- Entwurfsbeschreibung – Gesamtprojekt -

Kita Tauschbörse

Version: 1.1

Projektbezeichnung	Kita Tauschbörse	
Projektleiter	F. Teichmann	
Verantwortlich	B. Kalloch, R. Rößling	
Erstellt am		
Zuletzt geändert	30.04.12 14:43	
Bearbeitungszustand		in Bearbeitung
		vorgelegt
	X	fertig gestellt
Dokumentablage		
V-Modell-XT Version	1.3	

INHALTSVERZEICHNIS

1 ALLGEMEINES

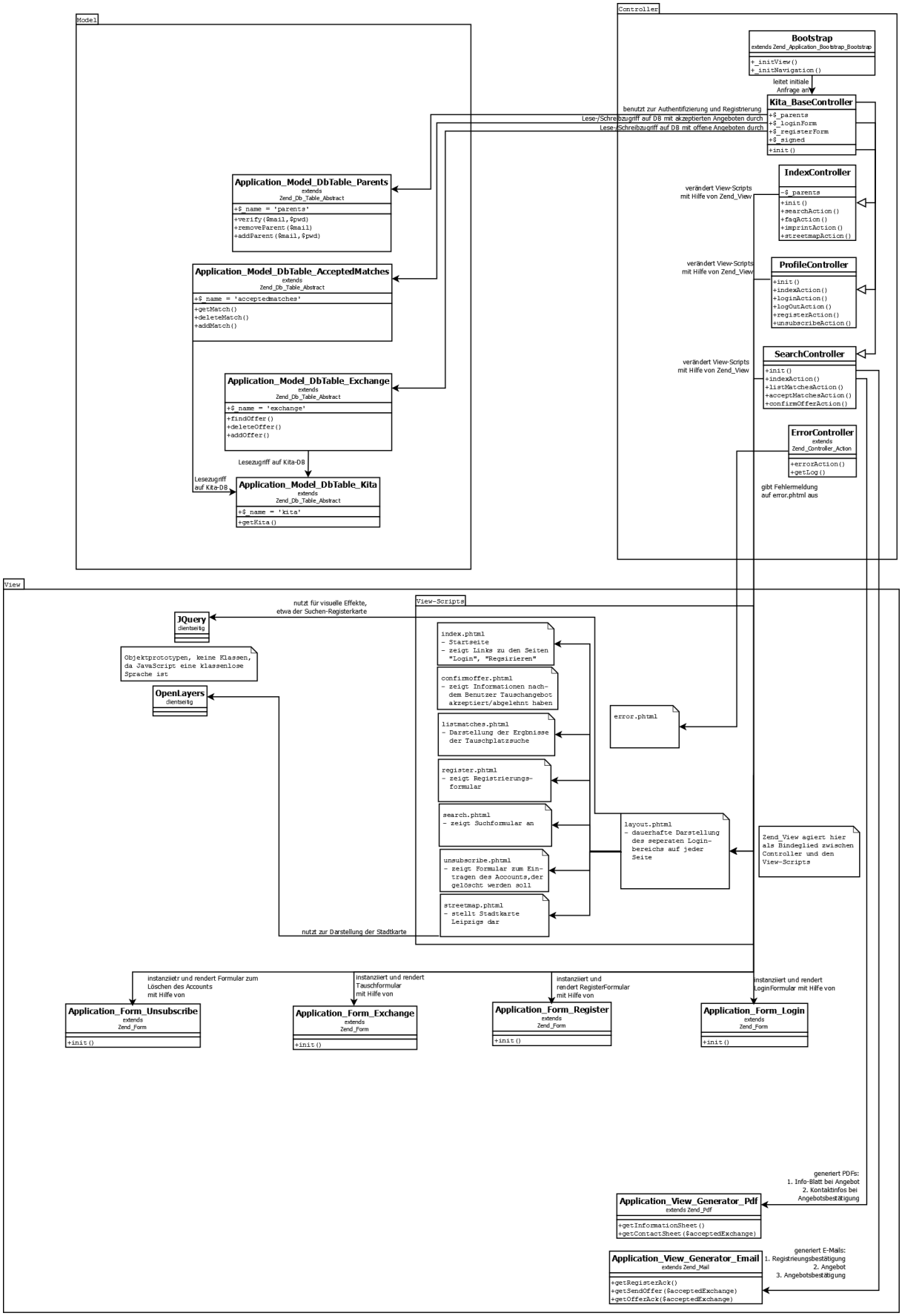
Unser Projekt ist ein zentralisiertes Tauschbörsensystem, welches seinen Service als reine Webapplikation bereitstellt. Es soll Familien mit Kleinkindern dabei helfen, auf einem möglichst einfachen und unkomplizierten Weg Kita-Plätze eintauschen zu können, welche in einer, für sie, besseren Region liegen.

2 PRODUKTÜBERSICHT

Die Nutzer können sich anonym an das System mit ihrer E-Mail Adresse registrieren und anmelden, um einen Tausch aufzugeben. Eine durch OpenLayers visualisierte Karte soll bei dem Finden von tauschbaren Kita-Plätzen helfen. Um einen Tausch einstellen zu können und um dadurch relevante Ergebnise zu erhalten, müssen der momentane Kita-Platz und das Alter des Kindes, sowie der gewünschte Ort angegeben werden.
Diese Daten werden temporär via MySQL in einem Datenbanksystem, neben den Nutzerdaten und Geodaten der Kita-Plätze, gespeichert. Falls nach einer gewissen Zeit noch kein Tausch zustande gekommen ist, wird der Tauscheintrag wieder aus Datenbank entfernt. Wurde jedoch ein passender Tausch gefunden, so werden durch das Zend Framework generierte .pdf Dokumente an beide Parteien entsandt, um einen Tausch zu initiieren.

3 GRUNDSÄTZLICHE STRUKTUR- UND ENTWURFSPRINZIPIEN FÜR DAS GESAMTSYSTEM

Für das Hauptprojekt behalten wir das MVC-Design bei und erweitern dessen elementare Bestandteile, den View, den Controller und das Model. Das folgende UML-Klassendiagramm stellt den Aufbau unseres Projektes dar:


Model
Application_Model_DbTable_Parents
extends
Zend_Db_Table_Abstract
+$_name = 'parents'
+verify($mail,$pwd)
+removeParent($mail)
+addParent($mail,$pwd)
Application_Model_DbTable_AcceptedMatches
extends
Zend_Db_Table_Abstract
+$_name = 'acceptedmatches'
+getMatch()
+deleteMatch()
+addMatch()
Application_Model_DbTable_Exchange
extends
Zend_Db_Table_Abstract
+$_name = 'exchange'
+findOffer()
+deleteOffer()
+addOffer()
Lesezugriff auf Kita-DB
Lesezugriff
auf Kita-DB
Application_Model_DbTable_Kita
extends
Zend_Db_Table_Abstract
+$_name = 'kita'
+getKita()
Controller
Bootstrap
extends Zend_Application_Bootstrap_Bootstrap
+_initView()
+_initNavigation()
leitet initiale
Anfrage an
Kita_BaseController
+$_parents
+$_loginForm
+$_registerForm
+$_signed
+init()
benutzt zur Authentifizierung und Registrierung
Lese-/Schreibzugriff auf DB mit akzeptierten Angeboten durch
Lese-/Schreibzugriff auf DB mit offene Angeboten durch
IndexController
-$_parents
+init()
+searchAction()
+faqAction()
+imprintAction()
+streetmapAction()
verändert View-Scripts
mit Hilfe von Zend_View
ProfileController
+init()
+indexAction()
+loginAction()
+logOutAction()
+registerAction()
+unsubscribeAction()
verändert View-Scripts
mit Hilfe von Zend_View
SearchController
+init()
+indexAction()
+listMatchesAction()
+acceptMatchesAction()
+confirmOfferAction()
verändert View-Scripts
mit Hilfe von Zend_View
ErrorController
extends
Zend_Controller_Action
+errorAction()
+getLog()
gibt Fehlermeldung
auf error.phtml aus
View
JQuery
clientseitig
nutzt für visuelle Effekte,
etwa der Suchen-Registerkarte
Objektprototypen, keine Klassen,
da JavaScript eine klassenlose
Sprache ist
OpenLayers
clientseitig
nutzt zur Darstellung der Stadtkarte
View-Scripts
index.phtml
- Startseite
- zeigt Links zu den Seiten
"Login", "Regsirieren"
confirmoffer.phtml
- zeigt Informationen nach-
dem Benutzer Tauschangebot
akzeptiert/abgelehnt haben
listmatches.phtml
- Darstellung der Ergbnisse
der Tauschplatzsuche
register.phtml
- zeigt Registrierungs-
formular
search.phtml
- zeigt Suchformular an
unsubscribe.phtml
- zeigt Formular zum Ein-
tragen des Accounts,der
gelöscht werden soll
streetmap.phtml
- stellt Stadtkarte
Leipzigs dar
error.phtml
layout.phtml
- dauerhafte Darstellung
des seperaten Login-
bereichs auf jeder
Seite
Zend_View agiert hier
als Bindeglied zwischen
Controller und den
View-Scripts
instanziietr und rendert Formular zum
Löschen des Accounts
mit Hilfe von
Application_Form_Unsubscribe
extends
Zend_Form
+init()
instanziiert und rendert
Tauschformular
mit Hilfe von
Application_Form_Exchange
extends
Zend_Form
+init()
instanziiert und
rendert RegisterFormular
mit Hilfe von
Application_Form_Register
extends
Zend_Form
+init()
instanziiert und rendert
LoginFormular mit Hilfe von
Application_Form_Login
extends
Zend_Form
+init()
generiert PDFs:
1. Info-Blatt bei Angebot
2. Kontaktinfos bei
Angebotsbestätigung
Application_View_Generator_Pdf
extends Zend_Pdf
+getInformationSheet()
+getContactSheet($acceptedExchange)
generiert E-Mails:
1. Registrieungsbestätigung
2. Angebot
3. Angebotsbestätigung
Application_View_Generator_Email
extends Zend_Mail
+getRegisterAck()
+getSendOffer($acceptedExchange)
+getOfferAck($acceptedExchange)

3.1 Controller

Das Vorprojekt deckte mit seiner Funktionalität weitestgehend den öffentlichen Bereich der Tauschbörse ab. Für das Hauptprojekt muss nun noch der private Bereich implementiert werden, welcher nur angemeldeten Benutzern zur Verfügung steht. Um diese Bereiche auch logisch zu trennen werden wir einen weiteren Controller, den *InternalController,* implementieren. Dieser beinhaltet die Funktion *listMatches()*, welche es erlaubt eine Tauschsuche entsprechend bestimmter Kriterien zu starten. Diese Funktion gibt dann eine Liste entsprechend passender Tauschangebote aus. Wenn der Nutzer eines der Angebote auswählt, wird eine Reihe weiterer Funktionen gestartet.
acceptMatchAction() regelt dann die Kommunikation mit dem anderen Elternpaar. Es wird eine E-Mail mit einem entsprechenden Informationsblatt an das andere Elternblatt gesendet.
confirmAction() wird dann ausgelöst, wenn das zweite Elternpaar auf die von *acceptMatchAcion()* gesendete E-Mail reagiert. Bei Akzeptieren des Tauschangebotes wird ein Eintrag in der Datenbank für akzeptierte Tausche erstellt.
Mit *unsubscribe()* wird den Eltern auch die Möglichkeit gegeben ihren Account zu löschen.
Nach einer Sitzung können sich die Eltern mit *logout()* beim System abmelden.
In regelmäßigen Abständen startet das System selbstständig eine Funktion zur Bereinigung seiner Datenbestände. Hierfür ist die Funktion *garbageCollection()* zuständig.

3.2 Model

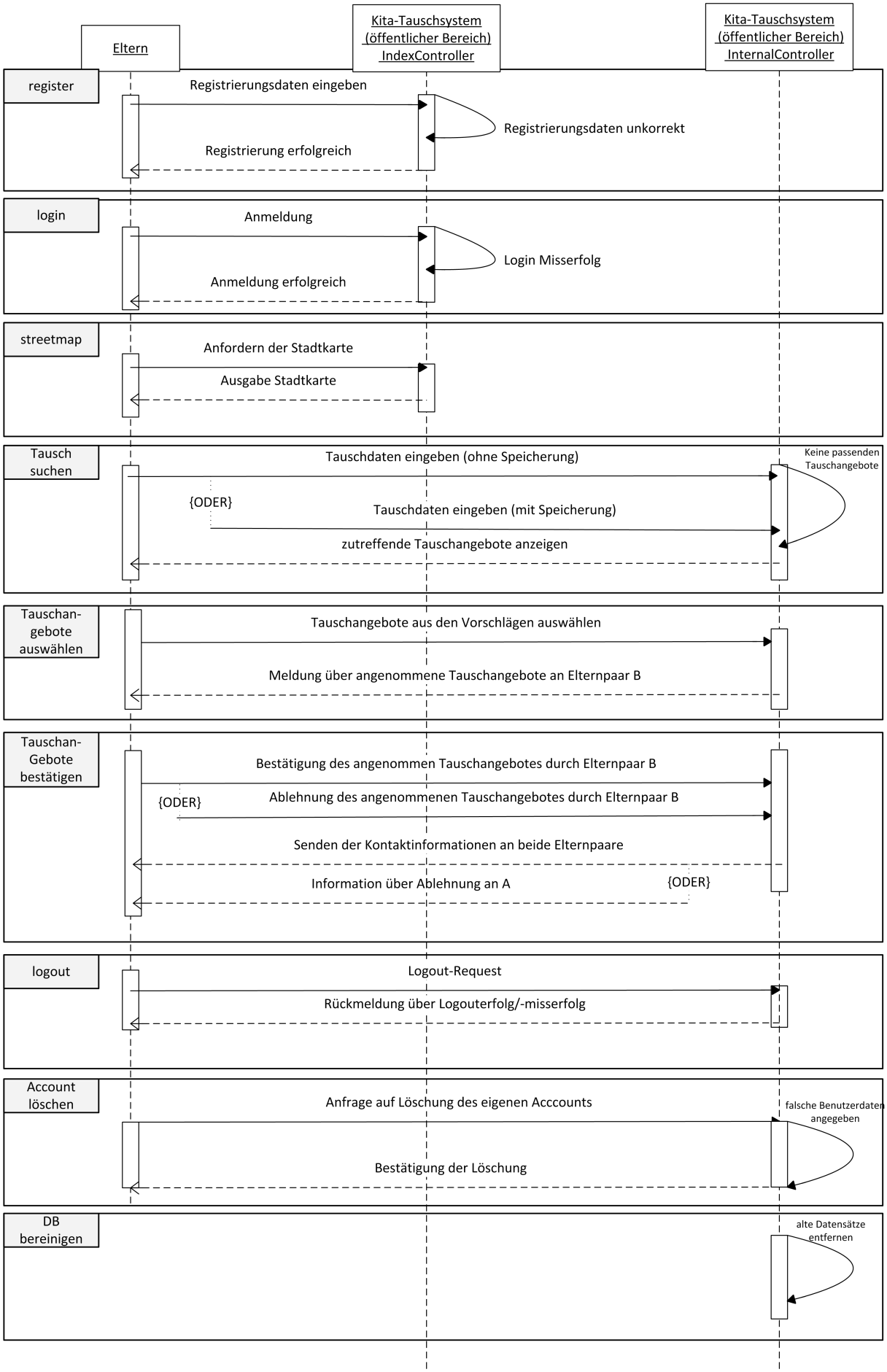
Der Model-Teil des MVC Musters wurde um weitere Klassen erweitert, die den Datenbankzugriff kapseln. Zum einen sorgt *Application_Model_DbTable_AcceptedMatches* dafür, dass auf die Tabelle der bereits bestätigten Tauschangebote zugegriffen werden kann. Das sind diejenigen Tauschangebote, für die zwei Elternpaare an einem gegenseitigen Tausch interessiert sind, bei denen also ein Vertrag zustande kommen könnte.
Das Modell *Application_Model_DbTable_Exchange* kapselt den Zugriff auf die Datenbank aller offenen Tauschangebote. Die Models enthalten Funktionen zum Einfügen, Auslesen und Löschen von Datensätzen in der Datenbank. Alle Funktionen werden mit bestimmten Parametern aufgerufen, um den betreffenden Datensatz eindeutig identifizieren zu können. Das sind bei den Tauschangeboten etwa die Altersgruppe (*$age*), die Preiskategorie, (*$category*) Standort der Kita (*$place*) und das gewünschte Tauschdatum (*$appointedDate*). Beiden Models wird eine Klasse zur Verfügung gestellt, mit deren Hilfe sie auf die Datenbank der Kitas zugreifen können. Das ist *Application_Model_DbTable_Kita*. Jeder Kita-Platz hat eine Platzkategorie, welche ebenfalls als Tabelle in der Datenbank repräsentiert ist. *Application_Model_DbTable_KitaCategory* erlaubt der Klasse zur Verwaltung der Kindergärten Zugriff auf diesen Teil der Datenbank.

3.3 View

Zum View sind weitere View-Scripts hinzugekommen, welche etwa den Nutzer Eingabemöglichkeiten zur Tauschplatzsuche, die Ergebnisse der Suche oder auch einen Dialog zum Löschen des Accounts bieten.
Die zusätzlichen Formulare, welche zur Suche von Tauschangeboten und zum Löschen des Accounts benötigt werden, werden durch zwei weitere View_Helper (*Application_View_Helper_GetUnsubscribeForm* und *Application_View_Helper_GetExchangeForm*) bereitgestellt. Von der Seite werden auch E-Mails und PDF-Dateien als Augaben an die Nutzer erstellt. Diese enthalten etwa Informationen über den weiteren Tauschverlauf oder die Kontaktdaten der Tauschpartner. Hierfür sind zwei Klassen *Application_View_Generator_Email* und *Application_View_Generator_Pdf* zuständig welche die jeweiligen Zend-Klassen *Zend_Mail* und *Zend_Pdf* erweitern.

4 GRUNDSÄTZLICHE STRUKTUR- UND ENTWURFSPRINZIPIEN DER EINZELNEN PAKETE

Unser Projekt besitzt selbst keine weiteren Pakete deren Struktur- und Entwurfsprinzipien beschrieben werden könnten. Das gesamte System selbst wird von einem Client mittels einem JavaScript fähigen Webbrowser aufgerufen wobei die beiden Controller *IndexControllor* und *InternalController* zuständig sind, alle vom Client erwünschten Funktionen zu navigieren. In unserem Projekt sind dass die Grundfunktionen Registrieren, Anmelden, Abmelden, Account löschen, Streetmap anzeigen und Tauschplätze suchen sowie Tauschangebote bestätigen.


Eltern
Kita-Tauschsystem (öffentlicher Bereich) IndexController
Kita-Tauschsystem (öffentlicher Bereich) InternalController
register
Registrierungsdaten eingeben
Registrierungsdaten unkorrekt
Registrierung erfolgreich
login
Anmeldung
Login Misserfolg
Anmeldung erfolgreich
streetmap
Anfordern der Stadtkarte
Ausgabe Stadtkarte
Tausch suchen
Tauschdaten eingeben (ohne Speicherung)
Keine passenden Tauschangebote
{ODER}
Tauschdaten eingeben (mit Speicherung)
zutreffende Tauschangebote anzeigen
Tauschan-gebote auswählen
Tauschangebote aus den Vorschlägen auswählen
Meldung über angenommene Tauschangebote an Elternpaar B
Tauschan-Gebote bestätigen
Bestätigung des angenommen Tauschangebotes durch Elternpaar B
{ODER}
Ablehnung des angenommenen Tauschangebotes durch Elternpaar B
Senden der Kontaktinformationen an beide Elternpaare
Information über Ablehnung an A
{ODER}
logout
Logout-Request
Rückmeldung über Logouterfolg/-misserfolg
Account löschen
Anfrage auf Löschung des eigenen Acccounts
falsche Benutzerdaten angegeben
Bestätigung der Löschung
DB bereinigen
alte Datensätze entfernen