# Testdokumentation

## Test der Android-Anwendung

Im Folgenden sind die Tests der Android-Anwendung dokumentiert.

Es wird jeweils der Anwendungsfall und die getestete Methode beschrieben, außerdem Normalablauf und Sonderfälle, bspw. einem Laufzeitfehler wie Verbindungsprobleme oder falsche Eingaben. Anschließend wird das zu erwartenden und das tatsächliche Testergebnis für den Normalablauf und den Sonderfall dokumentiert. In einigen Fällen wurden die Testergebnisse gekürzt.

**Mitarbeiterlogin:**

|  |  |
| --- | --- |
| Anwendungsfall | Die Bedienung kann sich im System mit ihrem Login-Namen und Login-Passwort authentifizieren. Diese Daten müssen sich zuvor in der Datenbank befinden.  (AW 1b) |
| Verwendete Methode | doInBackground()  onPostExecute() (Klasse: “LoginCheck”)  showToast (String text) |
| **Normalablauf** | Wird „Login“ im Navigation-Drawer ausgewählt, wird ein Bildschirm zum Anmelden im System angezeigt. Um sich anzumelden müssen der Login-Name und das Login-Passwort eingegeben werden.  Stimmen die Daten mit den hinterlegten Daten in der Datenbank überein, dann ist der Mitarbeiter erfolgreich angemeldet.  Der Startbildschirm wird angezeigt. (Bestellung aufgeben).  (Nach dem ersten Anmelden bleiben die Daten gespeichert) |
| Erwartetes Testergebnis | Die Bedienung hat sich erfolgreich im System authentifiziert und kann auf alle Funktionen zugreifen.  Datenbank-Eintrag der Login-Daten für die Bedienungen:    Eingabe in der Android-Anwendung: |
| Tatsächliches  Testergebnis | Die Bedienung hat sich erfolgreich im System authentifiziert und kann auf alle Funktionen zugreifen. Sie wird auf den Startbildschirm weitergeleitet. |
| **Sonderfall 1:** | Die Bedienung hat falsche Login-Daten eingegeben. Das heißt, das Login-Passwort und/oder der Login-Name sind falsch eingegeben worden. |
| Erwartetes Testergebnis | Die Bedienung kann sich im System nicht authentifizieren und hat somit keinen Zugriff auf die restlichen Funktionen.  Für die Bedienung erscheint eine Fehlermeldung, dass die Login-Daten fehlerhaft sind. So kann sie es erneut versuchen die korrekten Login-Daten anzugeben.  Um den Login-Vorgang abzubrechen, muss im Navigations-Drawer der gewünschte Bildschirm ausgewählt werden.  Mit der „Rückgängig“-Taste, kommt man ebenfalls auf den Startbildschirm. (Bestellung aufgeben).  Für diesen Test bleibt die Datenbank dieselbe wie im Normalfall dargestellt. |
| Tatsächliches  Testergebnis | Der Login-Name (wie hier im Beispiel) oder das Passwort ist falsch.  Folgender Bildschirm wird für die Bedienung sichtbar:    Die Fehlermeldung wird angezeigt. Die Bedienung kann Ihre Angaben korrigieren. |
| **Sonderfall 2:** | Es besteht keine Verbindung zur Datenbank.  Die Bedienung kann sich somit nicht im System authentifizieren. |
| Erwartetes Testergebnis | Die Bedienung kann sich im System nicht authentifizieren und hat somit keinen Zugriff auf die restlichen Funktionen.  Für die Bedienung erscheint eine Fehlermeldung, dass die Verbindung zur Datenbank fehlgeschlagen ist. |
| Tatsächliches  Testergebnis | Folgender Bildschirm wird für die Bedienung sichtbar:    Nachdem die Verbindung zur Datenbank wiederhergestellt worden ist, kann sich die Bedienung erneut anmelden. |
| Test  bestanden | Normalablauf: Ja  Sonderfall 1: Ja  Sonderfall 2: Ja |

**Login-Passwort ändern:**

|  |  |
| --- | --- |
| Anwendungsfall | Nachdem der Gastronom die Login-Daten für eine neu eingestellte Bedienung angelegt hat, sollte das Passwort geändert werden. So kann sichergestellt werden, dass nur die Bedienung auf die Login-Daten Zugriff hat.  (AW 2b) |
| Verwendete Methode | doInBackground()  onPostExecute() (Klasse: “ChangeLoiginPassword”)  showToast (String text) |
| **Normalablauf** | Damit das Passwort geändert werden kann, muss zunächst im Navigation-Drawer „Login-Passwort bearbeiten“ ausgewählt werden. Danach wird ein Bildschirm angezeigt, in der die Bedienung die Passwort-Änderung vornehmen kann. Dafür muss der Login-Name, das alte Login-Passwort und das neue - gewünschte - Passwort eingegeben werden. (Das neue Passwort muss wiederholt werden, sodass es zu keinem unbewussten Schreibfehler innerhalb des Passworts kommen kann.)  Nach erfolgreichem ändern des Passworts wird der Startbildschirm dargestellt. (Bestellung aufgeben). |
| Erwartetes Testergebnis | Bei korrekter Eingabe der alten und neuen Login-Daten wird das neue Passwort in der Datenbank hinterlegt. Das alte Passwort wird überschrieben. |
| Tatsächliches  Testergebnis | Die Datenbank wird mit dem neuen Login-Passwort aktualisiert. Die Bedienung kann sich nur noch mit dem neuen Passwort anmelden.  Die Datenbank sieht wie folgt aus: |
| **Sonderfall 1** | Der Login-Name oder das Login-Passwort wurden von der Bedienung falsch eingegeben.  Somit kann das Login-Passwort nicht geändert werden. |
| Erwartetes Testergebnis | Das Passwort in der Datenbank wird nicht aktualisiert.  Der Bedienung wird eine Fehlermeldung angezeigt, dass die von ihr eingegebenen Login-Daten fehlerhaft sind.  Die Eingaben für die Passwörter werden gelöscht. So wird die Bedienung gezwungen, diese nochmals einzugeben.  Anschließend kann die Bedienung erneut versuchen ihr Login-Passwort zu ändern. |
| Tatsächliches  Testergebnis | Die folgende Fehlermeldung wird der Bedienung angezeigt.    Die Datenbank wurde nicht aktualisiert. |
| **Sonderfall 2** | Die neuen Passwörter stimmen nicht überein. |
| Erwartetes Testergebnis | Das Passwort wird in der Datenbank nicht aktualisiert.  Der Bedienung wird in Rot eine Fehlermeldung angezeigt, dass die Passwörter nicht übereinstimmen. Die Änderung der Passwörter kann nicht erfolgen. |
| **Sonderfall 3** | Es besteht keine Verbindung zur Datenbank |
| Erwartetes Testergebnis | Es wird beim betätigen des „PASSWORT ÄNDERN“-Buttons eine Fehlermeldung dargestellt.  Die Datenbank kann nicht aktualisiert werden.  Nach erneuter Verbindung mit der Datenbank kann das Passwort geändert werden. |

|  |  |
| --- | --- |
| Tatsächliches  Testergebnis | Die folgende Fehlermeldung wird der Bedienung dargestellt.    Sie kann die Passwörter abgleichen und es nochmals versuchen.  Die Datenbank wurde nicht aktualisiert. |
| Test  bestanden | Normalablauf: Ja  Sonderfall 1: Ja  Sonderfall 2: Ja  Sonderfall 3: Ja |

**Artikel als produziert markieren:**

|  |  |
| --- | --- |
| Anwendungsfall | In der Datenbank soll vermerkt werden, dass ein Artikel von der Bedienung abgeholt wurde. Anschließend soll der Artikel zum jeweiligen Kunden gebracht werden.  Dadurch kann im Falle eines Fehlers identifiziert werden, welcher Artikel bereits an einen Kunden geliefert wurde.  (AW 3b) |
| Verwendete Methode | doInBackground()  onPostExecute() (Klasse: “UpdateOrder”)  showToast (String text) |
| **Normalablauf** | Damit der entsprechende Artikel „markiert“ werden kann muss im Navigation-Drawer „Bestellungsannahme“ ausgewählt werden. In dem dann angezeigten Bildschirm werden alle nicht produzierten Artikel aufgelistet. Dabei wird der Artikel -name, -kommentar und der dazugehörige Tisch angezeigt.  Der von der Bedienung angenommene Artikel kann in dieser Darstellung ausgewählt werden.  Mit anschließendem klicken auf den „BESTÄTIGEN“-Button werden die ausgewählten Artikel in der Datenbank als produziert hinterlegt. |
| Erwartetes Testergebnis | Die Artikel werden in der Datenbank als „produziert“ markiert.  Die Datenbank sieht wie folgt aus:    Im Test wird der Artikel mit der „orderedItemID“ *203 markiert.*  Im Beispiel der Artikel Cola an Tisch *A1* mit dem Kommentar *aus der Flasche*: |
| Tatsächliches  Testergebnis | Der Artikel mit der „orderedItemID“ *203* wurde als produziert hinterlegt.  Die Datenbank sieht wie folgt aus:    Die Daten wurden erfolgreich in der Datenbank eingetragen. |
| **Sonderfall** | Es besteht keine Verbindung zur Datenbank. Die Daten können somit nicht aktualisiert werden. |
| Erwartetes Testergebnis | Der Bedienung wird eine Fehlermeldung dargestellt, dass keine Verbindung zur Datenbank existiert.  Anschließend wird der Startbildschirm der Android-Anwendung dargestellt. Die Bedienung muss nach erfolgreichem wiederverbinden mit dem Netzwerk die Artikel nochmals eintragen. |
| Tatsächliches  Testergebnis | Die Bedienung erhält folgende Fehlermeldung: |
| Test  bestanden | Normalablauf: Ja  Sonderfall: Ja |

**Rechnung splitten:**

|  |  |
| --- | --- |
| Anwendungsfall | Möchte eine Gruppe von Kunden ihre Bestellung getrennt bezahlen, so kann der Preis einfach dargestellt werden. Die Bedienung kann für jeden Kunden eingeben welche Artikel bezahlt werden sollen. Anschließend kann sie an dem Bildschirm ablesen was der jeweilige Kunde zu zahlen hat. Dies kann die Bedienung für jeden Kunden separat machen.  (AW 4b) |
| Verwendete Methode | doInBackground()  onPostExecute() (Klasse: “UpdateOrder”)  showToast (String text) |
| **Normalablauf** | Eine Bedienung kommt an einen Tisch, an denen verschiedene Personen bezahlen möchten.  Sie geht in die Bestellung und klickt dann auf den Button „BEZAHLEN“. Danach wird ihr ein neuer Bildschirm angezeigt. Sie kann nun auswählen, ob sie Kunden gemeinsam bezahlen oder ob jeder Kunde für sich bezahlen möchte.  Wenn jeder Kunde einzeln Zahlen möchte, markiert sie die Checkbox „Separat Bezahlen“. Nun kann der Kunde die Artikel getrennt zahlen. |
| Erwartetes Testergebnis | Der Preis wird richtig berechnet und dargestellt.  Die Artikel die ausgewählt wurden, sind in der Datenbank als bezahlt hinterlegt.  Die Datenbankinhalte vor dem Test sehen wie folgt aus:    Die separat bezahlten Artikel:    Eine Cola und ein Burger werden für 7.0 € bezahlt. Der Rest der Rechnung bleibt offen. |
| Tatsächliches  Testergebnis | Die Datenbank wird aktualisiert. Die Artikel die bezahlt wurden werden auf „1“ gesetzt.  Der Preis wird in der Anwendung richtig berechnet und dargestellt.  Die Datenbankinhalte nach dem Test:    Die Artikel mit der „orderedItemID“ *207* und *209* wurden bezahlt. |
| **Sonderfall** | Die Rechnung wird auf einmal bezahlt. Die Kunden möchten ihre Rechnung nicht separat bezahlen. Der Vorgang verhält sich gleich wie im Normalfall. Jedoch darf die Checkbox „Separat Bezahlen“ nicht aktiviert werden. |
| Erwartetes Testergebnis | Alle Artikel werden in der Datenbank als bezahlt hinterlegt. Der Preis der Bestellung wird der Bedienung richtig dargestellt.    Der Rest der Bestellung aus dem Normalfall wird nun zusammen bezahlt. In der Android-Anwendung sieht das wie folgt aus: |
| Tatsächliches  Testergebnis | Die Datenbank wird aktualisiert. Die Artikel die bezahlt wurden werden auf „1“ gesetzt.  Der Preis wird in der Anwendung richtig berechnet und dargestellt.  Die Datenbank wird wie folgt dargestellt:    Alle Artikel mit „OrderID“ *95* wurden bezahlt. |
| Test  bestanden | Normalablauf: Ja  Sonderfall: Ja |

**Kundenbeleg anfordern**

|  |  |
| --- | --- |
| Anwendungsfall | Falls ein Kunde einen Beleg für den Besuch in der Gastronomie möchte, kann ein Kundenbeleg ausgedruckt werden. (Mit Tisch, |
| Verwendete Methode | doInBackground()  onPostExecute() (Klasse: “PrinterSalesCheck”)  showToast (String text) |
| **Normalablauf** | Eine Bedienung kommt an einen Tisch, an dem Kunden bezahlen möchten.  Sie geht in die Bestellung und klickt dann auf den Button „BEZAHLEN“. Danach wird ihr ein neuer Bildschirm angezeigt. Sie kann nun auswählen, ob die Kunden einen Beleg möchten oder nicht.  Wenn der Kunde einen möchte, muss die Checkbox „Kundenbeleg“ angewählt werden. Nach anschließendem klicken auf den Button „Bezahlen“, wird der Beleg ausgedruckt. |
| Erwartetes Testergebnis | Der Kundenbeleg wird mit Anschrift, bestellten Artikeln, Preis und Tisch ausgedruckt.  Im Test sieht die Bestellung wie folgt aus: |

|  |  |
| --- | --- |
| Tatsächliches  Testergebnis | Der Beleg sieht wie folgt aus: |
| **Sonderfall** | Die Kunden möchten getrennt Bezahlen und möchten einen Kundenbeleg.  Das Vorgehen ist dasselbe wie im Normalfall beschrieben. |
| Erwartetes Testergebnis | Der gesamte Beleg einer Bestellung wird ausgedruckt. Dieser kann dann für jeden Kunden einzeln ausgedruckt werden.  Im Test sieht die Bestellung wie folgt aus: |

|  |  |
| --- | --- |
| Tatsächliches  Testergebnis | Der Beleg sieht wie folgt aus: |
| Test  bestanden | Normalablauf: Ja  Sonderfall: Ja |

**Bestellten Artikel kommentieren:**

|  |  |
| --- | --- |
| Anwendungsfall | Der Kunde möchte bei einem Artikel eine bestimmte Art der Zubereitung oder hat einen sonstigen Wunsch, den der Koch beachten soll.  Dann kann dem bestellten Artikel eine Information mitgegeben werden. Diese Information wird dann ebenfalls in der Datenbank abgelegt. |
| Verwendete Methode | doInBackground()  onPostExecute() (Klasse: “UpdateOrder”)  showToast (String text) |
| **Normalablauf** | Ein Kunde äußert einen Wunsch zu einem Artikel. Dieser Wunsch kann dem Artikel angefügt werden. Dazu muss die Bedienung im „Bestellung aufgeben“ Modus auf den Artikel lange drücken. Es öffnet sich ein neues Fenster, in dem die Bedienung den geäußerten Wunsch anmerken kann.  Anschließend kann die Bestellung normal zu Ende geführt werden. |
| Erwartetes Testergebnis | Der Kommentar wird gespeichert und in der Datenbank hinterlegt. Der Küchenbeleg beinhaltet den Kommentar.  Der Bestellung wird folgender Kommentar angeheftet: |
| Tatsächliches  Testergebnis | Die Datenbank sieht nach der Bestellung wie folgt aus:    Der Küchenbeleg sieht wie folgt aus: |
| **Sonderfall** | Dem Artikel soll ein Wunsch hinzugefügt werden, nachdem dieser bereits bestellt wurde. |
| Erwartetes Testergebnis | In der Datenbank wird der Kommentar nicht mehr gespeichert und kann somit auch nicht berücksichtigt werden.  Der Kundenwunsch für den Artikel: |
| Tatsächliches  Testergebnis | Die Datenbank sieht wie folgt aus:    Die Datenbank wurde nicht aktualisiert. |
| Test  bestanden | Normalablauf: Ja  Sonderfall: Ja |