

Requirements-Dokumentation für das Projekt *"Home Harmony Hero"*

1. Einleitung

Die Requirements-Dokumentation beschreibt die Anforderungen an die Anwendung "Home Harmony Hero", die entwickelt wird, um die Kommunikation und Planung von Aufgaben, Einkäufen und Mitbewohnern in einem gemeinsamen Haushalt zu unterstützen.

Die Dokumentation beschreibt die funktionale und nicht-funktionale Anforderungen der Anwendung. Die Anwendung basiert auf JavaFX und einer MySQL-Datenbank.

2. Ziel

"Home Harmony Hero" soll die Organisation und Planung im Haushalt vereinfachen, indem es den Benutzern ermöglicht, Aufgaben zuzuweisen, Einkaufslisten zu verwalten und die Mitbewohner im Haushalt zu koordinieren. Die Anwendung soll benutzerfreundlich, intuitiv und effizient sein.

3. Anforderungen

3.1. Funktionale Anforderungen

- Die Anwendung ermöglicht es Benutzern, neue Mitbewohner hinzuzufügen.
- Mitbewohner können ausgewählt, bearbeitet und gelöscht werden.
- Einkaufslisten können erstellt, bearbeitet und gelöscht werden.
- Aufgaben können erstellt, bearbeitet und gelöscht werden.
- Der Status von Aufgaben kann geändert werden.
- Benutzer können Produkte zur Einkaufsliste hinzufügen und bearbeiten.
- Aufgaben können verschiedenen Mitbewohnern zugewiesen werden.

3.2. Nicht-funktionale Anforderungen

- Die Anwendung muss benutzerfreundlich und intuitiv sein.
- Die Datenbankverbindung muss sicher und stabil sein.
- Die Anwendung soll responsiv und gut strukturiert sein.
- Änderungen in der Einkaufsliste und Aufgabenliste sollen sofort in der Datenbank gespeichert werden.
- Die Anwendung sollte gut dokumentiert sein, um zukünftige Entwicklungen zu erleichtern.

4. Anforderungsdetails

4.1. Roommate Management

- Benutzer können Mitbewohner hinzufügen und löschen.
- Die Mitbewohnerdaten werden in der Datenbank gespeichert.

Requirement ID	1
Requirement Name	Neuen Mitbewohner hinzufügen
Requirement description	Benutzer können einen neuen Mitbewohner zur Mitbewohnergruppe hinzufügen
Type of requirement	Funktion
Priority	hoch
Status	implementiert

Requirement ID	2
Requirement Name	Mitbewohner anzeigen
Requirement description	Benutzer bekommen alle existierenden Mitbewohner angezeigt
Type of requirement	Funktion
Priority	mittel
Status	implementiert

Requirement ID	3
Requirement Name	Mitbewohner entfernen
Requirement description	Benutzer können einen Mitbewohner entfernen
Type of requirement	Funktion
Priority	hoch
Status	implementiert

4.2. Task Management

- Benutzer können Aufgaben erstellen, bearbeiten und löschen.
- Aufgaben können verschiedenen Mitbewohnern zugewiesen sein.
- Bearbeiten von Aufgaben durch Änderung des Titels, zugeordneten Mitbewohners oder Status
- Markieren von Aufgaben als “erledigt”, “ausstehend” oder “in Arbeit”
- Die Aufgabenliste wird in der Datenbank gespeichert.

Requirement ID	4
Requirement Name	Neue Aufgabe hinzufügen
Requirement description	Die Anwendung ermöglicht es Benutzern, neue Aufgaben hinzuzufügen.
Type of requirement	Funktion
Priority	hoch
Status	implementiert

Requirement ID	5
Requirement Name	Liste von Aufgaben anzeigen
Requirement description	
Type of requirement	Funktion
Priority	mittel
Status	implementiert

Requirement ID	6
Requirement Name	Aufgabendetails bearbeiten
Requirement description	
Type of requirement	Funktion
Priority	hoch
Status	implementiert

Requirement ID	7
Requirement Name	Aufgaben als abgeschlossen markieren
Requirement description	
Type of requirement	Funktion
Priority	mittel
Status	implementiert

Requirement ID	8
Requirement Name	Aufgaben löschen
Requirement description	
Type of requirement	Funktion
Priority	mittel
Status	implementiert

4.3. Shopping List Management

- Hinzufügen neuer Produkte zur Einkaufsliste mit Produktname, Mengenangabe und zugeordneten Mitbewohner
- Produkte können bearbeitet und gelöscht werden.
- Die Einkaufsliste wird in der Datenbank aktualisiert und gespeichert.

Requirement ID	9
Requirement Name	Neues Produkt zur Einkaufsliste hinzufügen
Requirement description	Die Anwendung ermöglicht es Benutzern, neue Produkte zur Einkaufsliste hinzuzufügen
Type of requirement	Funktion

Priority	mittel
Status	implementiert

Requirement ID	10
Requirement Name	Einkaufsliste anzeigen
Requirement description	Benutzer können die gesamte Einkaufsliste mit allen Produkten anzeigen
Type of requirement	Funktion
Priority	hoch
Status	implementiert

Requirement ID	11
Requirement Name	Produktdetails bearbeiten
Requirement description	Benutzer können die Details eines Produkts in der Einkaufsliste bearbeiten
Type of requirement	Funktion
Priority	mittel
Status	implementiert

Requirement ID	12
Requirement Name	Produkt von Einkaufsliste entfernen

Requirement description	Benutzer können ein Produkt von der Einkaufsliste entfernen
Type of requirement	Funktion
Priority	mittel
Status	implementiert

Requirement ID	13
Requirement Name	Einkaufsliste löschen
Requirement description	Benutzer können aus der Einkaufsliste löschen
Type of requirement	Funktion
Priority	hoch
Status	implementiert

5. Technische Anforderungen

1. Verwendung von JavaFX für die Benutzeroberfläche
2. Implementierung einer Datenbankverbindung für die Speicherung von Aufgaben, Einkäufen und Mitbewohnern
3. Einhaltung des MVC (Model-View-Controller) Design Patterns zur Trennung von Daten, Benutzeroberfläche und Steuerlogik

6. Benutzeroberfläche

1. Übersichtliche Darstellung der Aufgaben, Einkaufslisten und Mitbewohner in Tabellen
2. Verwendung von SplitMenuButtons zur Auswahl von Status, zugeordneten Mitbewohnern und Aktionen
3. Eingabefelder für die Eingabe von neuen Aufgaben, Produkten und Mitbewohnern
4. Buttons für das Hinzufügen, Bearbeiten und Löschen von Elementen
5. Tooltips für Buttons zur Anzeige von Hinweisen

7. Beispielhafte Nutzung der Anwendung

7.1. Mitbewohner hinzufügen

1. Benutzer klickt auf "Mitbewohner hinzufügen".
2. Benutzer gibt den Namen des neuen Mitbewohners ein und bestätigt.
3. Der neue Mitbewohner wird der Liste hinzugefügt.

7.2. Produkt zur Einkaufsliste hinzufügen

1. Benutzer wählt einen Mitbewohner aus der Liste aus.
2. Benutzer gibt ein Produkt, die Menge, von wem es hinzugefügt wird, ein und fügt es zur Einkaufsliste hinzu.
3. Das Produkt wird in der Einkaufsliste angezeigt.

7.3. Aufgabe erstellen

1. Benutzer gibt den Namen der Aufgabe ein und wählt einen Mitbewohner und Status aus.
2. Benutzer erstellt die Aufgabe, die in der Liste angezeigt wird.

8. Diagramme und Tabellen

8.1. Klassendiagramm

Das Klassendiagramm zeigt die Struktur und Beziehung zwischen "Roommate", "ShoppingList", "Task" und anderen Klassen im Projekt.

8.2. ER-Diagramm

Das Datenbankdiagramm zeigt die Struktur der Datenbanktabellen (Roommates, ShoppingList, Tasks) und deren Beziehungen.

8.3. Use-Case-Diagramm

Das Use-Case-Diagramm visualisiert die Benutzerinteraktionen mit der Anwendung, z.B. das Hinzufügen von Produkten zur Einkaufsliste oder das Bearbeiten von Aufgaben.

9. Zeitkonfiguration

Das Projekt "Home Harmony Hero" soll innerhalb von drei Monaten umgesetzt und getestet werden. Dabei wird folgender Zeitplan eingehalten:

1. Design und Implementierung der Datenbank und Datenmodelle
– Dauer: 2 Wochen
2. Entwicklung der Benutzeroberfläche und Implementierung der Basisfunktionalitäten –
Dauer: 6 Wochen
3. Feinschliff, Testing und Bugfixing, Vorbereitung auf die Abschlusspräsentation
–Dauer: 4 Wochen

10. Beispiele

- Implementierung der Funktion zum Bearbeiten von Aufgaben. Hierbei müssen verschiedene Bedingungen, z.B. die Auswahl des neuen Status und des zugewiesenen Mitbewohners, überprüft und umgesetzt werden
- Wenn ein Benutzer ein neues Produkt zur Einkaufsliste hinzufügt, muss er die Menge und den zugehörigen Mitbewohner auswählen.
- Verwaltung der Mitbewohner und deren Zuweisung zu Produkten und Aufgaben erfordert eine sichere Datenbankverwaltung und eine benutzerfreundliche Oberfläche für die Interaktion.

Fazit

Die Requirements-Dokumentation für das Projekt "Home Harmony Hero" bietet eine klare und umfassende Beschreibung der Anwendung, ihrer Funktionen und Anforderungen. Es dient als Leitfaden für Entwickler, Designer und Stakeholder, um sicherzustellen, dass die Anwendung gemäß den Spezifikationen entwickelt wird. Durch die Verwendung von Diagrammen, Tabellen und Beispielen wird sichergestellt, dass das Dokument leicht verständlich und informativ für alle Projektbeteiligten ist.