

Aufgabe 2C: App-Entwicklung

Erstellung einer servicebasierten mobilen Android

Das Fotoportal *PhotonPainter* setzt vermehrt auf mobilen Zugriff. Um den von unterwegs zugreifenden Nutzern ein besonderes Erlebnis zu gewähren, soll an mobile Endgeräte *PhotonApp*, eine Anwendung für Android ausgeliefert werden.

Ihre Aufgabe ist es, diese App zu erstellen. Kernfunktion der Anwendung soll es sein, mit dem mobilen Endgerät gemachte Bilder direkt zu übermitteln sowie die eigenen Bilder zu verwalten. Außerdem soll eine Slideshow aus den eigenen Bildern angezeigt werden können. Da es eine Single-User-Anwendung ist, soll sie auch die Benutzerdaten speichern und sich automatisch am Server anmelden. Eine letzte Funktion ist der PhotoGuide. Ist dieser aktiviert, zeigt *PhotonApp* Bilder an, die in der Nähe des aktuellen Standortes aufgenommen wurden, um den Benutzer auf potentielle Motive hinzuweisen und zu inspirieren.

Das Ergebnis muss folgende Anforderungen erfüllen:

- Die Slideshow soll automatisch aus allen Fotos des Benutzers zusammengestellt werden.
- Wegen der Vielfältigkeit von Android Geräten, soll die Anwendung Auflösungen von 480*320 und 800*480 unterstützen.
- Ist keine Verbindung mit dem *PhotonEmitter*-Server möglich, muss dem Benutzer das entsprechend mitgeteilt werden bzw. die App entsprechend reagieren.
- Um Fotos für den PhotoGuide auszuwählen, nutzen Sie einen Abgleich der GPS-Koordinaten mit dem aktuellen Standort des Nutzers. Für eine besonders gute Lösung können Sie zudem Blickrichtungsinformationen einbringen oder die Darstellung der Fotos auf einer Karte implementieren.
- Die Lösung soll ab Android 2.1 (API-Level 7) und somit auf den meisten Geräten lauffähig sein.
- *Alle verwendeten Bestandteile müssen entweder Public Domain sein oder die erforderlichen Nutzungsrechte müssen schriftlich vorliegen.*

Abzugeben sind:

- gut dokumentiertes Quellprojekt (Eclipse)
- Lauffähige .apk. Falls Sie die Maps API verwenden, signieren Sie bitte die Anwendung

Tipps:

- Beginnen Sie zunächst damit, die Verbindung zum *PhotonEmitter*-Backend zu programmieren. Ist eine Kommunikation möglich, können Sie die restlichen Funktionen umsetzen. Entwickeln Sie zuletzt den *PhotoGuide*, dieser ist die umfangreichste Funktion.
- Beachten Sie, dass Anwendungen in der Dalvik VM nur sehr wenig Arbeitsspeicher zu Verfügung steht. Skalieren Sie deshalb, wenn nötig, Bilder beim Laden von der Speicherkarte und helfen Sie dem Garbage Collector nicht benötigte Bitmaps aus dem Speicher zu räumen.
- Damit die Benutzeroberfläche während Up- und Downloads nicht blockiert wird, sollten diese Aktionen als asynchrone Tasks implementiert werden. Sollte es doch nötig sein, das UI zu blockieren, muss der Benutzer mit *ProgressDialogs* über den Status informiert werden.
- Achten Sie darauf, dass beim Pausieren oder Stoppen einer *Activity* alle Hintergrundprozesse beendet werden, sodass die Anwendung in diesem Zustand die CPU nicht belastet.