

Konzept-Review

Reviewteam: 4-1

Claudia Laube: 1415897

Daniel Gogl: 1518480

Sebastian Grabher: 1518792

Softwareteam: 4

Proseminargruppe: 1

Datum: 28.03.2017

Hinweis: Gestalten Sie den Review in konstruktiver Art und Weise. Konkrete Vorschläge können anhand von Kommentaren und Vorschläge im Konzeptpapier selbst gemacht werden. Bitte beachten Sie, dass das Projektteam auf Ihren Review schriftlich Stellung nehmen muss.

Zusammenfassung

Das Konzept beschreibt eine webbasierte Verwaltungssoftware für Kinderkrippen. Dabei werden die grundsätzlichen Funktionen für die Verwaltung unterstützt und noch weitere zusätzliche Features implementiert.

Das Systemkonzept bedient die Anforderungen der Aufgabenstellung. Manche Details wurden aber übersehen oder nicht genau ausgearbeitet. Die Planung wirkt durchdacht, hingegen das Klassendiagramm weist einige Schwächen und Inkonsistenzen auf.

Folgende Mängel wurden festgestellt:

Grobe Mängel:

- Eintragen einer Maximalbelegung pro Tag fehlt im gesamten Projekt
- Es fehlt die Möglichkeit Preise, Stichtag, und Stichzeit des Mittagessens zu konfigurieren
- Fehlende Beschreibungen von Use Cases die in anderen inkludiert sind
- Der Audit-Log wird nur im Systemüberblick und im Projektplan erwähnt
- Das Klassendiagramm sollte noch einmal überarbeitet werden und auch mit den Use Cases abgeglichen werden (Involvierte Klassen auch zu den Use Cases schreiben, damit Zusammenhang klarer wird)

Mittelschwere Mängel:

- Ablauf der Use Cases unklar
- Es ist unklar, wie bestimmt wird welche Kinder an welchen Tagen anwesend sind und was die Öffnungszeiten der Kinderkrippe sind
- Benennungen teilweise inkonsistent
- Formfehler bei der textuellen Beschreibung der Use Cases
- Fehlende Aufteilung der zukünftiger Verantwortungen

Geringe Mängel:

- In der Textbeschreibung zum Klassendiagramm, werden die Begriffe Teacher und Guardian erwähnt. Vermutlich ist dabei Personal und Bezugsperson gemeint. Dies war nicht klar ersichtlich
- Was bedeutet "Mädchen für alles" als Verantwortungsbereich?

1. Systemüberblick

Die betroffenen Personen und Akteure sind klar definiert.

Das System wirkt von der Oberfläche sehr einfach, was für die Zielgruppe von Kindergartenpersonal und Eltern sehr geeignet ist.

2. Use Cases

Die Akteure, Voraussetzung und das Ergebnis werden immer schön beschrieben, die Übersichtlichkeit ist auch gegeben, es wird aber oft nicht auf den genauen Ablauf eingegangen. Es haben sich einige Formfehler eingeschlichen, was aber nicht weiter

schlimm ist. Die Benennung der Use Cases und Akteure ist teilweise inkonsistent (Aufgaben verwalten/Aufgaben & Eltern/Elternteil). Manche Use Cases beschreiben ihre inkludierten Use Cases, diese sollten eigene Use Case bilden und ihre Vorgänger sollten darauf verweisen (z.B. Use Case "als pdf exportieren").

Bis auf den Use Case Anmeldung zum Mittagessen sind alle Use Cases aus dem Diagramm im Detail beschrieben.

In Summe stimmen die aufgelisteten Use Cases fast mit dem Lastenheft überein, einige Punkte sind aber unvollständig:

- Der Audit-Log und das Eintragen der Maximalbelegung ist nicht enthalten
- Eltern sollten laut Lastenheft nicht in der Lage sein neue Kinder hinzuzufügen
- Ungenau beschriebener Use Case "Monatliche Reports" (es wird nicht darauf eingegangen, dass automatisch eine Aufgabe erzeugt wird, die als Abrechnung für Mittagessens/Monat/Kind dient
- Fehlender Use Case "Mittagessen: Preis, Stichtag/Stichzeit ändern)

Um das Zusammenspiel zwischen Klassendiagramm und Use Cases deutlicher und übersichtlicher zu machen, sollte zu jedem Use Case auch die involvierten Klassen angegeben werden

3. Fachliches Klassendiagramm

Das Klassendiagramm ist übersichtlich gestaltet und die Relationen sind eindeutig beschriftet. Die textuelle Beschreibung verdeutlicht das Zusammenspiel der Klassen, welches ohne diese etwas schwer nachzuvollziehen wäre.

Dies ist hauptsächlich fehlenden Relationen, Klassen und Attributen geschuldet. Bei den Relationen werden gewisse Attribute sowohl als Attribut als auch als Beziehung dargestellt. Hier wäre es besser sich für eins von beiden zu entscheiden, um die Konsistenz zu verbessern (siehe zum Beispiel ParentCalender Attribut child und TeacherCalendar Attribut attendantChilds).

Nicht ganz klar war zuerst das "Teacher" für Personal und "Guardian" für Bezugsperson steht.

Einige in den Use Cases beschriebene Vorgänge sind mit dem derzeitigen Klassendiagramm gänzlich nicht umzusetzen, dazu zählen: Login/Logout, Aufgaben verwalten, Abmeldung von Kindern, sowie einige Sonderfeature Use Cases.

Andere sind nur eingeschränkt implementierbar, da zum Beispiel einige Attribute fehlen, das ist aber nicht weiter schlimm, da diese nur einfach noch hinzugefügt werden müssen.

Dadurch kommt es auch zu kleine Unstimmigkeiten beim GUI-Prototyp, wo zum Beispiel bei der Kontaktliste Attribute genannt sind, die nicht im Klassendiagramm sind.

Bei den Relationen ist bei ParentCalender nicht klar, wie Bezugspersonen mit der Klasse verbunden sind, oder wie Bezugspersonen mit den Eltern/Kindern verbunden sind. Unklar war uns auch die Funktion der Klasse TeacherCalender, da sie einerseits Ferienzeiten zu speichern scheint, andererseits aber auch eine Liste von anwesenden Kindern, allerdings ohne konkretes Datum oder konkreten Wochentag, enthält. Letzteres ist aber eine Funktionalität die auch von der Klasse DayPlanner bereitgestellt werden soll (Methode getAttendantChilds()) und wirkt dadurch

redundant.

Es gibt einige fehlende Attribute, so fehlt zum Beispiel beim Kind unter anderem noch das An- und Abmeldedatum, das laut Use Case "Kind abmelden" und Lastenheft aber vorhanden sein sollte. Auch ist unklar, wo gespeichert wird, an welchen Tagen ein Kind in der Krippe ist, da es zwar Standard Bring- und Abholzeiten gibt, aber keine Standardanwesenheitstage (falls dies nicht für jede Woche extra geschehen soll). Weiters fehlen einige Attribute, die bei Eltern und Bezugspersonen gespeichert werden müssten (zum Beispiel Foto oder Beziehung zum Kind). Es ist auch nicht möglich die Eltern zu inaktiven Usern zu machen, was laut Use Case "Kind abmelden" möglich sein sollte.

Bei der Klasse Person (die ja als Klasse für alle Benutzer, die sich im System anmelden können, verwendet wird) fehlen die Attribute E-Mailadresse und Passwort, die laut Usecase "Login/Logout" zum Anmelden verwendet werden sollten.

Es fehlt zusätzlich eine Möglichkeit global den Stichtag, die Stichtzeit und den Preis für das Mittagessen zu konfigurieren und zu speichern.

Eine Sache die auf jeden Fall noch geändert werden sollte ist, das Bezugspersonen keine Benutzer des Systems sein sollten, so wie sie es derzeit wären (da alle Instanzen der Klasse Person Benutzer sind), weil laut Lastenheft nur Eltern und Personal das System verwenden und eine Bezugsperson auch hier keinen Zugriff auf Kinder hat (nur Eltern haben Zugriff auf ParentCalender). Es sollte dann auch der Spezialfall berücksichtigt werden, dass es Personen gibt die einerseits für ein oder mehrere Kinder Elternteile sind und zum Beispiel zusätzlich auch noch als Bezugsperson für ein anderes Kind gelten.

Auch fehlt eine Klasse um die Aufgaben, die an die Eltern gesendet werden sollen zu speichern, stattdessen gibt es eine Klasse Mail, die nur speichert wer der Absender und wer der Empfänger ist. Der Mailverkehr sollte außerhalb des Systems stattfinden, es können zwar automatisch E-Mails verschickt werden, aber dazu sollte es nicht notwendig sein, einen Sender und einen Empfänger zu speichern, sondern stattdessen beispielsweise gewisse Standardemailtexte.

4. GUI-Prototyp

Der GUI-Prototyp wurde einheitlich mit einem Tool gezeichnet.

Leider fehlen einige Use Cases, wie die Aufgabendarstellung oder Bezugspersonen zu Kindern zugeordnet werden. Auch werden Attribute im GUI-Prototyp erwähnt, die im Klassendiagramm nicht sind, wie zum Beispiel eine Auftrennung von Name als Attribute zu Vorname und Nachname in der Kontaktliste.

Das Layout wirkt leicht bedienbar. Die Elemente sind bündig angeordnet und es werden Elemente wie Radiobuttons und Dropdown-Menüs verwendet. Die Navigation befindet sich im Vollbildmodus am oberen linken Rand, hingegen aktive Elemente wie Speichern und Hinzufügen sind immer in der rechten Oberen Ecke. Bei Popups befinden sich die Buttons einheitlich unten.

5. Projektplan

Grundsätzlich sind die Aufgaben auf den Mitgliedern gut aufgeteilt. Die Verantwortlichkeiten scheinen aber nur für das Konzept und nicht für die eigentliche

Implementierung aufgeteilt worden zu sein.

Im Git wird auch das Issue-System verwendet, wo auch schon Aufgaben für Personen eingeteilt wurde.

Der Umfang ist in zweiwöchentliche Segmente aufgeteilt. Die Implementierung beginnt mit den Grundfunktionalitäten und gegen Ende werden nur mehr Zusatzfeatures umgesetzt, was eine sinnvolle Planungsstruktur ist. Das letzte Segment nur für Tests und Planung einzuplanen wirkt ebenfalls plausibel. Zeitlich erscheint diese Planung umsetzbar.