Projektdokumentation

im Fach

''mobile Datenbanken''

mit dem Thema

“Finanzplanung”

HTW Berlin

Gruppenmitglieder:

Francine Mzebaze	s0536816
Michael Fomenko	s0539589
Stebatien Rambauske	
Steve Terence Nguembou	s0536220

Dozent: Prof. Dr. Thomas Baar

Abgabe: 09.07.2015

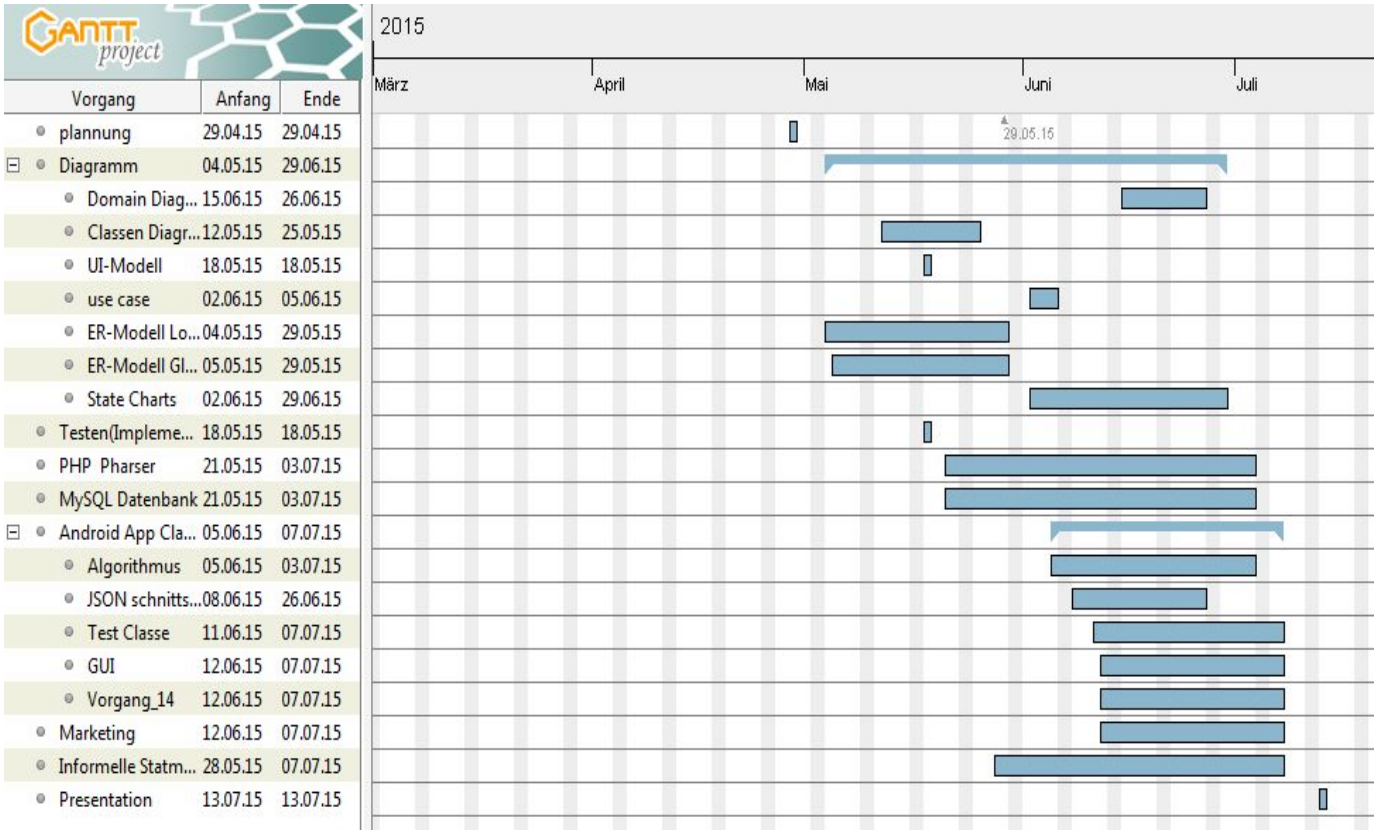
Inhaltsverzeichnis

1. Planung

Im Rahmen des Moduls “Embedded Mobile Datenbank” haben wir die Android Applikation “Finanzplanung” geplant, entwickelt und umgesetzt. Die Arbeit wurde so geplant, dass jede eine bestimmte Aufgabe erledigen mussten. Die Einteilung sieht folgendermaßen aus:

-Werbebotschaft, Planung, StateChart
Dieses Teil wurde von Francine Mzebaze gemacht.
- **Erstellung der GUI und Wireframes**
Diesen Part wurde von Nguembou Nana übernommen. Die Aufgabe bestand darin den Java Code zusammen mit dem (XML) Design zu entwickeln.
- Implementierung / Datenbank
Die Entwicklung einer Classe”dataAccess” um die Funktionen der Datenverarbeitung der GUI zu gewährleisenommen Fomenko übernommen. Es musste eine Datenbankmethode ausgesucht und eingebunden werden.
- PHP und Mysql Server einrichten
Diesen Schritt hat der Fomenko übernommen um den Zugriff der App auf eine externe Datenbank zu ermöglichen, es müssen PHP Scripte geschrieben werden um die Datenbankabfragen in JSON Format an die App zurück zu senden und den Zugriff auf den MySQL Server zu gewähren.
- Testen und Deployment
Die Aufgabe übernimmt Sebastien Rambauske. Die Funktionalitäten App müssen getesten werden.

Für die Programmierung des Hauptcodes haben wir die Entwicklungssoftware Android Studio genutzt. Desweiteren wurden andere Programme und Dienste wie Umlet, Pencil, PHP-Server, MySQL Server, GitHub, GimpShop und Windows Movie Maker verwendet.



Vorgang	Anfang	Ende
plannung	29.04.15	29.04.15
Diagramm	04.05.15	29.06.15
Domain Diag...	15.06.15	26.06.15
Classen Diagr...	12.05.15	25.05.15
UI-Modell	18.05.15	18.05.15
use case	02.06.15	05.06.15
ER-Modell Lo...	04.05.15	29.05.15
ER-Modell Gl...	05.05.15	29.05.15
State Charts	02.06.15	29.06.15
Testen(Impleme...	18.05.15	18.05.15
PHP Pharser	21.05.15	03.07.15
MySQL Datenbank	21.05.15	03.07.15
Android App Cla...	05.06.15	07.07.15
Algorithmus	05.06.15	03.07.15
JSON schnitts...	08.06.15	26.06.15
Test Classe	11.06.15	07.07.15
GUI	12.06.15	07.07.15
Vorgang_14	12.06.15	07.07.15
Marketing	12.06.15	07.07.15
Informelle Statm...	28.05.15	07.07.15
Presentation	13.07.15	13.07.15

Verfasser: Francine

2. Inception und Anforderungsanalyse

2.1 Kundennutzen

Erleichtert die Verwaltung der Finanzen durch das Festhalten der Ausgaben und das zentrale verteilen der Ausgaben an berechtigte. Darüberhinaus bietet diese App dem Kunden eine Übersicht der Ausgaben nach bestimmten Kriterien.

Kunden

- Verein
- WG
- Unternehmen
- Familie
- Privat

2.2 Werbebotschaft transportieren

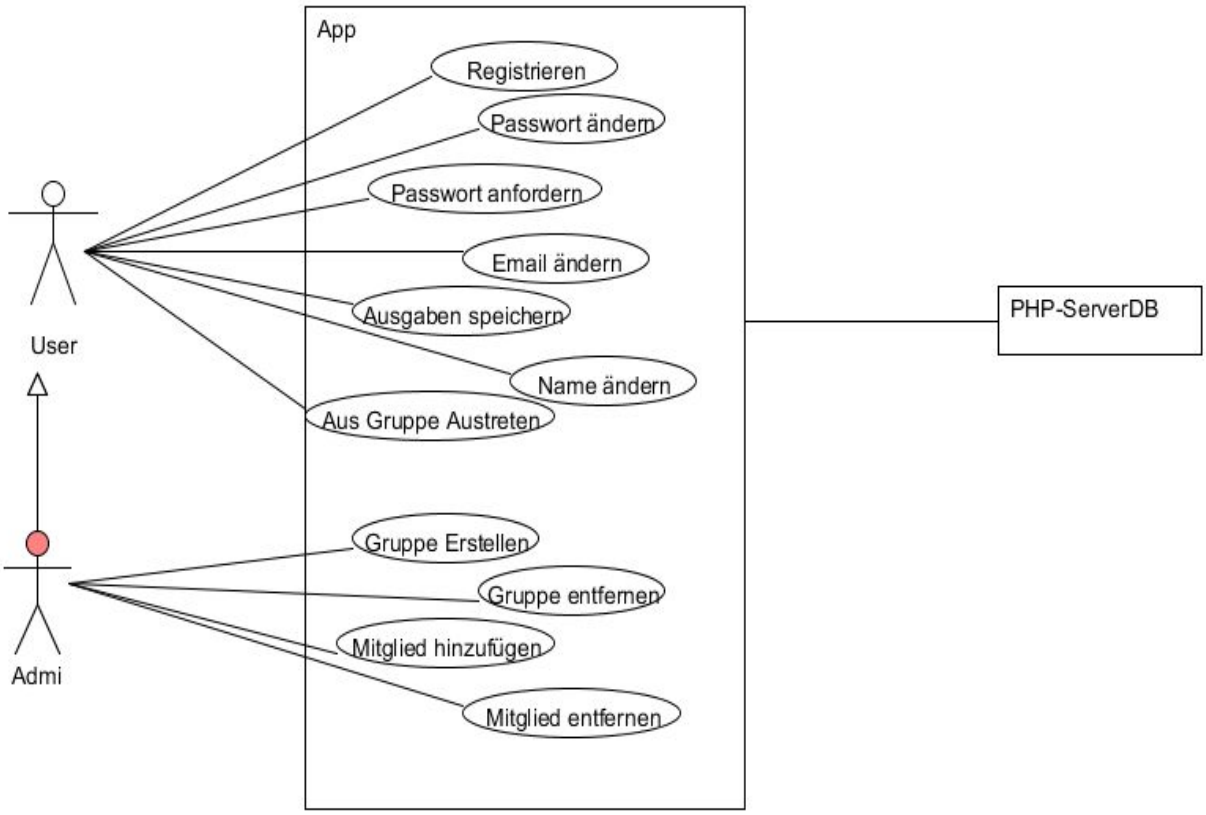
Kurzer Video-Clip/Trailer

Recording Video in Android 4.4
cd /Android/Sdk/platform-tools$
adb shell screenrecord --verbose ./sdcard/screencast-video.mp4

2.3 Grundfunktionalität und Oberfläche festlegen

Use Cases

Mit Hilfe von Use Cases werden alle möglichen Szenarien gebündelt, die eintreten können, wenn der User versucht, mit der App ein bestimmtes Ziel zu erreichen. Wie es hier zu sehen ist, haben wir zwei verschiedene User: der normale User und der Administrator(der jenige, der die Gruppe angelegt hat). Der Unterschied zwischen die beiden User besteht darin, dass der Administrator außer alles was der normale User machen kann, noch zusätzliche Sachen machen kann. Er kann zum Beispiel die Gruppe verwalten dh. Mitglieder hinzufügen oder entfernen oder die Gruppe löschen.

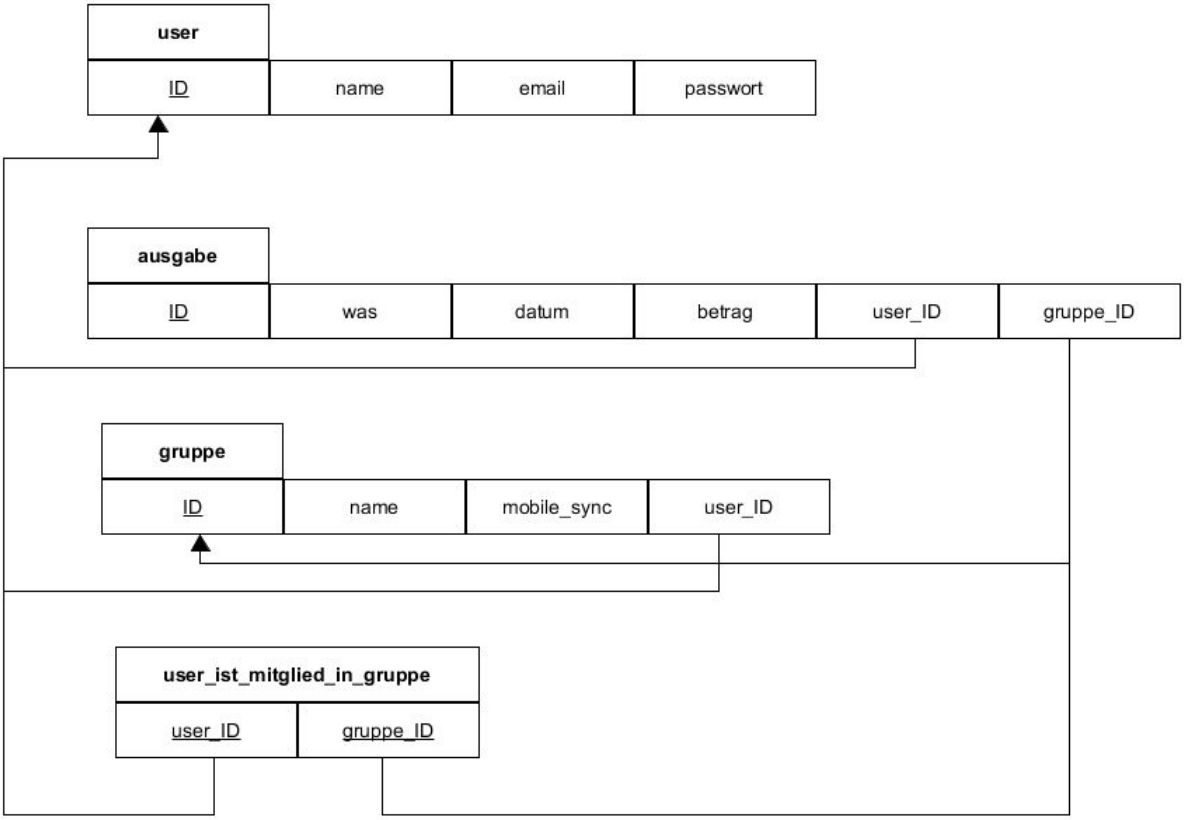


Verfasser: Fomenko, Steve

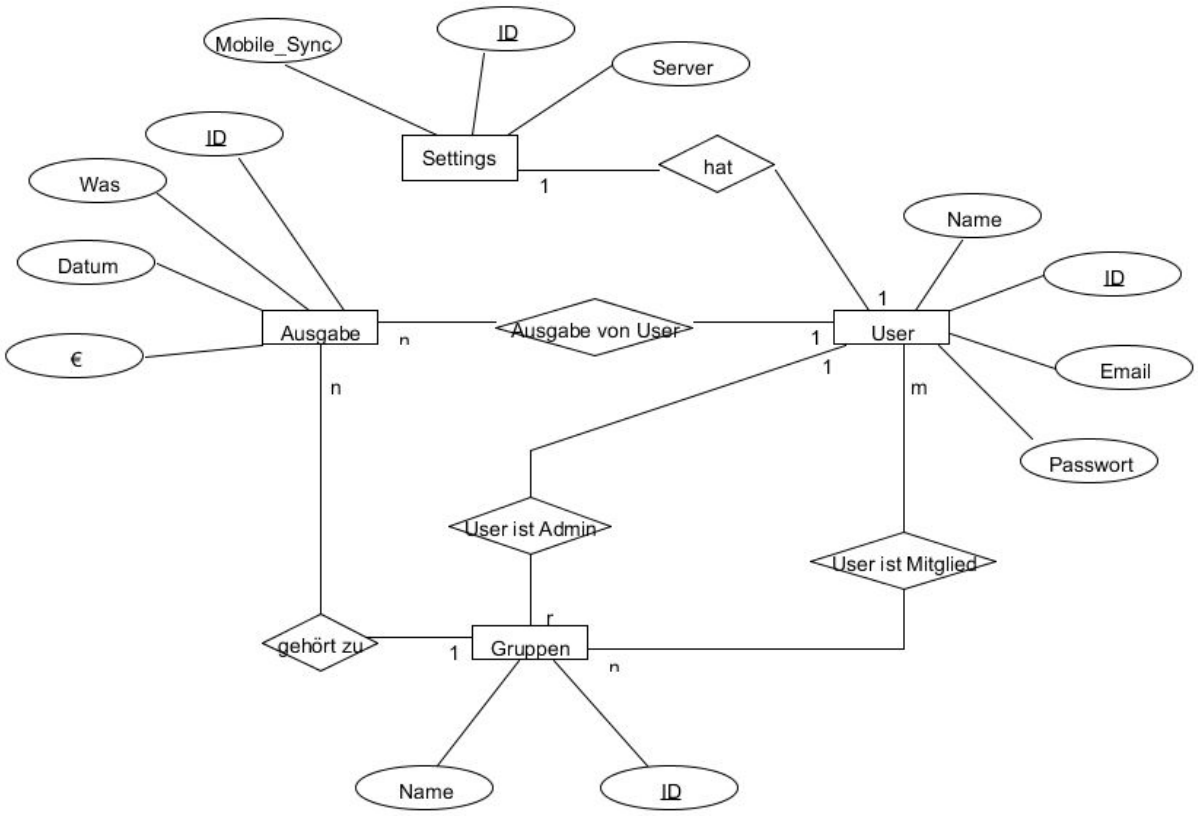
2.4 Datenbank Konzept

Neben dem Design wurde auch ein erstes Datenbankenmodell mit den erforderlichen Attributen erstellt.

Relationaler Datenbankerwurf

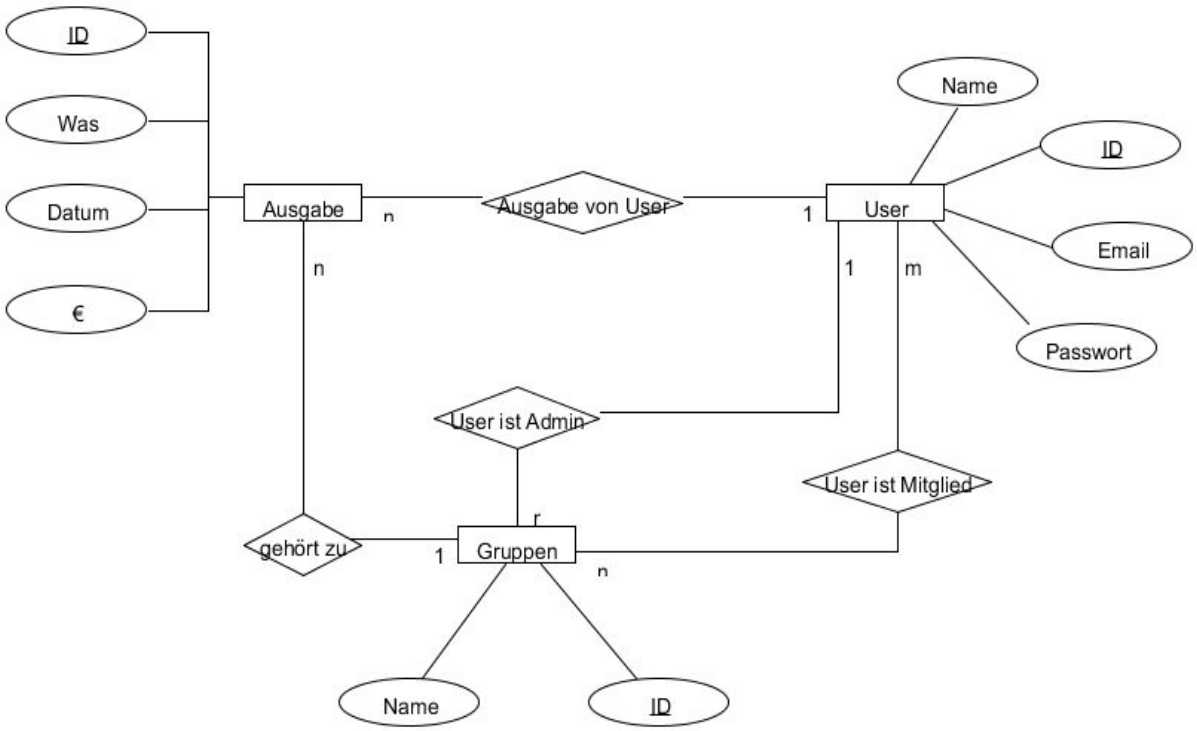


Konzeptioneller Datenbankentwurf Local



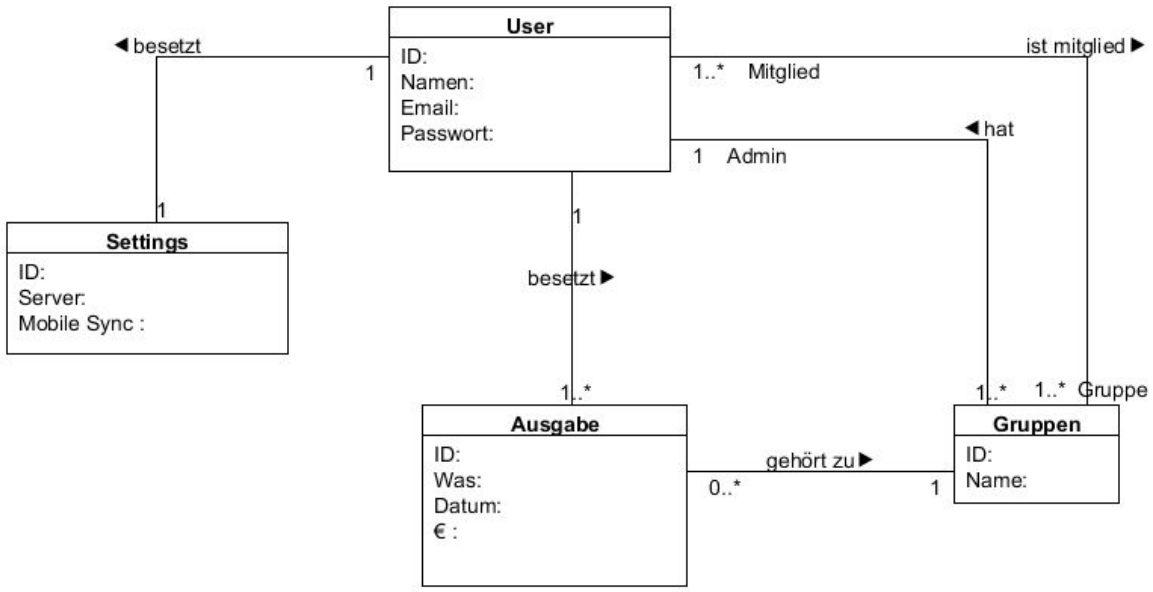
Verfasser: Fomenko, Francine

Konzeptioneller Datenbankentwurf Local



Verfasser: Fomenko, Steve

Datenbank-Schemata für lokale und zentrale DB



Verfasser: Francine

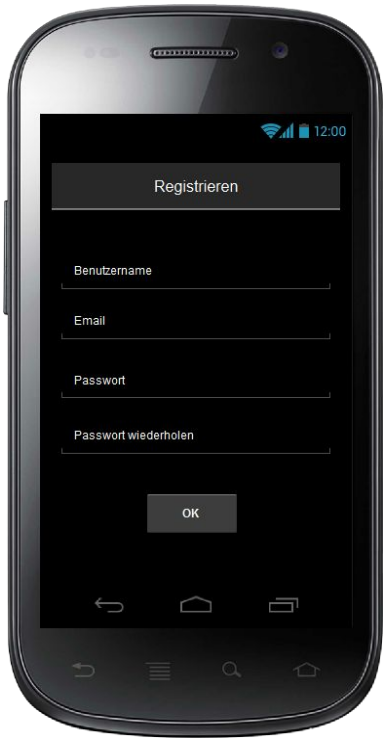
2.5 Wireframes

Um neben dem Konzept der Applikation auch ein Design zu geben, wurden mehrere Layouts(Wireframes) gefertigt, um einen ersten Eindruck über die Funktionalitäten und das Aussehen zu bekommen.

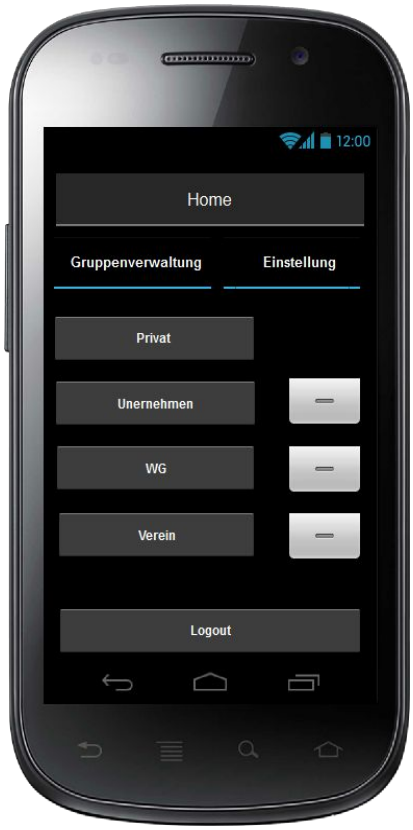
Hier haben wir unsere Startseite, die Loginseite. der User soll sich mit Benutzername und Passwort einloggen können. Falls er sein Passwort vergessen hat, gibt es unter Vergessen die Möglichkeit ein neues Passwort anzufordern.
Wenn er noch keine Zugangsdaten hat, dann hat er die Möglichkeit sich zu registrieren.

Geprüft soll werden:

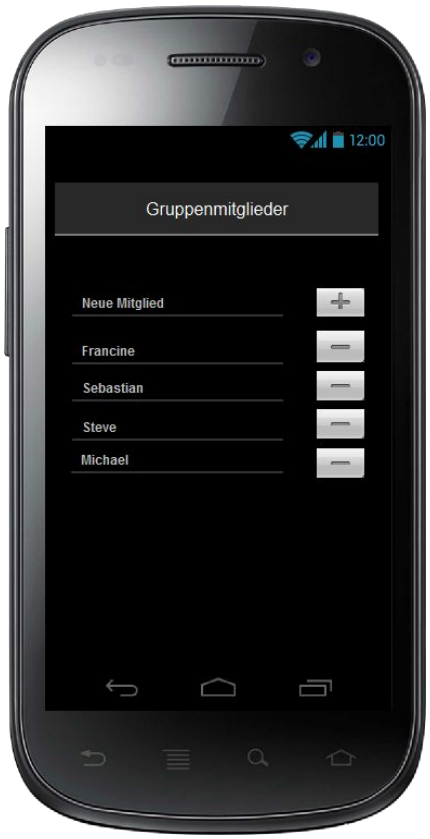
- ob alle Felder ausgefüllt wurden.
- ob Nutzerdaten übereinstimmen



Möchte ein User sich registrieren, so muss er einen Benutzernamen, ein Passwort und eine E-Mail Adresse hinterlegen. Es muss geprüft werden ob alle Felder ausgefüllt sind und ob der Benutzername schon existiert.

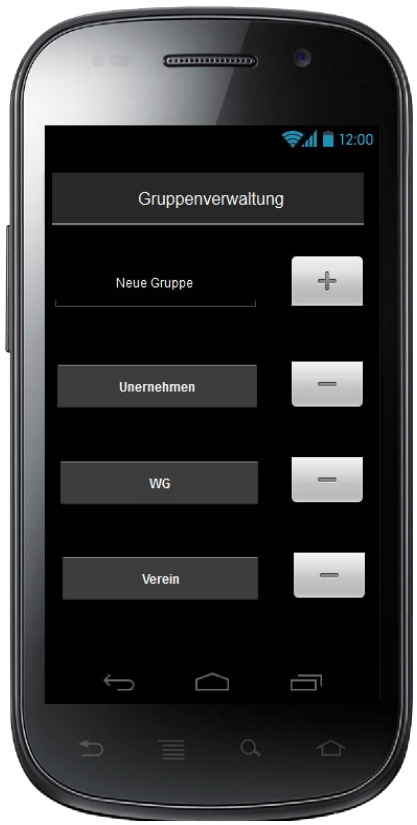


Hat sich der User erfolgreich eingeloggt, gelangt er zur **Home.** Hier hat er die Möglichkeit, sowohl Gruppen zu verwalten als auch Einstellung zu ändern. Außerdem kann er sich auch ausloggen.



Auf diesem Fenster können Mitglieder in eine Gruppe hinzugefügt werden. Diese können auch aus der Gruppe entfernt werden. Es soll geprüft werden, wenn der User ein neues Mitglied hinzufügen will, das Feld **Neue Mitglied** nicht leer ist.

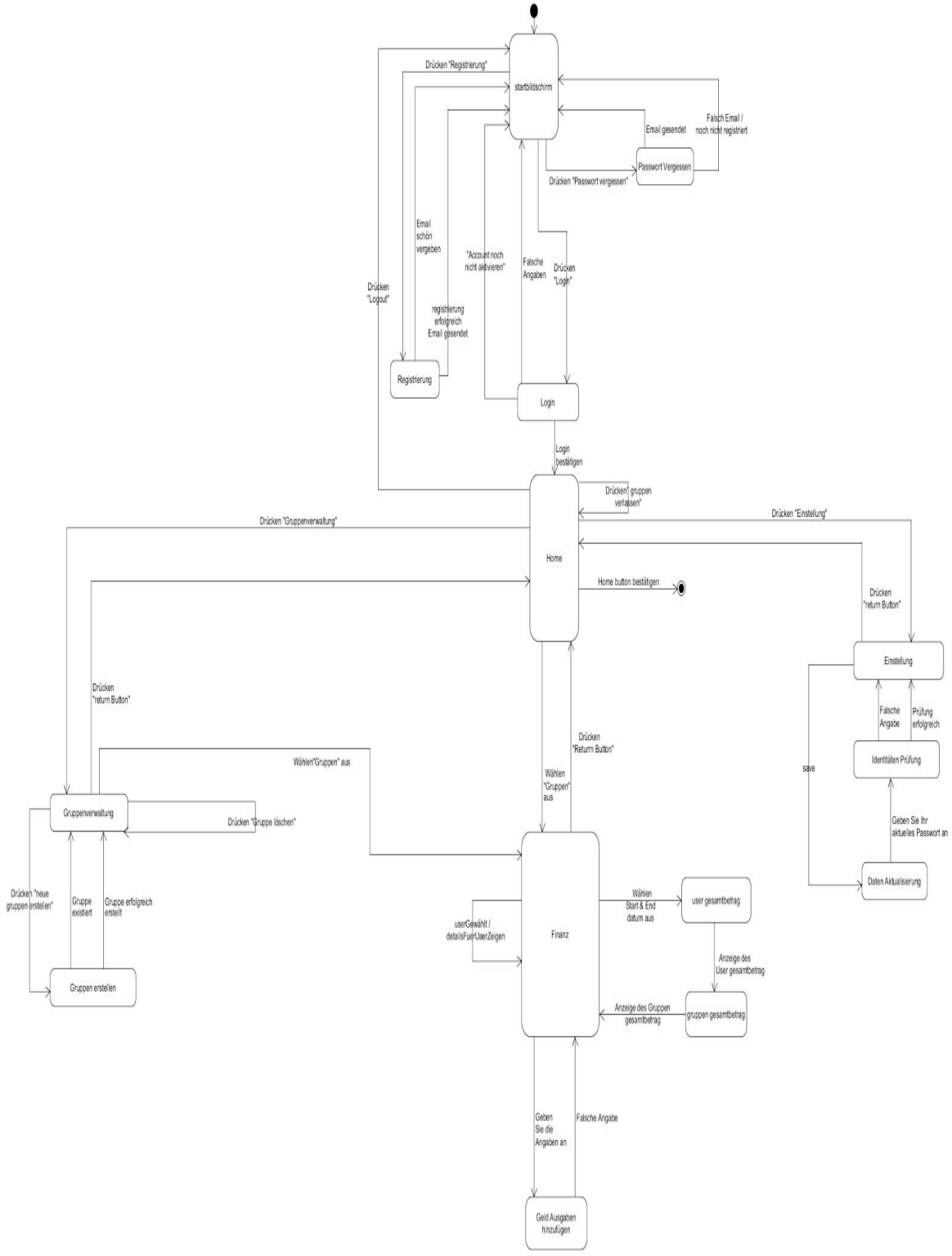
Hier hat der User bzw. der Administrator der Gruppe die Ansicht auf aller Transaktionen, die von Mitglieder gemacht wurden. Er kann sich es von einzelnen Mitglieder anzeigen lassen. Außerdem kann er die Ausgaben einer bestimmten Zeitraum anzeigen lassen.



Bei der **Gruppenverwaltung** geht es um das Hinzufügen einer neuer Gruppe oder eine gruppe zu entfernen.

Verfasser: Fomenko, Steve

Statechart



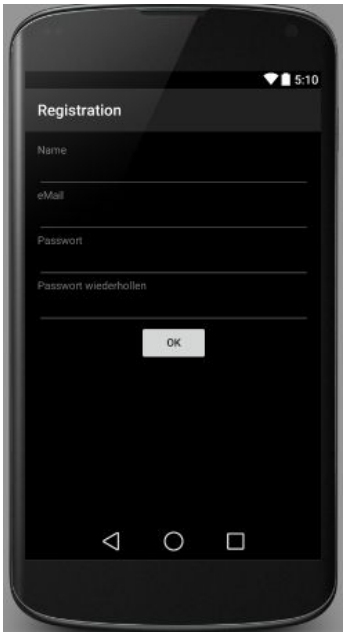
Verfasser: Fomenko, Francine

3. Implementierung

Nach beenden der Konzeptphase hat sich jeder an seine Aufgabenbereiche gemacht um das Konzept so gut wie möglich zu realisieren.

Design und Funktionalität:

Bevor mit der Hauptsachlichen Programmierung begonnen wurde, wurden die verschiedenen XML Dateien in Android-Studio erstellt, Farben angepasst und später parallel zum Code Anpassungen vorgenommen.

Activity_Login.xml	Activity_registration.xml
	

Verfasser: Fomenko, Steve

activity_home.xml	Activity_gruppenverwaltung.xml
	

Verfasser: Fomenko, Steve

Die restlichen XML- Datei können Sie im Code sehen. Die oben dargestellten XML-Dateien sind nur einigen Beispielen.

Ohne Ansteuerung der XML Layouts macht das Projekt wenig Sinn, deshalb wurden folgende Java Dateien (bzw. Klassen) erstellt.

XML-Dateien	**Java Dateien(bzw. Klassen)**
Activity_Login.xml	ActivityLogin.java
Activity_registration.xml	ActivityRegistration.java
activity_home.xml	ActivityHome.java
Activity_gruppenverwaltung.xml	ActivityGruppenverwaltung.java
activity_gruppenmitglieder.xml	ActivityGruppenmitglieder.java
activity_finanzen.xml	ActivityFinanzen.java

activity_settings.xml	ActivitySettings.java

Verfasser: Fomenko,Steve

Daten Classen

Classe	**Funktion**
Gruppe.java	Speichert den Datentypen Gruppe ab.
Mitglied.java	Speichert den Datentypen Mitglied ab.
Geldausgabe.java	Speichert den Datentypen Ausgaben ab.
ExpandebleListAdapter.java	Wird benötigt um eine ausfahrbare Liste auf der GUI zu erstellen .
Data_Access.java	Wird benötigt um auf die Nutzerdaten Zugriff zu bekommen.
ServiceHandler.java	Wird benötigt um eine http Verbindung zum PHP-Server zu erstellen.

Verfasser: Fomenko

Data_Access Classe

Rückgabewert	**Funktion**
int	addGeldausgabe(String datum,String was,Float betrag,Integer gruppen_id)
int	addGruppe(String gruppenname)
void	addGruppenMitglied(Integer gruppen_id, Integer mitglied_id)
String	addGruppenMitglied(Integer gruppen_id, String email)
int	addUser(Integer mitglied_id, String name, String email)
int	addUser(Integer user_id, String name, String email, String passwort)

void	databaseDelete(Context context)
int	deleteGruppe(Integer gruppen_id)
int	deleteMitglied(Integer gruppen_id, Integer mitglied_id)
Integer	existUser(String email)
Boolean	existUserInGruppe(String email, Integer gruppen_id)
ArrayList<Gruppe>	getGruppen()
Float	getGruppenGesamtbetrag(String startdatum, String enddatum, Integer gruppen_id)
Integer	getGruppenMasterID(Integer gruppen_id)
ArrayList<Mitglied>	getGruppenMitglieder(Integer gruppe_id)
Integer	getLoginState()
List<Gruppe>	getMeineGruppen()
Boolean	getMobileSyncStatus()
String	getName()
String	getServer()
List<Geldausgabe>	getUserGeldausgaben(String startdatum, String enddatum, Integer gruppen_id, Integer user_id)
Float	getUserGesamtbetrag(String startdatum, String enddatum, Integer gruppen_id, Integer user_id)
String	Login(String email, String passwort)
Boolean	LoginLocal(String email, String passwort)
void	Logout()
void	onCreate(SQLiteDatabase db)
void	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

String	registration(String name, String email, String passwort, String passwortValidation)
String	sendPasswortToEmail(String email)
void	setLoginState(Integer user_id)
void	setMobileSync(Boolean mobileSync, String passwort)
int	setNewName(String newName, String passwort)
int	setNewPasswort(String newpasswort, String oldpasswort)
int	setNewServer(String newServer, String passwort)
String	stripHtml(String html)
String	stripHtmlForArray(String html)
Boolean	validation(String passwort)
void	verlasseGruppe(Integer gruppen_id)

Verfasser: Fomenko

Datenbank

Die Lokale Datenbank wird als Puffer genutzt um die Zeiten ohne Internet die Werte des Users zu Speichern. Um die lokalen Daten richtig zu synchronisieren gibt es in jeder Tabelle der lokalen DB eine “status” Spalte:

Wert der Spalte “Status”	**Bedeutung**
d	Die Zeile sollte aus der Datenbank gelöscht werden
u	Die Zeile ist noch nicht synchronisiert mit der externen Datenbank
	es ist alles OK

4. Testen und Deployment

Beim Erstellen der Funktionen wurden die erstellten Funktionen auf deren Funktionalität überprüft und dabei die möglichen Fehler die durch falsche Bedienung der App entstehen könnten durch Überprüfung der eingaben, abgefangen.

6. Quellen

Youtube Tuttorials

Android App Programieren

https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
https://www.youtube.com/playlist?list=PL13I0cBsOJUc0OTIv09DG7_m2T0Si0Wbx

Android Studio Video Serie

https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E

sql Lite Tutorial Serie

https://www.youtube.com/playlist?list=PLS1QulWo1RIaRdy16cOzBO5Jr6kEagA07

Versionsverwaltungssystem Git in Eclipse

https://www.youtube.com/watch?v=r5C6yXNaSGo

Synfig Studio

https://www.youtube.com/watch?v=Q-Mf9hbft8E

SQLite

http://www.sqlite.org/datatype3.html
https://www.sqlite.org/foreignkeys.html
http://stackoverflow.com/questions/1942586/comparison-of-database-column-types-in-mysql-postgresql-and-sqlite-cross-map
http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/
http://tips.androidhive.info/2013/10/android-insert-datetime-value-in-sqlite-database/