HCI Meilenstein 3

Dursun, Camkerten a0027244@unet.univie.ac.at, Pektas Tarik

a1325165@unet.univie.ac.at, Bozkurt Yigit Berkay

a1029659@unet.univie.ac.at, und Ayyildiz Mert Ahmet

a1125172@unet.univie.ac.at

Universität Wien / HCI

SS16 / Gruppe 3

**3 Prototyp Beschreibung**

3.1 Side Panel

Eine Ausstattung unserer Finanzapplikation mit einem Side Panel wurde während der Entwicklung des low fidelity Prototyps bei den Testern durch Interviews sehr gut bewertet. Daher entschieden wir uns für die Implementierung dieses modernen Features, der dazu beiträgt, schnell zwischen verschieden Seiten zu wechseln und dabei die Übersicht zu behalten. Damit erreichen wir eine erhöhter Benutzerfreundlichkeit unserer Applikation. Folgende Seiten sind aus Side Panel erreichbar:

* Einnahmen (Income)
* Ausgaben (Expense)
* Sparziel (Saving Target)
* Berichte (Reports)
* Profile (Profiles)

3.2 Startseite der Applikation / Expense

Auf der Startseite unserer Finanz-Applikation haben wir die Funktion „Ausgabe“ implementiert, die sich während der Evaluierung unseres Prototyps als wichtigste und am öftesten zum Einsatz kommende Funktion herausgefiltert hat.

Folgende Eingabefelder sind auf der Ausgaben-Seite möglich:

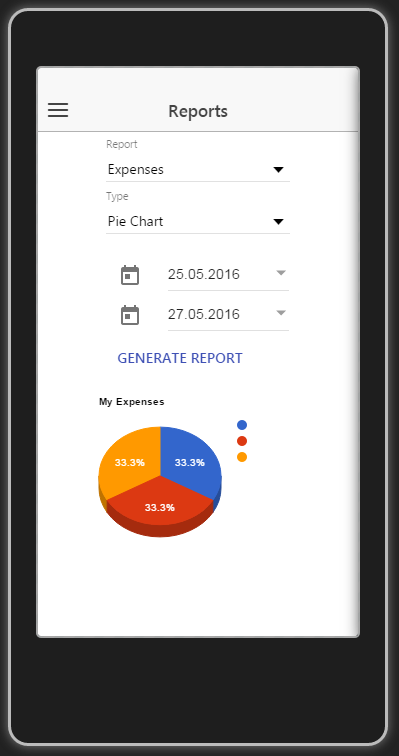
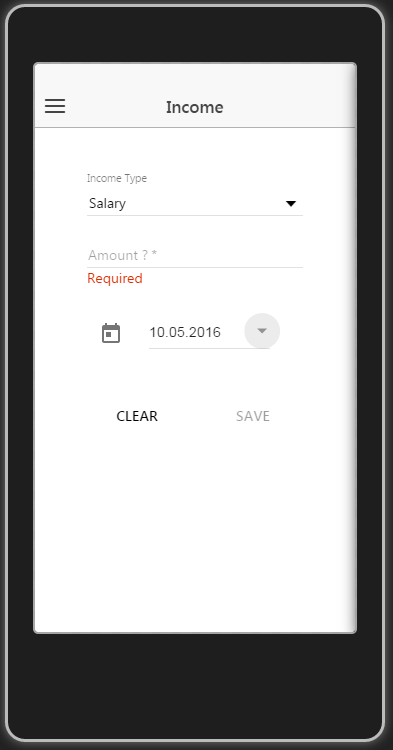
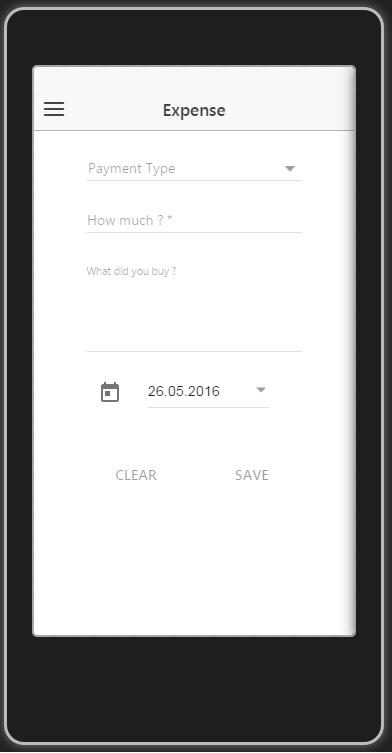
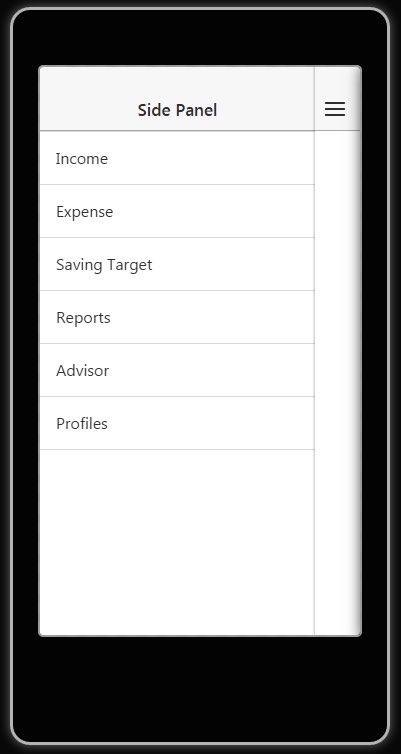
* Payment Type: In der Drop-Down-Liste dieses Feldes befinden sich die unterschiedlichen Bezahl-Möglichkeiten wie Kreditkarte, Bankomat und Bar.
* How much?: Eingabe von Ausgabenhöhe, das als Pflichtfeld eine Eingabe erfordert. Fehlen dieser Eingabe verhindert das Speichern der Daten.
* What did you buy?: Hier gibt man den Grund der Ausgabe ein.  
    
  Anschließend wählt man das Ausgabendatum durch Selektion aus dem Kalenderfeld ein und speichert die Daten mit einem Klick auf das Aktionsbutton „Save“ oder man löscht alle Eingaben mit „Clear“ und kann die Eingabe erneut starten. Die Selektion des Kalenderfeldes auf diese von uns implementierte Art und Weise war das Ergebnis unserer Entscheidung, die wir während unserer Interview festgestellt haben. Das trägt zu einem besseren Usability bei.

3.3. Reports

Eine weitere Funktion ist der Berichtfunktion. Wie auch in der Abbildung gezeigt, hat man die Möglichkeit Ausgaben durch Auswahl eines gewünschten Diagramms/einer Tabelle und mit der Festlegung der Zeitfenster (Anfang- und Enddatum) zu visualisieren. Folgende Berichtstypen unter Drop-Down Liste stehen zur Verfügung: Table, Pie Chart, Area Chart, Bar Chart und Line Chart.

3.4 Income

Eingabe von Einkommen wurde genau so wie die Eingabe von Ausgaben möglichst einfach und verständlich gehalten, damit die Benutzer bei der Verwendung der Applikation ohne große Mühe und wenig Zeiteinsetzung gewünschte Interaktionen führen können. Auf der Seite vom „Income“ ist die Wahl des Einkommens unter Drop-Down-Liste der Einkommenstyp möglich. Einkommen wurde als Gehalt/Lohn, Forderungen und andere untergliedert. Einkommenshöhe ist ein erforderliches um die eingegebene Daten anschließend zu speichern. Das Datum kann wie bei der Aufgabenfunktion aus dem Kalenderfeld ausgewählt werden.



Technische Designentscheidungen

Nachdem wir auch einige andere Frameworks getestet haben, darunter JQuery Mobile, entschieden wir uns letztendlich uns für AngularJS und dem Ionic Framework. Obwohl unserer Meinung nach auch Ionic als Mobile Framework auch nicht gerade optimal ist, wird das dennoch durch die sehr gute Struktur und Verwendbarkeit von AngularJs ausgeglichen. Dank AngularJS konnten wir auch die Funktionalitäten und das Design nochmals mit dem MaterialJs Framwork erweitern, welches uns insbesondere bei den Formerstellungen eine große Hilfe. Bezüglich des Designs haben wir auch Bootstrap verwendet, mussten aber aufgrund von Inkompatibilität, es wieder rausnehmen. Für den Reports Bereich haben wir uns, nach einiger Recherche, für GoogleCharts entschieden und können dies auch besten Gewissens weiterempfehlen, da es unserer Meinung nach sehr leicht integrierbar ist und zugleich auch sehr detaillierte Ergebnisse liefert. Für die Speicherung der Daten würde uns eine local storage (oder auch indexeddb) nicht ausreichen. Deshalb haben wir uns für eine SQLLite Datenbank entschieden die auch bei komplexeren Abfragen unsere App abdecken wird.