|  |  |  |
| --- | --- | --- |
| Name | User Story | Akzeptanzkriterien |
| Begrüßung | Als User möchte ich den **S**imple **B**arkeeper einfach per Sprachbefehl "Alexa starte Simple Barkeeper" starten können und von ihm begrüßt werden, um das Gefühl eines Gesprächs zu bekommen und zu wissen, dass Alexa mich verstanden hat. | * Nutzer startet SBmit Sprachbefehl "Alexa starte Simple Barkeeper". * SB begrüßt Nutzer, um Kontakt aufzubauen und Eingabe zu bestätigen. |
| Drink eingeben | Als User möchte ich dem Simple Barkeeper eigene Drinks inkl. Zutaten hinzufügen können, um sie mir nicht merken zu müssen, sie später abfragen zu können und den Drink vorgeschlagen zu bekommen. | * User ruft Funktion durch Ausspruch von "Drink eingeben" auf. * Nutzer kann eigene Drinks hinzufügen. * Drinkname wird vom Nutzer eingegeben und vom SB als Bestätigung widerholt, sowie Korrektheit durch Nutzer geprüft. * Bei Fehler erfolgt Neueingabe. * Jede Zutat und zugehörige Menge werden einzeln abgefragt, vom SB widerholt und auf Korrektheit vom Nutzer überprüft. * Bei Fehler erfolgt Neueingabe. * Nach jeder Zutat wird gefragt, ob weitere Zutat eingegeben werden soll. * Falls ja, wiederholt sich dieser Block. * Falls nicht, wird bestätigt, dass Drink gespeichert wurde, * Möglichkeit abzubrechen und in Hauptmenü zurückzukehren. |
| Drink löschen | Als User möchte ich nicht mehr benötigte Drinks (inkl. Zutaten) aus dem Simple Barkeeper löschen können, um sie nicht mehr vorgeschlagen zu bekommen. | * User ruft Funktion durch Ausspruch "Drink löschen" auf. * Drinkname wird vom Nutzer genannt. * SB widerholt den Namen und prüft somit ob der richtige Drink erkannt wurde. * Nach Ja/Nein Antwort des Nutzers wird entsprechender Drink gelöscht oder Abfrage widerholt. * Abschließende Bestätigung der Löschung. * Möglichkeit abzubrechen und in Hauptmenü zurückzukehren. |
| Favoriten festlegen | Als User möchte ich im Simple Barkeeper einen Drink als Favoriten festlegen können, um ihn später ohne namentliche Nennung abrufen zu können bzw. mir den Namen nicht merken zu müssen. | * User ruft Funktion durch den Ausspruch "Favoriten festlegen" auf. * Name des Favoriten wird vom SB abgefragt. * Falls Getränk vorhanden wird Name widerholt Bestätigung vom Nutzer eingeholt, dass er richtig verstanden wurde. * Falls ja, wird Favorit gespeichert und eine Meldung diesbezüglich ausgegeben. * Falls nein, wird Block widerholt. * Möglichkeit abzubrechen und in Hauptmenü zurückzukehren. |
| Favorit abrufen | Als User möchte ich im Simple Barkeeper einen festgelegten Drink, den Favoriten, ohne namentliche Nennung abrufen können, um nicht über den exakten Namen nachdenken zu müssen oder ihn mir zu merken. | * User sagt "Favorit abrufen" und erhält als Antwort vom SB den Namen des Drinks. * Fehlermeldung, falls zuvor kein Drink festgelegt wurde. * Möglichkeit abzubrechen und in Hauptmenü zurückzukehren. |
| Zutaten abfragen | Als User möchte ich vom Simple Barkeeper die Zutaten eines bestimmten Drinks abfragen können, um ihn mir zuzubereiten. | * User sagt "Zutaten abfragen" wird vom SB nach dem Getränkenamen/Favoriten gefragt. * Je nach Auswahl erhält der User vom SB als Antwort eine Liste von Zutaten und zugehörigen Mengen. * Ist kein Favorit festgelegt oder Drink nicht vorhanden, erfolgt Fehlermeldung durch SB. * Möglichkeit abzubrechen und in Hauptmenü zurückzukehren. |
| Drink vorschlagen | Als User möchte ich vom Simple Barkeeper tageszeitabhängig einen Drink vorgeschlagen bekommen - anhand einer genannten Zutat, eines Geschmacks oder per Zufall und unter der Angabe ob Alkohol erwünscht ist. Ich will mir keine eigenen Gedanken zu einem Getränkewunsch machen und eventuell vorhandene Zutaten aufzubrauchen. | * User ruft Funktion per Ausruf "Drink vorschlagen auf" und wird vom SB gefragt, ob Alkohol erlaubt ist. * Nach seiner Antwort wird er vom SB gefragt, ob nach Zutat/Geschmack/Zufall. * Bei der Wahl "Zutat" wird vom SB eine gewünschte Zutat erfragt, bei der Wahl "Geschmack" ein Geschmack. Bei Zufall wird random ausgewählt. * User nennt Zutat/Geschmack und erhält, einen entsprechenden Vorschlag vom SB (Drinkname). * Danach erfolgt Frage, ob weiterer Vorschlag erfolgen soll. * Falls ja, neuer Vorschlag nach bekannten Vorgaben. * Falls nein, Rückkehr zum Hauptmenü. * Möglichkeit abzubrechen und in Hauptmenü zurückzukehren. |
| Rezept abfragen | Als User möchte ich das Rezept eines Drinks abfragen können. Das heißt die Zubereitungsschritte sollen einzeln aufgelistet werden und es soll die Möglichkeit geben zum vorherigen Schritt zurück zu gehen, sich den aktuellen Schritt wiederholen zu lassen oder mit dem nächsten Schritt fortzufahren. | * Der User ruft die Funktion per Ausruf „Rezept abfragen“ auf und wird vom SB nach dem gewünschten Drink gefragt * Ist der vom User genannte Drink vorhanden, gibt der SB den ersten Schritt zur Zubereitung des Drinks aus. Der User kann dann „wiederholen“, „weiter“ oder „zurück“ sagen. * Sagt der User „wiederholen“ so wiederholt der SB den aktuellen Schritt. * Sagt der User „weiter“, so fährt der SB mit dem nächsten Schritt fort. Wenn es keinen nächsten Schritt gibt, teilt Alexa dem User mit, dass der Drink jetzt fertig ist und kehrt zum Hauptmenü zurück. * Sagt der User „zurück“ kehrt der SB zum vorherigen Schritt zurück und wiederholt ihn. Wenn es keinen vorherigen Schritt gibt, sagt der SB, dass der aktuelle Schritt der Erste ist und wiederholt ihn. * Der User kann außerdem jederzeit „abbrechen“ sagen um zum Hauptmenü zurückzukehren. |
| Hilfe verlangen | Als User möchte ich vom Simple Barkeeper erfahren, welche Optionen mir zur Verfügung stehen um ihn richtig verwenden zu können. Bei genauerer nachfrage soll ein Beispiel aufgeführt werden. | * User sagt "Hilfe" und der SB erklärt ihm kurz alle Punkte des Hauptmenüs. * User sagt „Hilfe“ + Funktionsname (z.B. „Hilfe Favoriten abfragen“) und der SB führt dem User ein Beispiel zur gewünschten Funktion auf. |
| Barkeeper schließen | Als User möchte ich den Simple Barkeeper auf Zuruf schließen können um das Gespräch zu beenden. | * Der SB wird durch den Ausruf "Beenden" beendet. * Der SB verabschiedet sich wie ein Gesprächspartner vom User. |