gZahl *2-stellig*
Breite Reihen	gZahl *2-stellig ... gemeint ist hier Anzahl der Plätze je Reihe*
Reservierungen	{Reservierung}
Reservierung	@Abhol Nr + Vorst Nr + Reihen Nr + Anz Plätze
Abhol Nr	gZahl *6-stellig*

Reihen Nr	gZahl *2-stellig*
Anz_Plätze	gZahl *2-stellig*
Reihen	{Reihe}
Reihe	@ID + Vorst Nr + Reihen Nr + Anz_gesamt + Anz_frei + max_Gruppe
ID	gZahl
Anz_gesamt	gZahl * >0 *
Anz_frei	gZahl * >=0 *
max_Gruppe	gZahl * > 0 *
Plätze	{Platz}
Platz	@ID + Platz Nr + Reihen Nr + belegt
Platz Nr	gZahl *2-stellig*
belegt	[0 1]
findet statt in	@Vorst Nr + Saal Nr
reserviert für	@Abhol Nr + Vorst Nr
reserviert_in	@Abhol_Nr + <Reihe>ID
sieht	@<Reihe>ID + Vorst Nr
liegt in	@<Platz>ID + <Reihe>ID
Spielplan	{Vorstellung}
Sitzplan	{Reihe + {belegte Platz Nr}}
belegte Platz Nr	Platz Nr
Geld_rein	Geldbetrag
Geld_raus	Geldbetrag
Karten	{Vorst Datensatz}
Vorst Datensatz	Datum + Filmtitel + Uhrzeit + Dauer + Saal Nr + Preis

1.2.2.5. ER-Diagramm



1.2.2.6. Prozessspezifikationen

1.2.2.6.1. Prozess *Spielplan_anzeigen*

Prozess: Spielplan_anzeigen Datum: 05.05.2004
Bearbeiter: M. Prokot

Voraussetzung

Übergabe: Datum

```
begin
Finde in Vorstellungen Vorstellung mit Vorstellung.Datum größer gleich Übergabe.Datum
if nicht gefunden
then
    schreibe Fehler
else
    Ausgabe Array mit Vorstellungen
endif
end
```

1.2.2.6.2. Prozess *Sitzplan_anzeigen*

Prozess: Sitzplan_anzeigen Datum: 05.05.2004
Bearbeiter: R. Meier

Voraussetzung

Übergabe: Vorst_Nr, [Abhol_Nr]

```
begin
Finde in Vorstellungen Vorstellung mit Vorstellung.Vorst_Nr gleich Übergabe.Vorst_Nr
Finde in Säle Saal mit Saal_Nr gleich Vorstellung.Saal_Nr

if Übergabe.Abhol_Nr
then
    Finde in Reservierungen Reservierung.Reihen_Nr wo Reservierung.Abhol_Nr gleich
    Übergabe.Abhol_Nr
endif

if nicht gefunden
then
    schreibe Fehler
else
    for i=1 bis Saal.Anz_Reihen
        Finde in Reihen Reihe mit Reihe.Vorst_Nr gleich Übergabe.Vorst_Nr
        Setze Z_Verkauft = 0 //Zählvariable für verkaufte Plätze
        for j=1 bis Saal.Breite_Reihen
            Finde in Plätze Platz mit Platz.Reihen_Nr gleich i und Platz.Platz_Nr gleich j
            if Platz.belegt
            then
                Setze Sitze[i,j] = 2 //Sitze: Array der Größe
                Breite_Reihen*Anz_Reihen //0 - frei, 1 - reserviert, 2 - verkauft
                Setze Z_Verkauft = Z_Verkauft +1
            endif
        endfor
        Setze j=1
        Setze k=Saal.Breite_Reihen - Reihe.Frei - Z_Verkauft
```

```

while j kleiner gleich k
    if Sitze[i,j] ungleich 2
    then
        Setze Sitze[i,j] = 1
    else
        Setze k=k+1
    endif
endwhile

if Übergabe.Abhol_Nr
then
    if Reservierung.Reihen_Nr gleich i
    then
        Ausgabe Reihenmarkierung
    endif
endif

endfor
Ausgabe Sitze //Grafische Ausgabe des Arrays
endif
end

```

1.2.2.6.3. Prozess Plätze_reservieren

Prozess: Plätze_reservieren

Datum: 05.05.2004

Bearbeiter: R. Meier

Voraussetzung:

Übergabe: Reihen_Nr, Anz_Plätze

```

begin
Finde in Reihen Reihe mit Reihe.Vorst_Nr == Übergabe.Vorst_Nr und Reihe.Reihen_Nr ==
Übergabe.Reihen_Nr

if gefunden
then
    if Reihe.frei größer gleich Übergabe.Anz_Plätze
    then
        Gib zurück SucheFreieAbholNr
        Schreibe Reservierung_Nr
        Setze tmp = Reihe.Frei – Übergabe.Anz_Plätze
        Schreibe in Reihe.Frei = tmp
    elseif
        Schreibe Fehler, Reihe voll
        Aktiviere Sitzplan
    endif
endif
endif
end

```

1.2.2.6.4. Prozess Karten_verkaufen

Prozess: Karten_verkaufen

Datum: 05.05.2004

Bearbeiter: M. Schubert

Voraussetzung

Übergabe: Vorst_Nr, Reihen_Nr, Anz_Plätze, Start_platz

begin

Starte Prozess Sitzplan_anzeigen(Vorst_Nr)

Ausgabe Eingabeformular //Startplatznummer, Reihennummer, Anzahl der Plätze

Starte Prozess Platz_vergabe(Start_platz, Reihen_nr, Anz_Plaetze, 0, Vorst_Nr)

end

1.2.2.6.5. Prozess Reservierte_Karten_verkaufen

Prozess: Reservierte_Karten_verkaufen

Datum: 05.05.2004

Bearbeiter: M. Prokot

Voraussetzung

Übergabe: Abhol_Nr

begin

Finde in Reservierungen Reservierung mit Reservierung.Abhol_Nr gleich Übergabe.Abhol_Nr

Finde in Reihen Reihe mit Reihe.Vorst_Nr gleich Reservierung.Vorst_Nr

Finde in Vorstellungen Vorstellung mit Vorstellung.Vorst_Nr gleich Reservierung.Vorst_Nr

Finde in Säle Saal mit Saal.Saal_Nr gleich Vorstellung.Saal_Nr

Finde in Reservierungen Reservierung mit Reservierung.Reihen_Nr gleich Reservierung[Reihen_Nr]
und Reservierung.Vorst_Nr gleich Reservierung[Vorst_Nr]

Starte Prozess Sitzplan_anzeigen(Reservierung.Vorst_Nr, Übergabe.Abhol_Nr)

Ausgabe Reihen-Reservierungen

Ausgabe Eingabeformular //Startplatzeingabe, Reihennummer

Starte Prozess Platz_vergabe(Start_platz, Reihen_Nr, 0, Abhol_Nr, Reservierung.Vorst_Nr)

Lösche in Reservierungen Reservierung wo Reservierung.Abhol_nr gleich Übergabe.Abhol_Nr

end

1.2.2.6.5.1. Prozess Platzvergabe

Prozess: Platzvergabe

Datum: 05.05.2004
Bearbeiter: M. Prokot,
M. Schubert

Vorraussetzung:

Übergabe: Start_platz, Reihen_Nr, Anz_Plaetze, Abhol_Nr, Vorst_Nr

begin

if Übergabe.Anz_Plaetze ungleich 0 und Übergabe.Abhol_Nr
then

Finde in Reservierungen Reservierung mit Reservierung.Abhol_Nr gleich Übergabe.Abhol_Nr
Finde in Reihen Reihe mit Reihe.Vorst_Nr gleich Reservierung.Vorst_Nr

Finde in Plaetze Platz mit Platz.Reihen_ID gleich Reihen.ID

for i = Übergabe.Start_platz bis Übergabe.Start_platz + Reservierung.Anz_Plaetze
Setze in Plaetze Platz.belegt gleich 2 wo Platz.Reihen_ID gleich Reihen.ID und
Platz.Platz_Nr gleich i
Ausdruck Karte i

else

Finde in Reihen Reihe mit Reihe.Vorst_Nr gleich Übergabe.Vorst_Nr
Finde in Plaetze Platz mit Platz.Reihen_ID gleich Reihen.ID

for i = Übergabe.Start_platz bis Übergabe.Start_platz + Übergabe.Anz_Plaetze
Setze in Plaetze Platz.belegt gleich 2 wo Platz.Reihen_ID gleich Reihen.ID und
Platz.Platz_Nr gleich i
Ausdruck Karte i

end

1.2.2.6.6. Prozess Alle Reservierungen löschen

Prozess: Alle_Reservierungen_löschen

Datum: 05.05.2004
Bearbeiter: M. Schubert

Vorraussetzung:

Übergabe: Vorst_Nr

begin

Finde in Reservierungen Reservierung mit Vorst_Nr gleich Übergabe.Vorst_Nr

If gefunden
then

Setze in Reihen Reihe.Anz_Frei gleich Reihe Anz_Frei + Reservierung.Anz_Plätze wo
Vorst_Nr gleich Übergabe.Vorst_Nr
Lösche in Reservierungen Reservierung mit Vorst_Nr gleich Übergabe.Vorst_Nr
Rückgabe = Ergebnis von Starte Prozess Sitzplan_anzeigen

else

Ausgabe Fehler
endif
end

1.2.2.6.7. Prozess Vorstellung_hinzufügen

Prozess: Vorstellung_hinzufügen Datum: 05.05.2004
Bearbeiter: M. Zelend

Vorraussetzung:

Übergabe: Vorst_Datensatz

begin
Füge ein in Vorstellungen Vorstellung mit Übergabe.Vorst_Datensatz und neuer Vorst_Nr
Füge ein in Reihen: Vorstellung.Saal.Anz_Reihen Reihen mit
neuer ID
Vorst_Nr = neuer_Vorst_Datensatz.Vorst_Nr
Anz_gesamt = Vorstellung.Saal.Breite_Reihen
Anz_frei = Anz_gesamt
max_Gruppe = Anz_gesamt

Für alle eingefügten Reihen:
Füge ein in Plätze: Reihe.Anz_gesamt Plätze mit
Status = 0
Platz_Nr = fortlaufende Nr
Reihen_Nr = Reihe.ID

end

1.2.2.6.8. Prozess Vorstellung_bearbeiten

Prozess: Vorstellung_bearbeiten Datum: 05.05.2004
Bearbeiter: M. Zelend

Vorraussetzung:

Übergabe: Vorst_Nr

begin
Finde in Vorstellungen Vorstellung mit Vorstellung.Vorst_Nr gleich Übergabe.Vorst_Nr

if gefunden
then
Ausgabe Formular Vorstellungsdatensatz *Editieren kann
vorgenommen werden*
Setze in Vorstellungen Vorstellung gleich Vorstellungsdatensatz
else
Ausgabe 'Vorstellung nicht gefunden'
endif
end

1.2.2.6.9. Prozess Vorstellung löschen

Prozess: Vorstellung_löschen

Datum: 05.05.2004

Bearbeiter: M. Zelend

Vorraussetzung:

Übergabe: *Vorst_Nr*

```
begin
    Finde in Vorstellungen Vorstellung mit Vorstellung.Vorst_Nr gleich Übergabe.Vorst_Nr
    if gefunden
        then
            Loesche      in Vorstellungen Vorstellung mit Vorstellung.Vorst_Nr gleich
Übergabe.Vorst_Nr
            Für alle Reihen mit Reihe.Vorst_Nr = Vorst_Nr:
                Lösche in Plätze alle Plätze mit Platz.Reihen_Nr = Reihe.ID
                Lösche Reihe
        else
            Ausgabe 'Vorstellung nicht gefunden'
    endif
end
```

1.3. Definition der Nutzerschnittstelle

1.3.1. Webbasiertes System zur Kartenreservierung

1.3.1.1. E/A

Zum Navigieren im System wird eine Maus vorausgesetzt. Für die Eingabe der Reservierungsdaten ist eine Tastatur nötig. Das System läuft in einem Browserfenster. Für die Darstellung wird ein Farbmonitor vorausgesetzt.

1.3.1.2. Layout

Das System umfasst lediglich die Spielplan- und Sitzplananzeige, sowie die Reservierungsfunktion. Das Layout dieser Funktionen soll das gleiche wie im Kassensystem sein (s.u.)

1.3.2. Kassen- und Adminsystem

1.3.2.1. E/A

Das Programm, welches auf den Kassenrechnern des Kinos laufen soll, setzt zur korrekten Darstellung einen Farbmonitor voraus. Zum Steuern und zur Dateneingabe sind hier Maus und Tastatur nötig. Zusätzlich erfolgt die Ausgabe der Platzkarten über einen Drucker.

1.3.2.2. Layout

Legende: Bedienknopf mit Beschriftung

 Ausgabewert des Datenkatalogs

 Eingabewert des Datenkataloges

Login	
Name : <input type="text"/>	Passwort: <input type="text"/>

Admin - Bereich	
Datum <input type="text"/> Zeit <input type="text"/> Nummer <input type="text"/> Titel <input type="text"/> Saal <input type="text"/> Preis <input type="text"/> <input type="text"/> <input type="text"/>	
Vorstellung[0].datum <input type="text"/> . uhrzeit <input type="text"/> . Vorst_nr <input type="text"/> . filmtitel <input type="text"/> . saal_	
Nr <input type="text"/> . geldbetrag <input type="text"/> <input type="text"/> .	
:	
Vorstellung[k].datum <input type="text"/> ... <input type="text"/> ... <input type="text"/> ... <input type="text"/> ... <input type="text"/>	
<input type="button" value="→ neue Vorstellung erstellen"/>	

Admin - Bereich → Vorstellung bearbeiten	
Vorstellung[m] <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	
Datum <input type="text"/> Zeit <input type="text"/> Nummer <input type="text"/> Titel <input type="text"/> Saal <input type="text"/> Preis <input type="text"/> <input type="text"/>	
<input type="button" value="→ Zurück"/>	
<input type="button" value="Reserviere"/>	
<input type="button" value="→ neue Vorstellung erstellen"/>	

Reservieren / Verkaufen

Datum	Zeit	Nummer	Titel	Saal	Preis
Vorstellung[m].datum	.uhrzeit	.Vorst_nr	.filmtitel	.saal_Nr	.geldbetrag

Plätze je Reihe

Plätze je Reihe

Anzahl frei:

Anz_Frei



Anzahl Plätze

Platz_Anz

→ Reservieren

Anzahl Plätze

Platz_Anz

Startplatz

Platz_Nr

→ Verkaufen

→ Zurück

Reserviere

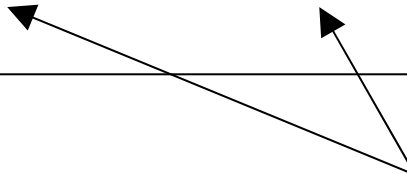
Kartenverkauf

Preis: □Preis□□□□□□□□

□□□□□□Bezahlter Betrag: □

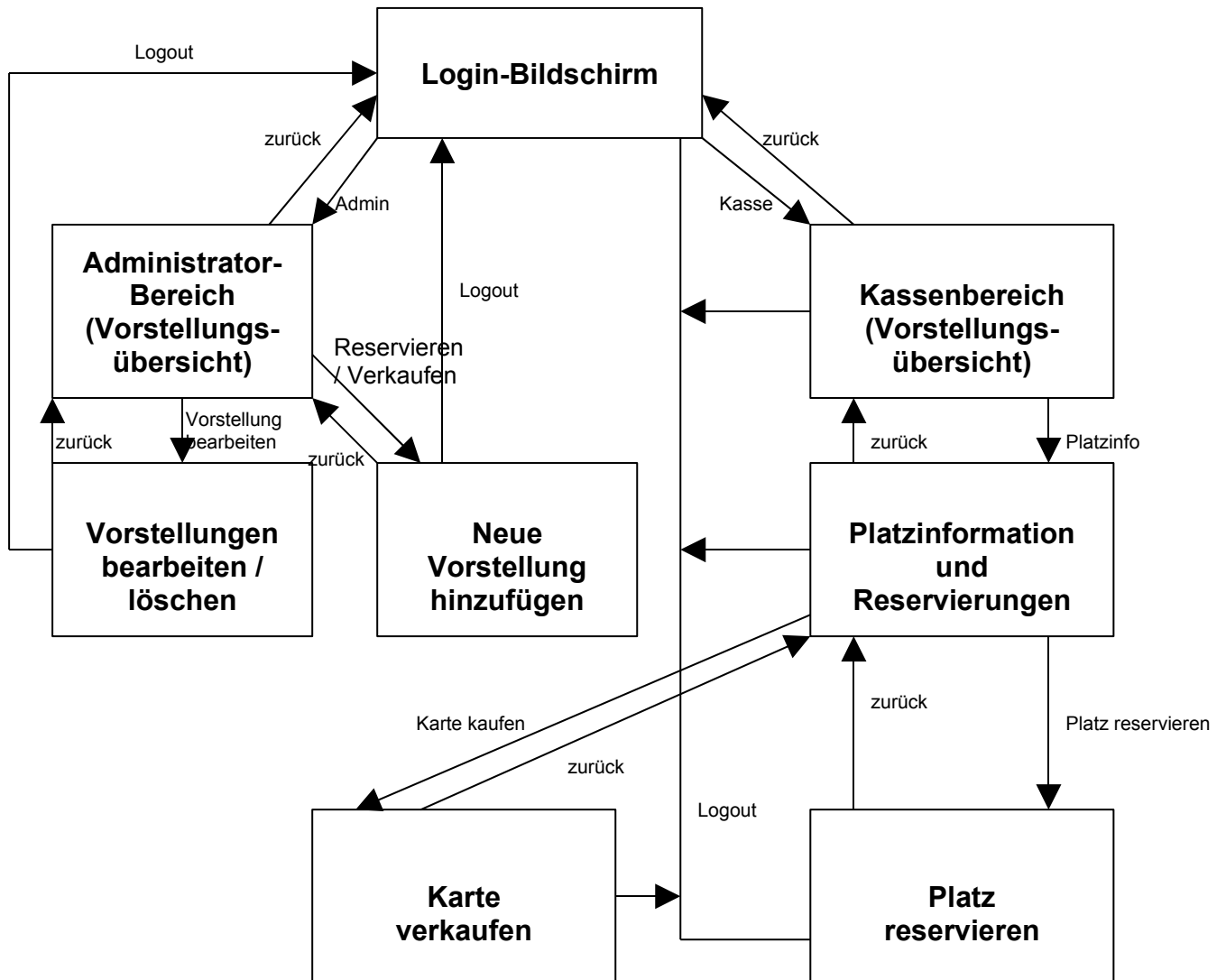
□□Rückgeld:□

Rückgeld□□□□□□□□□□□□□□



Berechnetes Rückgeld

1.3.2.3. Zustandsdiagramm



2. Spezifikation der operationellen Anforderungen

2.1. Operationelle Anforderungen an die Daten, Datenbasen und Datenströme

2.1.1. Datenkatalog mit Speichergrößen

Es wird davon ausgegangen, dass für die Speicherung folgender Platz aufgewendet wird:

-gZahl: 2 Bytes

-rZahl: 4 Bytes

-Zeichenkette: 1 Byte je Zeichen der maximalen Länge

<u>Element</u>	<u>Strukturbeschreibung</u>	<u>Bytes</u>
Vorstellungen	{Vorstellung}	
Vorstellung	@Vorst Nr + Vorst_Datensatz	2+68=70
Vorst Nr	gZahl *3-stellig, positiv*	2
Vorst_Datensatz	Datum + Filmtitel + Uhrzeit + Dauer + Saal_Nr + Preis	6+50+4+2+2+4=68
Datum	Tag + Monat + Jahr	6
Filmtitel	Zeichenkette50	50
Uhrzeit	Stunden + Minuten	4
Dauer	gZahl *3-stellig, 0 <= Dauer <=300*	2
Saal_Nr	gZahl *2-stellig, 1 <= Saal_Nr <= 20*	2
Preis	Geldbetrag	4
Tag	gZahl *2-stellig, 1 <= Tag <= 31*	2
Monat	gZahl *2-stellig, 1 <= Monat <= 12*	2
Jahr	gZahl *4-stellig, 2004 <= Jahr <= 2100*	2
Stunden	gZahl *2-stellig, 00 <= Stunden <= 23*	2
Minuten	gZahl *2-stellig, 00 <= Minuten <= 59*	2
Geldbetrag	rZahl *6-stellig, davon 2 Nachkommastellen*	4
Säle	{Saal}	
Saal	@Saal Nr + Anz Reihen + Breite Reihen	2+2+2=6
Anz Reihen	gZahl *2-stellig*	2
Breite_Reihen	gZahl *2-stellig ... gemeint ist hier Anzahl der Plätze je Reihe*	2
Reservierungen	{Reservierung}	
Reservierung	@Abhol Nr + Vorst Nr + Reihen Nr + Anz Plätze	2+2+2+2=8
Abhol Nr	gZahl *6-stellig*	2
Reihen Nr	gZahl *2-stellig*	2
Anz_Plätze	gZahl *2-stellig*	2
Reihen	{Reihe}	
Reihe	@ID + Vorst_Nr + Reihen_Nr + Anz_gesamt + Anz_frei + max_Gruppe	2+2+2+2+2+2=12
ID	gZahl	2
Anz_gesamt	gZahl * >0 *	2
Anz_frei	gZahl * >=0 *	2
max_Gruppe	gZahl * > 0 *	2
Plätze	{Platz}	
Platz	@ID + Platz_Nr + Reihen_Nr + belegt	2+2+2+2=8
Platz_Nr	gZahl *2-stellig*	2
belegt	[0 1]	1
findet statt in	@Vorst Nr + Saal Nr	4
reserviert für	@Abhol Nr + Vorst Nr	4
reserviert in	@Abhol Nr + <Reihe>ID	4
sieht	@<Reihe>ID + Vorst Nr	2+2=4
liegt_in	@<Platz>ID + <Reihe>ID	2+2=4
Spielplan	{Vorstellung}	
Sitzplan	{Reihe + {belegte Platz Nr}}	
belegte Platz Nr	Platz Nr	2
Geld rein	Geldbetrag	4
Geld raus	Geldbetrag	4

2.1.2. Abschätzungen

Es wird davon ausgegangen, dass im System die Vorstellungen und Reservierungen einer Woche gespeichert werden. Um die zu speichernden Datenmengen einschätzen zu können, werden drei Kinotypen definiert.

- kleines Kino (minimaler Speicheraufwand)
- mittleres Kino (durchschnittlicher Speicheraufwand)
- großes Kino (maximaler Speicheraufwand)

Das **kleine Kino** hat **einen Saal** mit 5 Reihen á 5 Sitze und **2 Vorstellungen** am Wochenende. Es werden 10 Besucher pro Woche angenommen. **Reserviert hat keiner.**

Das **mittlere Kino** hat **6 Säle** mit 15 Reihen á 15 Sitzen. Pro Tag finden hier je Saal 2 Vorstellungen statt. Zusätzlich finden am Wochenende (Fr, Sa, So) in 4 Sälen Spätvorstellungen statt. Das macht **96 Vorstellungen**. Mit 50% verkaufter Karten sind das etwa 10.000 Besucher je Woche. Vorbestellt wurden davon etwa 60% zu je drei Karten; d.h. etwa **2.000 Reservierungen**.

Im **großen Kino** gibt es **15 Säle** mit 20 Reihen á 30 Sitzen. Es finden hier täglich pro Saal 4 Vorstellungen statt. Das ergibt **420 Vorstellungen** und etwa 250.000 Besucher pro Woche (bei Ausverkauf der Karten). Es liegen auch **250.000 Reservierungen** für jeden Platz einzeln vor.

2.1.2.1. Speicherbedarf

Speicher	Größe je Satz	Kleines Kino	Mittleres Kino	Großes Kino
Reservierungen	8	$0*8=0$	$2.000*8=16k$	$250.000*8=2M$
Vorstellungen	70	$2*70=140$	$96*70=6720$	$420*70=26776$
Säle	6	$1*6=6$	$6*6=36$	$15*6=90$
Reihen	12	$2*5*12=120$	$6*15*12=1080$	$15*20*12=3,6k$
Plätze	8	$2*5*5*8=400$	$6*15*15*8=10800$	$15*20*30*8=72k$
	Gesamtbedarf:	<u><i>666B</i></u>	<u><i>34636B</i></u>	<u><i>2102466B</i></u>

Im Mittel kommen wird also ein Speicherbedarf von etwa 34 kB angenommen. Bei einem sehr großen Kino werden etwa 2 MB benötigt.

2.1.2.2. Datenflüsse

Datenfluss	Größe je Element	Kleines Kino	Mittleres Kino	Großes Kino
Spielplan	70	$2*70=140$	$96*70=6720$	$420*70=29,4k$
Sitzplan	---	$5*(12+5*(2))=600$	$15*(12+15*(2))=630$	$20*(12+30*(2))=1440$
Karten	68	$5*68=340$	$15*68=1020$	$30*68=2040$

Am meisten ins Gewicht fallen die Spielplananfragen, welche beim großen Kino mit knapp 30 kB zu Buche schlagen.

2.1.3. Genauigkeit

Bis auf die Bezahlfunktion arbeitet das System nur mit Ganzzahlen. Bei den für die Bezahlfunktion verwendeten Zahlen handelt es sich um Geldbeträge, die auf zwei Nachkommastellen genau eingegeben werden. Da sie mit ihnen nie dividiert wird, entstehen keine neuen Nachkommastellen, so dass die Speicherung/Ausgabe ohne Rundungen auskommt. Es entstehen also keine Ungenauigkeiten.

2.2. Operationelle Anforderungen an die Prozesse

Der einzige Prozess, in dem rekursiver Rechenaufwand benötigt wird, ist der Prozess Reservierte_Karten_verkaufen

3. Spezifikation wichtiger Qualitätsanforderungen

3.1. Benutzbarkeit

Die Nutzer des Systems gehören entweder den Nutzergruppen Administrator, Kasse oder Besucher an. In allen Fällen wird dabei die Bedienung so einfach wie möglich gestaltet. Grundlage dafür ist eine grafische Oberfläche, deren Benutzung in wenigen Worten erklärbar ist, und wenige Vorkenntnisse im Umgang mit Computern erfordert.

3.2. Zuverlässigkeit

Es ist vorgesehen, eine selbsterklärende und bedienerfreundliche Benutzeroberfläche zur Vermeidung von Bedienfehlern zu entwerfen. Dies beinhaltet u.a. auch eine effektive Fehlererkennung sowie verständliche Fehlerhinweise.

Die Daten im Datenbanksystem können indirekt gesichert werden, indem eine Bestätigung an den Nutzer erst erfolgt, wenn die Daten erfolgreich in die Datenbank geschrieben wurden. (Dies betrifft in erster Linie Platzreservierungen, Kartenverkauf und Änderungen im Spielplan.) Sollte ein Fehler während des Speichervorgangs auftreten (z.B. Absturz), erfolgt keine Erfolgsmeldung an den Nutzer und die Änderungen sind erneut vorzunehmen.

Die Sicherheit der gespeicherten Daten ist nicht Bestandteil des zu entwickelnden Systems und wird unabhängig von diesem gewährleistet.

3.3. Integrität

Das System für die Gäste bedarf keinerlei Zugriffsschutz, da es jedem erlaubt sein sollte, sich die Belegung anzusehen oder Karten vorzubestellen.

Das System für die Kassen besteht aus einem Programm, welches nur auf den Kassenrechnern installiert ist. Da die E/A-Geräte hinter dem Tresen stehen, ist auch hier unbefugter Zugriff ausgeschlossen.

Die Administratorfunktionen stehen auch nur in einem Programm zur Verfügung, welches nicht öffentlich zugänglich ist.

Ein Login mit Nutzerkennzeichen und/oder Passwörtern ist somit an keiner Stelle erforderlich.

3.4. Flexibilität

Erweiterungen sind nicht vorgesehen. Da das Programm nur auf den Kassensrechnern laufen soll, braucht das System auch nur für diese eine Plattform erstellt werden.

3.5. Portabilität

Eine Portierung des Systems ist nicht notwendig und es müssen auch weiterhin keine Veränderungen vorgenommen werden um eine Portierung zu erreichen. Ein Portierungsminimum ist bereits durch die vorliegenden Programmteile erreicht.

4. Basismaschine und Entwicklungsumgebung

4.1. Basismaschine

4.1.1. Hardware

Minimale Systemkonfiguration für das Kinokassenprogramm:

- IBM oder 100% kompatibler PC, Tastatur, Maus, VGA-fähiger Monitor
- Festplatte mit einer verfügbaren Kapazität von min. 10 MB
- CDROM-Laufwerk

Minimale Systemkonfiguration für das Onlineportal:

- Internetfähiger Rechner

4.1.2. Betriebssystem

für das Kinokassenprogramm:

Windows 9x/ME/2000/XP

4.1.3. Sonstige Basissoftware

für das Kinokassenprogramm:

Webserver (z.B. Apache)

MySQL Server

für das Onlineportal:

Webbrowser (z.B. IE5/6, Mozilla, ...)

4.2. Entwicklungsumgebung

Als Entwicklungsumgebung ist Delphi 6/7 vorgesehen.

5. Planung