

Übungsblatt 6

Abgabe und Präsentation

Abgabe: Schicken Sie das technische Konzept sowie den finalen Prototypen bis zum 12.12.2012 (gerne auch schon früher) als PDF an steinberger@cip.ifi.lmu.de (Betreff: "PEM Gruppe X - Technisches Konzept").

Präsentation: Präsentieren Sie Ihre Ergebnisse (Evaluierung, Prototyp, technisches Konzept, aktueller Stand der Implementierung) am 19.12.2012 im Kurs. Pro Team stehen ca. 10 Minuten für die Präsentation und ca. 10 Minuten zum Diskutieren zur Verfügung.

Tipps & Tricks

Tragen Sie sich in die Tipps & Tricks Tabelle (<http://goo.gl/P0A8r>) ein, falls Sie noch nicht dran waren.

Evaluierung

a) Recherche: Hinterfragen Sie alle UI-Elemente und Input-Mechanismen Ihres ersten Prototypen und prüfen Sie, ob Sie evtl. besser geeignete Komponenten finden. Gehen Sie dabei alle UI-Elemente in XCode durch und recherchieren Sie weitere Libraries, wie z.B. [Cocoa Controls](#) und [iOS Mix](#). Überarbeiten Sie darauf basierend den Prototyp.

b) Heuristische Evaluierung: Evaluieren Sie das User Interface Ihres Prototypen anhand von [Heuristiken](#). Erstellen Sie hierzu zunächst eine Liste von Heuristiken (z.B. basierend auf [Nielsen's 10 Heuristics](#), den [Apple Human Interface Guidelines](#), [Principles of User Interface Design](#) und [UX Magazine](#)). Achten Sie darauf, dass Ihre Liste nicht nur Aspekte abdeckt, auf die Sie sowieso schon viel Wert gelegt haben (der Zweck der Evaluierung ist, Schwachstellen zu finden). Führen Sie dann unabhängig voneinander die Evaluierung durch, tragen Sie die Ergebnisse zusammen und erstellen Sie eine Liste mit Usability Problemen. Bewerten Sie die Usability Probleme nach Wichtigkeit (siehe [Severity Rating](#)) und überarbeiten Sie darauf basierend das User Interface.

c) Nutzerstudie: Evaluieren Sie den überarbeiteten Prototypen, indem Sie eine Nutzerstudie durchführen. Legen Sie dazu typische Aufgaben fest, die der Nutzer mit der App machen wird. Überlegen Sie sich, ob es Sinn macht, einen interaktiven Prototyp zu verwenden (Hinweis: mit [Pop](#) können aus Papierprototypen klickbare iOS Prototypen erstellt werden). Führen Sie eine [Think Aloud Study](#) mit drei Leuten durch, die der Zielgruppe entsprechen (möglichst keine PEM-Studenten). Achten Sie dabei besonders darauf, was der Nutzer vom Screen bzw. von UI-Elementen erwartet und wo er nicht weiter weiß. (Falls das Ergebnis der Studie "Die Teilnehmer finden die App eigentlich intuitiv" ist, wurde die Studie nicht gut genug durchgeführt. ;-))

Finalisierung des Konzepts

a) User Interface: Finalisieren Sie Ihren Prototypen basierend auf Recherche, heuristischer Evaluierung und Nutzerstudie.

b) Technisches Konzept: Legen Sie fest, welche Features Sie auf jeden Fall implementieren und welche Sie weglassen, falls am Ende des Semesters nicht genügend Zeit bleibt. Geben Sie an, welche APIs Sie verwenden werden. Überlegen Sie, inwieweit Sie Server-Kommunikation, persistente Datenspeicherung, echte Daten und andere Komponenten, die nichts mit dem User Interface zu tun haben, implementieren werden. Recherchieren Sie hierfür auch:

- **Core Data:** Persistente Datenspeicherung von Objective-C. Hier können Klassen, Attribute und Beziehungen ähnlich wie bei Storyboards in einem Visual Editor in XCode erstellt werden. Die Dateien für die Klassen können darauf basierend generiert werden. Siehe [iOS Data Management](#) (Übersicht), [Core Data Introduction](#) (kurze Einführung und Tutorial), [Ray Wenderlich Tutorial](#) (ausführliches Tutorial).
- **Parse:** Erledigt die Serverseite von vielen Apps, u.a. Push Notifications, Data in the Cloud, User Management, Facebook-Login.
- **Ray Wenderlich Tutorials:** Übersicht über viele iOS Technologien mit ausführlichen Tutorials. Bietet sich an, um APIs kennenzulernen und den Aufwand für PEM abzuschätzen.

Start des Software Prototypen

a) Repository: Legen Sie ein SVN oder Git Repository an, so dass alle Teammitglieder am Projekt arbeiten können. XCode unterstützt sowohl SVN als auch Git (siehe [Dokumentation](#)). Falls Sie Git verwenden, können Sie Ihr Repository z.B. bei [GitHub](#) erstellen (siehe [Ray Wenderlich Tutorial](#)). Falls Sie SVN verwenden, können Sie mit Ihrem cip-Account ein Repository anlegen (siehe [Dokumentation der Rechnerbetriebsgruppe](#)). Dabei können Sie wie folgt vorgehen:

- VPN starten, dann verbinden (in der Konsole): `ssh cipname@remote.cip.ifi.lmu.de`
- Repository anlegen (in der Konsole): `mk_svn_repo.sh pem_gruppeX`
- Repository für Teammitglieder freigeben: Im cip-Space die Datei `~/public_svn/auth` im Texteditor öffnen und alle cip-Namen mit Schreibrechten eintragen (z.B. `mustermann = rw`). (Zugriff auf den cip-Space z.B. über den FTP-Client [Cyberduck](#).)
- XCode-Projekt erstellen und z.B. auf dem Desktop speichern.
- Neuen Ordner anlegen (z.B. auf dem Desktop) und 3 Unterordner erstellen: `branches`, `tags`, `trunk`. Das XCode-Projekt in den `trunk`-Ordner schieben.
- Den Ordner (mit seinen 3 Unterordnern) ins SVN-Repository importieren, z.B. mit dem SVN-Client [Versions](#): `File > Import` (oder über die Konsole). Das XCode-Projekt ist jetzt im SVN-Repository. Der lokale Ordner (mit seinen 3 Unterordnern) kann jetzt gelöscht werden.
- Das XCode-Projekt aus dem SVN-Repository auschecken (z.B. mit [Versions](#) oder über die Konsole), um damit lokal zu arbeiten. Das ausgecheckte XCode-Projekt öffnen. Commits und Updates können direkt über XCode gemacht werden.

b) Implementierung: Fangen Sie mit der Implementierung der App an.