Funktionsbeschreibung:

[Funktion: Kontaktdaten abrufen aus der DB 1](#_Toc353624003)

[Funktion: Kontaktdaten filtern 1](#_Toc353624004)

[Funktion: Kontakt suchen 1](#_Toc353624005)

[Funktion: Kontakt anlegen 2](#_Toc353624006)

[Funktion: Kontakt bearbeiten 2](#_Toc353624007)

[Funktion: Kontakt löschen 2](#_Toc353624008)

[Funktion: Angebot abrufen aus der DB 3](#_Toc353624009)

[Funktion: Angebot anlegen 3](#_Toc353624010)

[Funktion: Angebot bearbeiten 3](#_Toc353624011)

[Funktion: Angebot löschen 3](#_Toc353624012)

## Funktion: Nutzer Registrieren

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich, sofern noch nicht geschehen, beim System registrieren, um sich anmelden zu können |
| Input | * Name + neues Passwort + Passwort bestätigen |
| Output | * Rückmeldung über Erfolg und Misserfolg der Registrierung |
| Vorbereitungen | * keine |
| Nachbedingungen | * Nutzer ist registriert |
| Fehlerfälle | * Nutzername schon vorhanden * Passwort falsch bestätigt * Zugriff auf DB gescheitert |

## Funktion: Nutzer anmelden

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich beim System anmelden, um zu einem erweiterten Funktionsumfang Zugriff zu erlangen |
| Input | * Name + Passwort |
| Output | * Rückmeldung Erfolg / Misserfolg der Anmeldung |
| Vorbereitungen | * User muss registriert sein |
| Nachbedingungen | * Nutzer ist angemeldet |
| Fehlerfälle | * User ist nicht registriert * Zugriff auf DB gescheitert |

## Funktion: Nutzer abmelden

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich vom System abmelden, um unerlaubte und unerwünschte Zugriffe durch andere zu verhinden |
| Input | * Abmelden bestätigt |
| Output | * Rückmeldung Erfolg / Misserfolg der Abmeldung |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Nutzer ist abgemeldet |
| Fehlerfälle |  |

## Funktion: Status des Angebotes ändern

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann den Zustand eines Angebotes von „offen“ auf „abgelaufen“ oder „bestellt setzen“ |
| Input | * Statusänderung (nach „Abgelaufen“/ „Gekauft“) |
| Output |  |
| Vorbereitungen | * Nutzer ist angemeldet * Angebot ist selektiert |
| Nachbedingungen | * Status wurde geändert |
| Fehlerfälle | * keine |

## Funktion: Rechnung erstellen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich zu einem Bestimmten Angebot eine Rechnung erstellen lassen |
| Input | * Alle Daten aus bereits vorhandenem Daten Datenbankdaten ( *AngebotID, Gesamtpreis (aus Einzelpreis \* Anzahl + Lieferkosten) , Kontakt des Käufers, Kaufdatum* ) |
| Output | * Ausgabe der Rechnung bei Erfolg |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Bezahlt-Status der Rechnung auf „offen“, Verschickt-Status auf „offen“, Interne ID zugewiesen |
| Fehlerfälle | * Zugriff auf DB gescheitert |

## Funktion: Kontaktdaten abrufen aus der DB

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann nach dem er sich an das System angemeldet hat, sein Kontaktbuch einsehen welches eine Liste der von Ihm angelegten Kontakte beinhaltet. |
| Input |  |
| Output | * Alle Kontakte die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Hochzählen des Bestellungszählers, nach Überprüfung der Rechnungen |
| Fehlerfälle |  |

## Funktion: Kontaktdaten filtern

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit beim ansehen seines Kontaktbuches die Daten nach bestimmten Kriterien Filtern zu lassen. |
| Input | * Auswählen der Art der Filterung   + Alphabetische Anordnung nach Name/Anschrift etc.   + Nach Anzahl Bestellungen |
| Output | * Alle Kontakte die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * keine |

## Funktion: Kontakt suchen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Kontakte direkt Suchen zu lassen. |
| Input | * Name ggf. Anschrift |
| Output | * Alle Kontake die auf die jeweiligen Suchkritieren zutreffen |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * Kontakt existiert nicht in der DB ->eine Meldung wird angezeigt das die Suchanfrage 0 Treffer ergab * Rechnung nicht verfügbar in der DB ->sollte die Rechnung von dem Kunden nicht auffindbar sein wird der Zähler wieder runtergezählt von der Anzahl der Bestellungen |

## Funktion: Kontakt anlegen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit neue Kontake in sein Kontaktbuch hinzuzufügen. |
| Input | * Pflicht Eingabe von Vorname, Nachname, Anschrift, * Optionale Angaben: email, telefon, mobil, titel |
| Output | * Anzeige des neu Angelegten Kontaktes |
| Vorbereitungen | * Nutzer ist angemeldet, |
| Nachbedingungen | * Bestellungszähler wird auf 0 gesetzt, Kontakt wird in die DB eingetragen wenn es zu keinem fehler in der Kontrolle kommt |
| Fehlerfälle | * Kontakt existiert bereits in der DB ->Eingabe Felder Vorname + Nachname werden überprüft in Kombination mit der Anschrift, sollte der Kontakt schon existieren bekommt der Nutzer einen Hinweis * Kontakt enthält Fehlerhafte Daten -> mit Hilfe von Try catch Kontrolle und Vermeidung von Eingabe von Fehlerhaften eingaben |

## Funktion: Kontakt bearbeiten

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Kontaktdaten zu bearbeiten. |
| Input | * wie beim „Anlegen des Kontaktes“ |
| Output | * Anzeige des geänderten Kontaktes |
| Vorbereitungen | * Nutzer ist angemeldet, Kontakt Daten wurden geladen und werden angezeigt |
| Nachbedingungen | * Kontrolle auf Fehlerhafte eingaben (Leerfelder und falsche Formateingabe) |
| Fehlerfälle | * Siehe Kontakt anlegen |

## Funktion: Kontakt löschen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Kontake in sein Kontaktbuch zu löschen. |
| Input | * Auswählen des zu löschenden Kontaktes durch ID bzw Name |
| Output | * Meldung das Löschung erfolgreich war |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Der Rechnungszähler (Bestellungen) ist auf 0 |
| Fehlerfälle | * Kontakt existiert nicht in der DB -> der Nutzer bekommt einen Hinweis das der von ihm gewählte Kontakt nicht existiert mehr in der DB * Fehler beim Löschen des Kontakt in der DB->Hinweis, Kontakt wird wieder hergestellt * Kontakt besitz verknüpfungen zu Rechnungen-> keine löschung möglich in dem fall, benutzer wird darauf hingewiesen |

## Funktion: Angebot abrufen aus der DB

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann nach dem er sich an das System angemeldet hat, die Liste seiner angelegten Angebote einsehen. |
| Input | * Keine |
| Output | * Alle Angebote die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * Nutzer besitzt keine Angebote-> in dem Fall wird eine Leere Liste angezeigt |

## Funktion: Angebot anlegen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit neue Angebote in sein anzulegen.  Es werden hier bei zwei einträge in die DB gemacht 1. Produkt anlegen 2. Angebot anlegen. |
| Input | * Angebot   + Pflicht Eingabe von Startpreis, buynowprice, produkt (wird im zusammenhang mit dem angebot angelegt), Auktionshaus, URL   + Optionale Angaben: end datum des angebots * Produkt   + Pflicht: je nach art des Produkttypes gibt es hier unterschiedliche eingabe Felder bitte hierzu auf das DatenModel Dokument schaun! Abschnitt Produkt |
| Output | * Anzeige des neu Angelegten Angebots wenn erfolgreich * Bei Fehler gleiche anlege Formular mit hinweis |
| Vorbereitungen | * Nutzer ist angemeldet, |
| Nachbedingungen | * Start datum des angebots auf das erstellungsdatum des angebots setzen * Verknüpfung zum angemeldeten Nutzer vermerken durch FK * Status des angebots wird beim erstellen immer auf offen gesetzt * Produkt muss erfolgreich vor anlegen des Angebots in de DB eingetragen worden sein |
| Fehlerfälle | * Angebotkonnte nicht erstellt werden in der DB ->Eingabe war fehlerhaft, der Nutzer einen Hinweis hier, Produkt eintrag wird wieder aus der DB gelöscht * Angebot enthält Fehlerhafte Daten -> mit Hilfe von Try catch Kontrolle und Vermeidung von Eingabe von Fehlerhaften eingaben * Produkt existiert nicht in der DB -> Fehler meldung, |

## Funktion: Angebot bearbeiten

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Angebots zu bearbeiten. |
| Input | * wie beim „Anlegen des Angebots“ |
| Output | * Anzeige des geänderten Angebots |
| Vorbereitungen | * Nutzer ist angemeldet, Angebots Daten wurden geladen und werden angezeigt |
| Nachbedingungen | * Kontrolle auf Fehlerhafte neu eingaben (Leerfelder und falsche Formateingabe) |
| Fehlerfälle | * Siehe Angebot anlegen |

## Funktion: Angebot löschen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann Angebote wieder löschen. Beim Löschen werden auch die Verknüpften Produkte aus der DB gelöscht. |
| Input | * Keine |
| Output | * Keine |
| Vorbereitungen | * Nutzer ist angemeldet, Angebot zum löschen ausgewählt |
| Nachbedingungen | * Kontrolle ob es Verknüpfungen zu Rechnungen gibt und Produkte (Produkt wird auch aus der DB gelöscht) |
| Fehlerfälle | * Angebot existiert nicht in der DB ->Fehlermedlung Löschen ist in diesem Fall nicht möglich * Angebot ist gekoppelt an eine Rechnung -> das Angebot kann in diesem Fall nicht gelöscht werden da erst die Rechnung aus dem System genommen werden muss * Produkt nicht in der DB gefunden -> löschen Rückgängig machen und eine Fehlermeldung ausgeben |

## Funktion: Produkt anlegen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann ein neues Produkt in das System einpflegen.  Hierzu werden je nach Produktart Schablonen zur Eingabe benutzt |
| Input | * Typ des Artikels (siehe erbende Klassen von IArtikel) * Typzugehörige Daten (siehe Datenmodell) |
| Output | * Anzeige des angelegten Produktes |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Produktname einzigartig |
| Fehlerfälle | * Produkt existiert bereits --> Nutzer bekommt Hinweis, Neueingabe |

## Funktion: Produkt bearbeiten

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann ein bestehendes Produkt bearbeiten welches vorher ausgewählt werden muss |
| Input | * Produktname oder ID |
| Output | * Produktdaten des eingegebenen Produkts |
| Vorbereitungen | * Nutzer ist angemeldet * Produkt existiert |
| Nachbedingungen | * Überprüfung ob geänderte Produktdaten gültig |
| Fehlerfälle | * Produktname/ID nicht vorhanden --> Hinweis, Neueingabe * Produktdaten nicht gültig --> Hinweis, Daten betreffende Neueingabe |

## Funktion: Produkt löschen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann ein bestehendes Produkt löschen |
| Input | * Produktname oder ID |
| Output |  |
| Vorbereitungen | * Nutzer ist angemeldet * Produkt existiert |
| Nachbedingungen | * Überprüfen, ob Produkt existiert |
| Fehlerfälle | * Produkt existiert nicht --> Hinweis, Neueingabe |

## Funktion: Produktsuche

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann ein bestehendes Produkt suchen und sich die Daten anzeigen lassen. |
| Input | * Produktname oder ID oder ISBN/EAN |
| Output | * Produktdaten des Produkts, auf das die eingegebenen Daten zutreffen |
| Vorbereitungen | * Nutzer ist angemeldet * Produkt existiert |
| Nachbedingungen | * Überprüfen, ob Produkt existiert |
| Fehlerfälle | * Produkt existiert nicht --> Hinweis, Neueingabe |

## Funktion: Rechnung suchen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann eine bestehendes Rechnung suchen und sich die Daten anzeigen lassen. |
| Input | * ID und/oder isShipped und/oder isPaid und/oder date |
| Output | * Rechnung auf die die eingegebenen Daten zutrifft |
| Vorbereitungen | * Nutzer ist angemeldet * Rechnung wurde erstellt |
| Nachbedingungen |  |
| Fehlerfälle | * Rechnung wurde nicht gefunden --> Hinweis, Neueingabe |