
Wishta

Das Wünsche-Android-App

Philippe Panchaud, Firat Kik, Zora Fuchs - 18. Juni 2015



Beschreibung	3
Anforderungen	3
Grundanforderungen	3
Zusatzanforderungen	3
Arbeitsaufteilung	4
Klassen-Übersicht	5
Storyboard	6
Testcases	7
Reflexion	8

Beschreibung

Die App Wishta soll ein Netzwerk für Wünsche werden. User können Wünsche erstellen und Wünsche, welche schon erstellt wurden, zu seinen eigenen hinzufügen. Die Informationen werden auf einer externen Datenbank gespeichert, damit sich der User auf jedem Handy einloggen kann und auch alle Wünsche, die bis anhin erstellt wurden, anschauen kann.

Anforderungen

Grundanforderungen

Die Grundanforderungen sind folgende:

- Es kann ein neuer User erstellt werden.
- Der User kann sich einloggen
- Es gibt ein Menu mit allen nötigen Menüpunkten
- Der User kann all seine Wünsche in einer Liste anschauen und darin suchen.
- Der User kann generelle Wünsche zu den seinigen hinzufügen.
- Der User kann neue generelle Wünsche erstellen.
- Es gibt eine externe Datenbank, auf welcher die Wünsche und User gespeichert werden.

Zusatzanforderungen

Wir sind uns bewusst, dass es nicht möglich sein wird, alle Zusatzanforderungen zu erfüllen. Wenn uns jedoch etwas Zeit bleibt, werden wir beginnen an folgenden Anforderungen zu arbeiten:

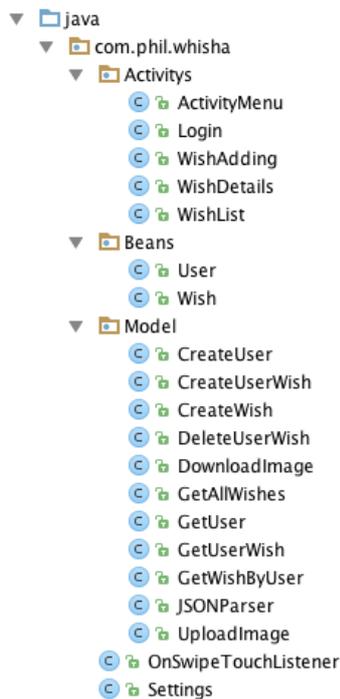
- Der User kann ein Bild zu seinem Profil hinzufügen.
- Der User kann sein Profil nachträglich abändern.
- Texte zu persönlichen Wünsche können nachträglich abgeändert werden.
- Der User kann Freunde hinzufügen, welche seine Wünsche anschauen können.
- Der User kann einem Freund Geld versprechen, damit sich dieser seinen Wunsch erfüllen kann.
- Zu den Wünschen kann man Anbieter, die das Produkt verkaufen, wie viel das Produkt bei diesem Anbieter kostet und den Link zu dem Anbieter hinzufügen.
- Man kann zu den Wünschen Tags hinzufügen, nach welchen man Suchen kann.
- Man kann ein Bild zu einem Wunsch hinzufügen.
- Sicherheit bei Datenbank und Userregistrierung berücksichtigen, zB. sql injection vermeiden, Mehrfachregistrierung vermeiden usw.

Arbeitsaufteilung

Wir haben bei diesem Projekt mit GitHub gearbeitet, was für uns alle etwas neues war. Die Zusammenarbeit hat damit sehr gut geklappt. Wir haben keine wichtige Daten verloren und die Zusammenarbeit hat auch bei grösseren Distanzen sehr gut funktioniert. Wir haben die Arbeiten wie folgt Aufgeteilt:

- Firat Kik: Design (Storyboard erstellen, Views programmieren) + Administratorisches (Kommunikation mit Lehrer)
- Philippe Panchaud: Logik (Programmieren der Activities und dem Grundgerüst des Programms)
- Zora Fuchs: Datenbank (Erstellung des Datenbankmodells, Erstellen der Datenbank, Suchen und Registrieren bei einem Server, Erstellen der Datenbankanbindung mithilfe von JSON und PHP), Dokumentation

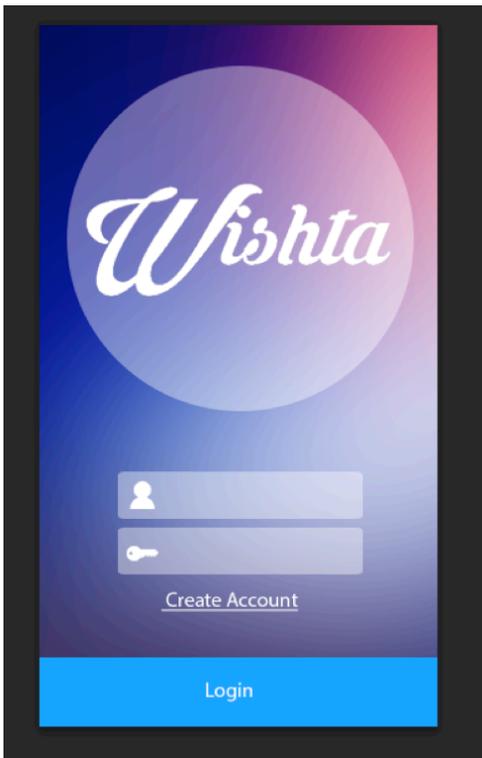
Klassen-Übersicht



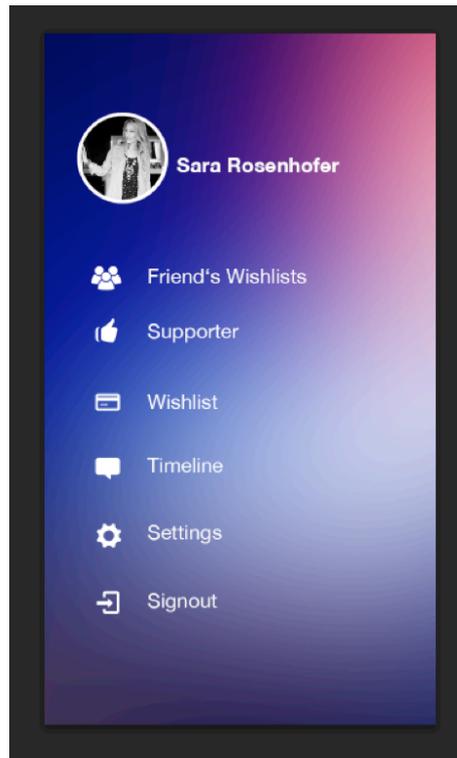
Wir haben viele Verschiedene Klassen gemacht. Im Package Activitys sind alle Activities, im Package Beans sind die Klassen, welche die Attribute für die User und Wish speichern und im Package Model findet man die AsyncTask Klassen mit ihren Helferklassen.

- ActivityMenu ist für das Menu zuständig
- Login ist die Activity für das Login
- WishAdding ist die Activity, welche für das hinzufügen von neuen Wishes verantwortlich ist. Dafür listet sie alle Wishes auf.
- WishDetails ist die Activity, welche Details zu einem Wish anzeigt.
- WishList ist die Activity, welche alle persönlichen Wishes auflistet.
- User speichert alle benötigten Daten über den User
- Wish speichert alle benötigten Daten über einen Wish
- CreateUser: Datenbankbindung um einen neuen User in der Datenbank hinzuzufügen
- CreateUserWish: Datenbankbindung um einen öffentlichen Wunsch zu den privaten Wünschen hinzuzufügen.
- CreateWish: Datenbankbindung um einen neuen Wish in der Datenbank hinzuzufügen
- DeleteUserWish: Datenbankbindung um einen privaten Wunsch zu löschen
- DownloadImage: Holt ein Bild mit einer bestimmten url vom Server herunter.
- GetAllWishes: Datenbankbindung um alle Wishes herauszulesen
- GetUser: Datenbankbindung um zu überprüfen ob eine Username/Passwort Kombination existiert und die Userdaten gerade aus der Datenbank liest.
- GetUserWish: liest fehlende Informationen aus der Userwish-Tabelle
- GetWishByUser: liest alle Wishes hinaus, die von dem User gewünscht wurden
- JSONParser ruft PHP-Script auf und holt das zurückgegebene JSON-Format zurück.
- UploadImage übergibt ein Bild einem PHP-Script, welches dieses auf dem Server speichert
- OnSwipeTouchListener: Detects Swipe-Events
- Settings: leere Klasse

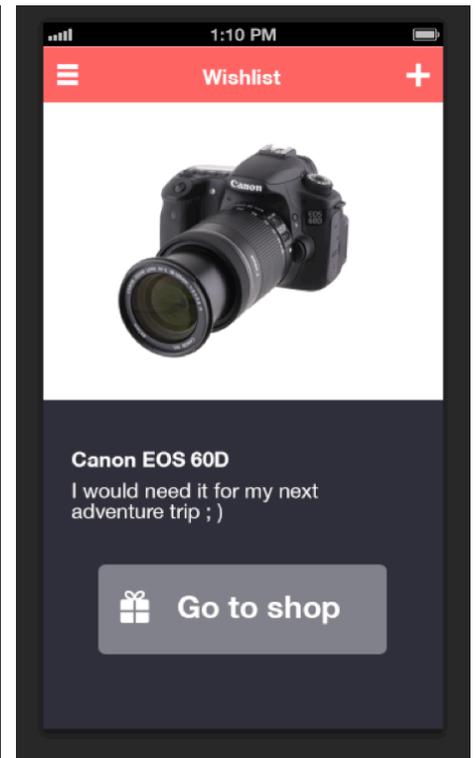
Storyboard



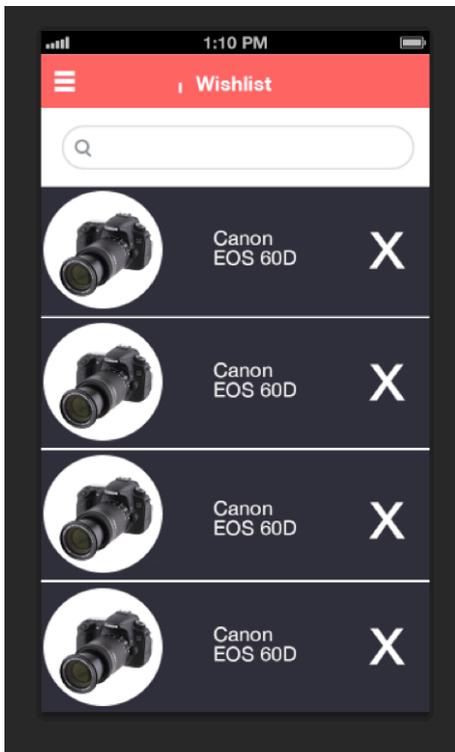
Login



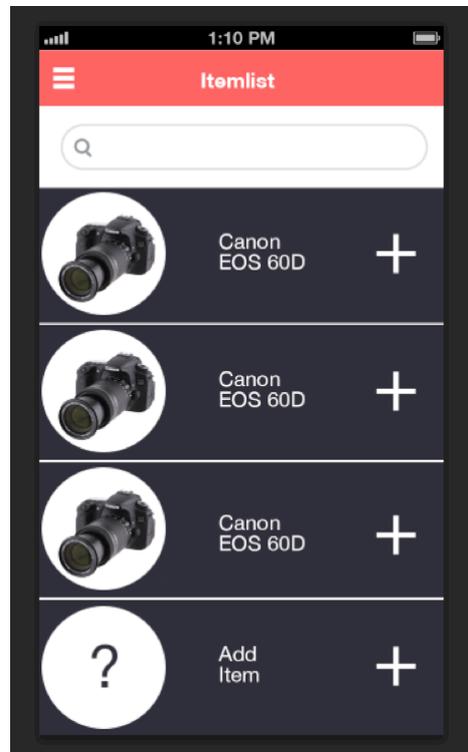
Menu



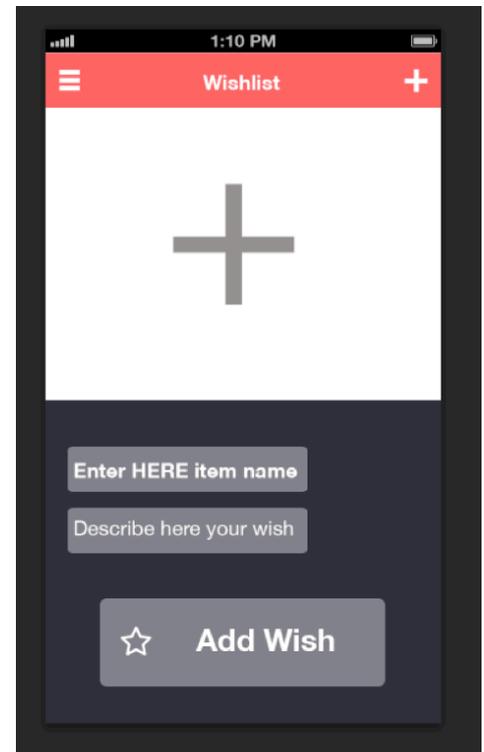
Detailansicht



Wunschliste



Persönlicher Wunsch
hinzufügen



Allgemeiner Wunsch
hinzufügen

Testcases

Nr.	Activity	Eingabe	Erwartetes Ergebnis	Tatsächliches Ergebnis	Ergebniss
1	Login	Benutzername, richtiges Passwort, klick auf Login	Man wird auf die Menu-Ansicht weitergeleitet.	Man wird auf die Menu-Activity weitergeleitet.	Bestanden
2	Login	Benutzername, falsches Passwort, klick auf Login	Es wird eine Fehlermeldung ausgegeben.	Es wird eine Fehlermeldung ausgegeben.	Bestanden
3	Login	klick auf „Create Account“	Ein Popup wird geöffnet, bei dem Userangaben eingegeben werden können.	Ein Popup wird geöffnet, bei dem Userangaben eingegeben werden können.	Bestanden
4	Login	Eingabe eines einmaligen Benutzername, klick auf „Create Account“	Login wird wieder angezeigt.	Login wird wieder angezeigt.	Bestanden
5	Login	Eingabe eines schon vergebenen Benutzername, klick auf „Create Account“	Es wird eine Fehlermeldung ausgegeben.	Es wird eine Fehlermeldung ausgegeben.	Bestanden
6	Menu	Klick auf Wishlist	Es wird auf die Wunschliste-Ansicht weitergeleitet.	Es wird auf die Wunschliste-Ansicht weitergeleitet.	Bestanden
7	Menu	Klick auf Logout	Es wird auf die Login-Ansicht weitergeleitet.	Es wird auf die Login-Ansicht weitergeleitet.	Bestanden
8	Wunschliste	Nichts	Es werden alle persönlichen Wunsch angezeigt.	Es werden alle persönlichen Wunsch angezeigt.	Bestanden
9	Wunschliste	Klick auf einen Wunsch.	Es wird auf die Detailansicht weitergeleitet.	Es wird auf die Detailansicht weitergeleitet.	Bestanden
10	Wunschliste	Klick auf Kreuz	Der Wunsch wird gelöscht.	Es wird auf die Detailansicht weitergeleitet.	Faild
11	Wunschliste	Eingabe in Suche	Nur übereinstimmende Wünsche werden angezeigt.	Nur übereinstimmende Wünsche werden angezeigt.	Bestanden
12	Detailansicht	Klick auf „Go To Shop“	Man wird auf die Webseite eines Shops weitergeleitet.	Nichts passiert.	Failed

Nr.	Activity	Eingabe	Erwartetes Ergebnis	Tatsächliches Ergebnis	Ergebniss
13	Pers. Wunsch hinzufügen	Eingabe in Suche	Nur übereinstimmende Wünsche werden angezeigt.	Nur übereinstimmende Wünsche werden angezeigt.	Bestanden
14	Pers. Wunsch hinzufügen	Klick auf Plus	Wunsch wird zu persönlichen Wünschen hinzugefügt.	Wunsch wird zu persönlichen Wünschen hinzugefügt.	Bestanden
15	Pers. Wunsch hinzufügen	Klick auf Plus bei Add Item	Man wird auf Allgemeiner Wunsch hinzugefügt weitergeleitet	Man wird auf Allgemeiner Wunsch hinzugefügt weitergeleitet	Bestanden
16	Pers. Wunsch hinzufügen	Nichts	Nur noch nicht gewünschte Wünsche werden angezeigt.	Nur noch nicht gewünschte Wünsche werden angezeigt.	Bestanden
17	Allg. Wunsch hinzufügen	Eingabe aller benötigten Angaben, Klick auf „Add Wish“	Wunsch wird als allg. Wunsch und pers. Wunsch hinzugefügt	Wunsch wird als allg. Wunsch und pers. Wunsch hinzugefügt	Bestanden

Reflexion

Wir haben die Grundanforderungen erreicht, jedoch konnten wir weniger Zusatzanforderungen erfüllen als wir zuvor gedacht haben. Grundsätzlich ist alles gut gelaufen und wir hatten nur ein grösseres Problem, nämlich die Datenbankanbindung zu einer externen MySQL-Datenbank. Dies wollten wir zuerst mit JDBC machen, jedoch konnten wir die Library nicht ins Android Studio einbinden und so mussten wir eine andere Lösung finden:

Die Datenbankabfragen werden von PHP-Scripts auf dem Server getätigt, die nötigen Attribute dafür werden mithilfe von POST übermittelt und die Query-Ergebnisse mithilfe des JSON-Formats zurückgegeben und vom Java-Code ausgewertet. Der grosse Nachteil ist, dass viele Sicherheitslücken bestehen. Zum Beispiel können die PHP-Scripts von jedem, der die URL kennt ausgeführt werden... Es gäbe Wege diese Lücken sicherer zu machen, dafür hatten wir jedoch keine Zeit mehr.

Projekt auf GitHub: <https://github.com/vilamonon/Whisha.git>