Briefing

über die Anwendung „ordered.online“
der Gruppe 2 im Praktikum „Application Development for Mobile and Ubiquitous Computing“
an der TU Dresden im Wintersemester 2019/2020

Felix Kästner, Philipp Matthes

Kontextuelle Aspekte

Im Folgenden sind die kontextuellen Aspekte unserer Applikation aufgelistet.

Kontextuelle Dimension	**Kontextueller Aspekt**	**Implementation**
Persönlicher Kontext	Nutzer sollten nur deren persönliche Informationen geben, wenn dies dringend notwendig ist	Login wird nur angefragt, wenn der Nutzer ein zu identifizierender Manager ist
Sozialer Kontext	Mehrere Nutzer sollten (in einer sozialen Gruppe) zusammen bestellen können	Nutzer können sich in die selbe WebSocket-Session einloggen und damit gemeinsam bestellen
Operationaler Kontext	Es bestehen zwei Rollen: der Manager und der Kunde. Beide besitzen unterschiedliche Interessen, müssen aber lückenlos miteinander interagieren	Wir implementieren eine Separation zwischen App und Manager, welche an der richtigen Stelle interoperabel sind, zum Beispiel bei der Bestellung
Technischer Kontext	Die Plattform, auf der die Applikation läuft, kann eine geringe Bandbreite haben und allgemein auf unterschiedlichen Betriebssystemen basieren.	Die Applikation ist darauf ausgelegt, mit geringen Bandbreiten zu funktionieren. Wir transportieren ausschließlich kompakte JSON-Datenpakete. Durch hybride Entwicklung (React Native, Expo) gewährleisten wir Unabhängigkeit vom Betriebssystem.

Geplante und umgesetzte Adaptionskonzepte

Typ der Adaption	Initiation	Reaktion	Implementationsstand
Filtering (Adaption der Applikationsdaten)	Nutzer sucht nach einer Location, gibt dabei einen Standort oder eine Textsuche an	Suche wird serverseitig nach den Attributen der Location gefiltert	Vollständig implementiert
Lazy Loading (Adaption der Applikationsdaten)	Nutzer bestellt Produkte und legt diese somit in der Order Session ab	Server verteilt Aktualisierung, wobei hier nur die ID des Produktes geschickt wird. Das Produkt wird separat vom Server geladen, wenn es benötigt wird.	Vollständig implementiert
Error Handling (Adaption der Kommunikation)	Nutzer scannt QR Code. Der QR Code hat die falsche Länge oder besteht nicht aus alphanumerischen Symbolen.	Der QR Code wird bereits in der App validiert und nicht an das Backend weitergeleitet.	Vollständig implementiert
Error Handling (Adaption der Kommunikation)	Nutzer scannt QR Code der richtigen Länge, jedoch ist dieser ungültig.	Der QR Code wird im Backend geprüft und eine HTTP Error Response wird zurückgegeben, mit einem Reason Identifier. Die App wechselt nicht in die Order Session.	Vollständig implementiert
Error Handling (Adaption der Kommunikation)	Nutzer loggt sich ein, jedoch ist seine Session abgelaufen bzw. die Login Credentials sind ungültig.	Das Backend gibt eine HTTP Response mit einem Error Code und einer Identifikation zurück. Der Client zeigt eine entsprechende Meldung an.	Teilweise implementiert
Queuing (Adaption der Kommunikation)	Nutzer tätigt eine Bestellung, jedoch besteht aktuell keine Internetverbindung	Bestellung wird in einer Warteschlange eingereiht, bis wieder eine Verbindung besteht.	Nicht implementiert
Prioritization (Adaption der Datentransmission)	Eine große Anzahl von Nutzern möchte sich gleichzeitig einloggen.	Skaliere den verantwortlichen Backend-Service entsprechend, um den Workload zu orchestrieren.	Teilweise implementiert