VI. Функционални изисквания

# Указания

Копирайте този файлове във вашите директории, като пред името на файла добавите вашия факултетен номер. Прочетете всяка задача и отговорете на посоченото място.

# Задачи:

## Задача 1: Идентифициране на основни функционалности

Опишете **10** основни функционалности на системата от курсовия ви проект. Всяка функционалност трябва да описва **действие или поведение на системата**, а не нейно качество или характеристика.

При описването се вземете предвид следните насоки:

1. **Мислете от гледна точка на потребителя** – какви действия той ще може да извършва чрез системата.
2. **Използвайте глаголи** в описанията – например *„позволява“, „създава“, „проверява“, „изпраща“, „генерира“*.
3. **Не включвайте нефункционални характеристики** като „бърза“, „сигурна“, „интуитивна“ – освен, че това са нефункционални изисквания, често са и неправилни термини.
4. Всяка функционалност трябва да бъде **самостоятелна**, **ясно формулирана** и **измерима**.
5. Ако е възможно, групирайте сходните функционалности по подсистеми (напр. *„Управление на потребители“*, *„Отчети“*, *„Известия“*).

Списъкът ви с фунционални изисквания ще се променя и допълва при изпълнението на следващите задачи.

|  |  |
| --- | --- |
| № | **Изискване** |
| FR1 | Потребителя да може да търси, добавя, редактира, премахва игри. |
| FR2 | Потребителя да вижда резервациите. |
| FR3 | Потребителя да може да вижда броя на посетителите за деня. |
| FR4 | Потребителя да може да създава предстоящи събития. |
| FR5 | Потребителя да може да изпраща имейли към абонираните посетители на клуба |
| FR6 | Потребителя да има личен профил |
| FR7 | Следене на наличните напитки и храна |
| FR8 | Администратора да вижда activity log. |
| FR9 | Администратора да може да добавя нови потребители. |
| FR10 | Администратора да вижда продадените продукти |

## Задача 2: Формулиране на функционални изисквания в структурирана форма

Опишете 5 от функционалните изисквания от задача 1 в структуриран формат, следвайки шаблона по-долу.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Функционално изискване** | **Вход** | **Изход** | **Предусловия** | **Следусловия** |
| FR9 | Администратора да може да добавя нови потребители. | Въведени валидни данни | Връщане на съобщение за успешно регистриране | Няма потребител с посоченото име и имейл | Новият потребител е добавен |
| FR5 | Потребителя да може да изпраща имейли към абонираните посетители | Избран шаблон, списък с абонирани имейли | Изпратени имейли; съобщение за успешно изпращане | Потребителят е влязъл в системата и има абонатска база | Абонираните посетители получават съобщението |
| FR1 | Потребителят да може да търси, добавя, редактира и премахва игри | Име на игра, категория, жанр, брой играчи, продължителност | Актуализиран списък с игри; съобщение за успешно действие | Потребителят е влязъл в системата и има права за управление на игри | Новата/редактираната/изтритата игра е отразена в базата данни |
| FR4 | Потребителят да може да създава предстоящи събития | Име на събитието, дата, час, описание, лимит на участници | Потвърждение за създаване; събитие добавено в календара | Потребителят е влязъл в системата | Събитието се появява в списъка на предстоящи събития |
| FR3 | Потребителят да може да вижда броя на посетителите за деня | Текуща дата | Брой посетители за деня | Потребителят е влязъл в системата | Показва точен брой на регистрирани посетители за текущия ден |

## Задача 3: Откриване на неясни или противоречиви изисквания

По-долу са дадени няколко изисквания (някои добри, други с грешки). Посочете кои са неясни, непълни или противоречиви. Предложете поправена версия с ясно измерими критерии.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Оригинално изискване** | **Проблем** | **Поправена версия** |
| 1 | Системата трябва да зарежда страниците бързо. | Неясно „бързо“ | Системата трябва да зарежда продуктова страница за под 3 секунди |
| 2 | Клиентът трябва лесно да може да намери продукт. | Неясно „лесно“ | Системата трябва да позволява търсене на продукт чрез **ключова дума** и **филтри по категория, цена и наличност** |
| 3 | Поръчката се потвърждава автоматично, освен ако не е отказана от администратора. | Противоречие – не е ясно кога точно се потвърждава | Системата трябва да **потвърждава автоматично** поръчката **след успешно плащане,** освен ако **администраторът не я маркира като „отказана“** |
| 4 | Системата трябва да предлага най-добрите оферти. | Неясно | предлага **оферти с най-ниска цена** или **най-висока отстъпка** зададенакатегорияпродукт**.** |
| 5 | Резервацията може да се направи само ако има свободна стая. | Неясно | Резервацията може да се направи само ако има **свободна стая** за избраната дата. |
| 6 | Системата трябва да предлага лесна навигация между продуктите. | Неясно | Системата трябва да предоставя **основно меню и бутони** |

Проверете вашите функционални изисквания от задача 1. Редактирайте ги, ако намерите прилики между тях и изискванията от текущата задача.

## Задача 4: Дефиниране на зависимости между изисквания

По-долу имате списък от функционални изисквания на система, както и таблица на функционалните зависимости, в която за всяко изискване са описани неговите предходни. По подобен начин определете кои изисквания зависят от други за **вашите функционални изисквания** (например „FR5 не може да се реализира преди FR2“). Изобразете зависимостите в таблица, подобна на тази от примера. Обсъдете как тези зависимости влияят върху планирането на разработката.

**Пример:**

|  |  |
| --- | --- |
| **№** | **Изискване** |
| FR1 | Системата трябва да позволява регистрация на нов потребител. |
| FR2 | Системата трябва да позволява вход с потребителско име и парола. |
| FR3 | Потребителят може да търси книги по заглавие, автор или жанр. |
| FR4 | Потребителят може да заема книга онлайн. |
| FR5 | Системата трябва да изпраща имейл потвърждение при заемане на книга. |
| FR6 | Администраторът може да добавя и изтрива книги от каталога. |
| FR7 | Системата автоматично отбелязва просрочени книги и уведомява потребителя. |

**Таблица на функционални зависимости:**

|  |  |  |
| --- | --- | --- |
| **Изискване** | **Зависи от** | **Обяснение** |
| FR2 | FR1 | Входът е възможен само след регистрация. |
| FR4 | FR2, FR3 | За заемане е нужен влязъл потребител и избрана книга. |
| FR5 | FR4 | Имейлът се изпраща само при успешно заемане. |
| FR7 | FR4 | Просрочията възникват само след заемане на книга. |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Изискване** | **Зависи от** | **Обяснение** |
| FR1 | FR6 | Необходимо е профил за да може системата да знае кой потребител извършва действията |
| FR4 | FR6 | Изисква профил, за да се знае кой създава събитието. |
| FR7 | FR6, FR9 | Само администратор или потребител с права може да актуализира наличностите. |
| FR10 | FR7 | Продажбите на напитки и храна са част от данните, следени чрез FR7. |
| FR8 | FR6 | Да се проследи дадената смяна на служител |

## Задача 5: Приоритизация на функционалните изисквания по модела MoSCoW

MoSCoW е техника за приоритизация, която разделя изискванията в четири категории:

|  |  |  |
| --- | --- | --- |
| **Категория** | **Обозначение** | **Значение** |
| **Must have** | (M) | Задължителни изисквания. Без тях системата не може да функционира. Те формират **MVP** (минимален жизнеспособен продукт). |
| **Should have** | (S) | Много важни изисквания, но не критични за първата версия. Ако времето позволи, трябва да бъдат реализирани. |
| **Could have** | (C) | Полезни изисквания, които добавят стойност, но не са приоритетни. Реализират се само, ако има достатъчно ресурси. |
| **Won’t have (this time)** | (W) | Изисквания, които са извън обхвата на текущата версия, но може да се включат в бъдещи версии. |

Класифицирайте вашите функционалности по модела MoSCoW. Аргументирайте накратко защо сте му дали съответния приоритет.

**Пример:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Код** | **Изискване** | **Приоритет (M/S/C/W)** | **Обосновка** |
| FR1 | Регистрация на нов потребител | M | Без нея няма достъп до услугата. |
| FR2 | Търсене на ресторанти по местоположение | M | Основна функционалност. |
| FR3 | Възможност за оценка на поръчката | C | Полезна, но не е критична. |
| FR4 | Запазване на любими ресторанти | S | Удобство, подобрява потребителския опит. |
| FR5 | Плащане с дебитна карта | M | Основен метод за плащане. |
| FR6 | Плащане с криптовалута | W | Извън обхвата на първата версия. |
| FR7 | Push известия при доставка | S | Полезна, но може да се добави по-късно. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Код** | **Изискване** | **Приоритет (M/S/C/W)** | **Обосновка** |
| FR1 | Потребителя да може да търси, добавя, редактира, премахва игри. | M | Основна функция на системата – без нея няма управление на библиотеката с игри. |
| FR2 | Потребителя да вижда резервациите. | S | Важно за организацията, но системата може да работи и без този модул в първа версия. |
| FR3 | Потребителя да може да вижда броя на посетителите за деня. | C | Полезна справка, но не е критична за функционирането на системата. |
| FR4 | Потребителя да може да създава предстоящи събития. | C | Полезно за потенциално разрастване на клуба, но в по-нататъчен етап. |
| FR5 | Потребителя да може да изпраща имейли към абонираните посетители на клуба | C | Подпомага комуникацията, но може да се реализира на по-късен етап. |
| FR6 | Потребителя да има личен профил | M | Нужно за функциониране на цялата система |
| FR7 | Следене на наличните напитки и храна | M | Нужно за да се знае кога и колко продукти да се презаредят. |
| FR8 | Администратора да вижда activity log. | S | Важно за проследяване на дейностите в системата, но не е коренна функционалност. |
| FR9 | Администратора да може да добавя нови потребители. | M | Необходимо за управление на достъпа и поддръжката на системата. |
| FR10 | Администратора да вижда продадените продукти | M | Основна отчетна функция, нужна за следене на оборота и продажбите. |

## Задача 6: Тестване на функционалностите

Изберете 3 функционалности и опишете как бихте ги тествали, като включите очакван резултат, критерии за успех, др.

1. Функционалност FR1 – Потребителят да може да търси, добавя, редактира и премахва игри

Цел на теста: Да се провери дали системата правилно обработва действия за управление на игри (CRUD операции).

Стъпки за тестване:

1. Отворете екрана „Игри“.
2. Въведете ключова дума за търсене (напр. Monopoly).
3. Добавете нова игра с коректни данни.
4. Редактирайте съществуваща игра – промяна в описанието.
5. Изтрийте тестовата игра.

Очакван резултат:

* Търсенето показва точните резултати.
* Новата игра се появява в списъка.
* Редакцията се запазва успешно.
* Изтритата игра изчезва от списъка.

Критерий за успех: 100% от действията се извършват без грешка и базата данни се актуализира коректно.

2. Функционалност FR6 – Потребителят да има личен профил

Цел на теста: Да се уверим, че профилът може да бъде създаден, редактиран и достъпен само от собствения потребител.

Стъпки за тестване:

1. Създайте нов профил с валидни данни (име, имейл, парола).
2. Влезте с въведените данни.
3. Променете името и паролата.
4. Опитайте да отворите профила от друг акаунт.

Очакван резултат:

* Профилът се създава успешно.
* Входът е възможен само с правилна парола.
* Промените се запазват.
* Достъпът от друг акаунт е отказан.

Критерий за успех: Всички операции върху профила се изпълняват успешно, без нарушаване на защитата и личните данни.

3. Функционалност FR7 – Следене на наличните напитки и храна

Цел на теста: Да се провери дали системата правилно показва, актуализира и намалява наличностите при продажби.

Стъпки за тестване:

1. Отворете списъка с продукти.
2. Проверете текущите количества.
3. Регистрирайте продажба (напр. 2 напитки и 1 десерт).
4. Проверете дали количествата са намалели правилно.
5. Опитайте да продадете продукт с нулева наличност.

Очакван резултат:

* Количествата се актуализират веднага след продажба.
* Системата не позволява продажба на изчерпани продукти.
* Данните в отчета за продажби се обновяват автоматично.

Критерий за успех: Всички актуализации се извършват точно; няма несъответствия между наличности и продажби.