

Entwicklung mobiler Applikationen

WS2022/2023

Prüfungsleistung

Jan Brodhaecker | 2022

Prüfungsleistung

Präsentation

- formelle Kriterien
 - ab sofort (14.11.2022), bis spätestens 14./15. Dezember 2022
 - ca. 15-20 Minuten
 - Folien als Zusammenfassung
 - kein extra Handout

Prüfungsleistung

Präsentation

- Inhalt
 - Vorstellung des Programmierentwurfs
 - Technologie
 - Funktionalität bzw. geplante Features
 - Diskussion eines konkreten Problems während der Entwicklung (inkl. Quellcode)
 - bspw. nativ vs. hybrid vs. cross-platform, Anbindung einer bestimmten Bibliothek, Zugriff auf bestimmte Funktionalität, Bugs, etc.

Organisatorisches

Programmierentwurf

- formelle Kriterien (1/2)
 - Abgabe bis 31.01.2023
 - Abgabe per public/private github.com Repository
 - falls private, `janbrodhaecker` als collaborator hinzufügen
 - der letzte Commit vor dem Abgabedatum wird berücksichtigt
 - Verhältnis der Contributor beachten (bspw. ein Commit auf den `main`-Branch einen Tag vor Abgabe wäre eher ungünstig)

Organisatorisches

Programmierentwurf

- formelle Kriterien (2/2)
 - **solte das Projekt nicht kompilieren/ausführbar sein, führt dies zu einer 5,0**
 - evtl. Anweisungen zur Ausführung (bspw. API-Key, etc.) sollten in einer Readme Datei innerhalb des Repositories vermerkt werden

Organisatorisches

Programmierentwurf

- inhaltliche Kriterien (1/5)
 - der Funktionsumfang des Programmierentwurfs soll vollständig sein
 - bspw. Features die zum Absturz führen oder nicht vollständig sind, sollten nicht angeboten werden
 - der Programmierentwurf soll stabil sein
 - bspw. keine Berechtigungen erteilt, Web-Request ohne Internet-Verbindung, leerer Input, Orientierung, etc.

Organisatorisches

Programmierentwurf

- inhaltliche Kriterien (2/5)
 - die Funktionalität des Programmierentwurfs soll effizient implementiert werden
 - bspw. Verbindungen abbauen sobald die Applikation in den Hintergrund geht, keine unnötigen/duplizierten Web-Requests oder sonstige Aufrufe
 - Lifecycle-Events sind zu beachten
 - bspw. Daten in einem Formular sollten zwischengespeichert werden, sobald die Applikation in den Hintergrund geht

Organisatorisches

Programmierentwurf

- inhaltliche Kriterien (3/5)
 - der Programmierentwurf sollte ein remote bzw. ein lokales API anbinden oder andere 3rd-Party-Bibliotheken verwenden
 - bspw. Firebase, Geofences, Volley, Google-Material, etc.
 - der Programmierentwurf sollte eine native Geräte-Funktionalität verwenden
 - bspw. Standort, Kamera, Datei-System, Content-Provider, ...

Organisatorisches

Programmierentwurf

- inhaltliche Kriterien (4/5)
 - der Programmierentwurf sollte sich intuitiv benutzen lassen, dazu zählen u.a auch User-Experience und User-Interface Design
 - bspw. Verwendung einer UI-Bibliothek (Google Material), oder eines Wizards zum Anlegen von Einträgen, etc.

Organisatorisches

Programmierentwurf

- inhaltliche Kriterien (5/5)
 - der Quellcode des Programmierentwurfs sollte den Clean-Code Regeln folgen (DRY, KISS)
 - bspw. Methoden sollten nicht unnötig lang sein, Wiederverwendung von einzelnen Code-Bausteinen, Aufteilung in verschiedene Klassen
 - der Quellcode des Programmierentwurfs sollte sauber formatiert sein und Logs enthalten, Kommentare können an komplexeren Stellen Sinn machen
 - bspw. Verwendung eines Code-Formatters

Organisatorisches

Programmierentwurf

- Zusammensetzung
 - 20%: der Funktionsumfang des Programmierentwurfs soll vollständig sein
 - 20%: der Programmierentwurf soll stabil sein
 - 10%: die Funktionalität des Programmierentwurfs soll effizient implementiert werden
 - 10 %: Lifecycle-Events sind zu beachten
 - 5 %: der Programmierentwurf sollte ein remote bzw. ein lokales API anbinden oder andere 3rd-Party-Bibliotheken verwenden
 - 5 %: der Programmierentwurf sollte eine native Geräte-Funktionalität verwenden
 - 10 %: der Programmierentwurf sollte sich intuitiv benutzen lassen, dazu zählen u.a auch User-Experience und User-Interface Design
 - 10 %: der Quellcode des Programmierentwurfs sollte den Clean-Code Regeln folgen (DRY, KISS)
 - 10 %: der Quellcode des Programmierentwurfs sollte sauber formatiert sein und Logs enthalten, Kommentare können an komplexeren Stellen Sinn machen