PS Softwareentwicklung und Projektmanagement

Konzept-Review

Reviewteam: 4-2

Simon Driendl: 1416348
Elisabeth Arzberger: 0923737

Softwareteam: 4

Proseminargruppe: 1

Datum: 28.03.2017

Zusammenfassung

Im Großen und Ganzen erscheint das Konzept gelungen. Die meisten Anforderungen wurden spezifiziert und die GUI-Prototypen wirken benutzerfreundlich. Uns sind einige Inkonsistenzen aufgefallen, die wir in einem Überblick nach den Richtlinien des Foliensatzes SEPMPS_SS17_1_Konzeptreview klassifiziert haben und im Folgenden noch genauer erläutern werden. Insbesondere das Klassendiagramm sollte noch überarbeitet werden.
Eine gute Projektplanung ist vorhanden.

Grobe Mängel

- Fehlende Klasse/Attribute zum Anlegen eines Userobjekts (Username, Passwort, Email).
- Fehlende Klasse/n zur Abdeckung des Anforderungsbereiches “Aufgaben verwalten”
- Fehlende Attribute, um Personen/Kinder zu inaktivieren
- Fehlende Klasse/Attribute, um Wochentage für die Anmeldung zu speichern
- Folgende fehlende Usecases:
    - Eintragen von freien Tagen
    - Eintragen der täglichen Maximalbelegung durch das Personal
    - Festlegen der täglichen Bring- und Abholzeiten durch das Personal
    - Einsicht des Audit-Logs

Mittelschwere Mängel

- Fehlende Angabe von involvierten fachlichen Klassen bei den Usecases
- Erstellen einer Mittagessen-Abrechnung fehlt beim Usecase Monatliche Reports
- Keine direkte Beziehung von Child zu Person - problematisch in Datenbank
- 1 zu 1 Beziehung Person-ParentCalendar problematisch
- Vermischung zwischen Modelklassen und Controllerfunktionen

Geringfügige Mängel

- Konzept-Angabe aus dem Proseminar nach dem Titelblatt kann weggelassen werden

Systemüberblick

Der Systemüberblick ist gut verständlich, die Zielgruppe wird eindeutig definiert.
Die zu unterstützenden Abläufe werden sehr detailliert - ähnlich zu einer Usecase-Übersicht aufgelistet.

Use Cases

Ein schönes Usecasediagramm ist vorhanden. Auch Zusatzfeatures aus dem Lastenheft wurden vom Team bedacht.
Die Beschreibung der Usecases könnte noch übersichtlicher sein, wenn man sie nach Initiator ordnen würde(z.B. zuerst alle Usecases mit Initiator/Person Personal, dann Usecases mit Initiator Eltern, dann gemeinsame Usecases) .

Laut Lastenheft sollte das Personal die Zeiträume für das Bringen und Abholen vorgeben können.
In den Usecases ist der Ablauf aber so beschrieben, als könnten Eltern beliebige Hol- und Bringzeiten für ihre Kinder eintragen, ohne dass das Personal darauf Einfluss nehmen kann.
Auch die in der Angabe geforderten Möglichkeiten, Feiertage und die tägliche Maximalbelegung einzutragen, werden nicht beschrieben.
In den Vorlesungsunterlagen gibt es ein Beispiel für die textuelle Beschreibung von Usecases, das auch die Angabe von involvierten fachlichen Klassen beeinhaltet. Diese Angabe könnte hier auch noch hinzugefügt werden, so dass man die Übereinstimmung der Usecases mit dem Klassendiagramm direkt überprüfen kann.
Die Funktionalität Einsehen eines Audit-Logs wird in der Systemübersicht erwähnt, ist jedoch nicht in den Usecases enthalten.
Der Usecase Monatliche Reports behandelt nur die Essens-Anmeldung, sollte laut Angabe jedoch auch das Erstellen einer Abrechnung beeinhalten, was auch beim Erstellen der Datenstruktur bedacht werden sollte.
Ansonsten sind die Usecases gut verständlich geschrieben.

Fachliches Klassendiagramm

Uns ist besonders aufgefallen, dass eine Klasse/Felder zum Anlegen eines Users bzw. Benutzerkontos (Username, Passwort, E-Mail-Adresse) fehlen, obwohl der Usecase Login/Logout vorhanden ist und die Angabe einer Email auch im Usecase Elternteil anlegen als verpflichtend genannt wird.
Ihr habt zwar eine Klasse UserAdministration , diese sieht allerdings aus wie ein Controller für die Userverwaltung, ein tatsächliches Userobjekt ist nicht vorhanden.
Der Zusammenhang Parent-Children soll anscheinend über die indirekte Beziehung Person with Role Parent - Parent Calendar -Child hergestellt werden. Jedoch ist bereits die 1:1 Beziehung Person - ParentCalendar problematisch, weil dies bedeutet, dass pro Person jeweils immer nur eine Lunchanmeldung, eine Abwesenheit des Kindes etc. abgespeichert werden kann und jedesmal bei einem erneuten Setzen überschrieben werden muss.(Die Daten jeweils als Array abzuspeichern, wäre aus Datenbanksicht ein Problem, da dann keine normalisierten

Tabellen mehr vorliegen). Wir würden empfehlen, eine direkte Beziehung Person(with role parent) - Child einzuführen, so dass in der Datenbank Personen ein oder mehrere Kinder zugeordnet werden können. PersonCalendar kann dann entsprechend sowohl mit Person, als auch mit Child verknüpft werden.
Für die gewünschte Funktionalität des Systems wird außerdem noch eine Datenstruktur (zB. eine Klasse Task) benötigt, die den Bereich Aufgaben abdeckt.
Im Usecase Kind abmelden und auch im Lastenheft wird erwähnt, dass Eltern, deren Kinder abgemeldet wurden, nur noch eingeschränkte Benutzerrechte haben, indem sie als inaktiv markiert werden. Ein entsprechendes Attribut könnte bei Person bzw. Child hinzugefügt werden, genauso wie eine Telefonnummer. Auch eine Struktur zum Speichern der Wochentage, für die das Kind angemeldet ist, fehlt noch.
Die Klasse TeacherCalendar ermöglicht das Eintragen von freien Tagen, wie es auch im Lastenheft gefordert wird - diese Möglichkeit sollte in den Usecases noch erwähnt werden.
Eine kleine Inkonsistenz stellt für uns die Tatsache dar, dass die genannten Rollen Admin, Teacher, Parent und Guardian in den Usecases nicht, oder mit deutscher Bezeichnung erwähnt werden. So stellt sich uns beispielsweise die Frage, ob es überhaupt Usecases gibt, die nur auf die Rolle Admin zutreffen. Außerdem würden wir auf die Angabe von Methoden(z.B. zahlreiche Getter, Methoden wie assignChild()) im fachlichen Klassendiagramm verzichten, da wir dies dem Bereich Controller zuordnen würden.
Die Beschreibung des Klassendiagramms war sehr hilfreich bei der Durchsicht des Klassendiagramms.

GUI-Prototyp

Die Idee, einen Elternkalender zu erstellen, in dem Eltern relevante Daten für die tägliche Organisation an einer Stelle eintragen können, wirkt benutzerfreundlich und hat uns besonders gefallen.

Projektplan

Es wurden keine Verantwortlichkeiten für das Projekt aufgeteilt, nur für das Konzept. Es steht zwar dabei, dass Verantwortlichkeiten später noch aufgesplittet werden sollen, dies hätte man aber etwas genauer in die Tabelle eintragen können.
Die Reihenfolge der geplanten Inkremente erscheint sinnvoll und deren Umsetzung realistisch. Die einzelnen Segmente werden übersichtlich in einer Tabelle präsentiert und sind detailliert beschrieben.
Testen und Debuggen ist als eigenes Segment vor der Abgabe eingeteilt. Dies ist sinnvoll zur abschließenden Kontrolle, wir gehen davon aus, dass Testen implizit auch während des gesamten Projektverlaufes stattfindet. Insgesamt ist uns die Projektplanung sehr positiv aufgefallen.