



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

Проект по дисциплината: Валидация и
Верификация на програмни системи

Тема: Портал за продажба на железопътни билети.

Студент – Владимир Данаилов Куцев

Факултет. - ФКСТ

Спец. - КСИ

Група 42 №121220150

Дата: 21/04/2024

Съдържание.

1. Функционални и не функционални изисквания.
2. Списък от Бизнес изисквания.
3. Процедура по инспектиране на кода.
4. Списък с откритите дефекти
5. Основен метод за изчисление на крайната цена за билет
6. Контролен списък за преглед на кода
7. Самостоятелен преглед на кода по използване на контролния списък

Функционални и не функционални изисквания.

1. Функционални изисквания

- *Закупуване на билет*
-Потребителя може да разглежда билети спрямо параметри като Време локация и други.
- *Динамично Ценообразуване*
-Цената на билета се определя спрямо специфичните условия: Време карта за отстъпка и други
- *Потребителски профили*
-потребителите могат да създават и управляват своите профили.
- *Система за управление на отстъпките*
-системата позволява управление на направената отстъпка.
- *Управление на резервираните билети*
-Билетите могат да биват премахвани или променяни преди тяхното изтичане.
- *Количка за пазаруване*
-потребителите могат да виждат закупените от тях билети както да променят, ако е възможно, техните параметри.

2. Не функционални изисквания

- *Използваемост*
-системата трябва да бъде лесна и интуитивна за ползване
- *Производителност*
-Системата трябва да реагира бързо и оптимално
- *Защита*
-Системата трябва да бъде сигурна и да защитава личните данни на потребителя
- *Мащабируемост*
-системата трябва да предвидена за увеличаване на нейния капацитет и възможности
- *Надеждност*

-системата трябва да осигурява висока достъпност и обработка на грешки, за да се гарантира непрекъсната работа на системата и целостта на данните

- *Поддръжка*

-кода и системната архитектура трябва да са проектирани за лесни актуализации и поддръжка

Списък от Бизнес изисквания.

1. *Предоставя ли системата цялостен инструмент за търсене на билети за влак?*
2. *Всички елементи на потребителския интерфейс интуитивни и достъпни ли са?*
3. *Системата прилага ли ефективно правила за динамично ценообразуване?*
4. *Спазени ли са стандартите за сигурност за защита на данните?*
5. *Мащабируема ли е системата, за да поеме бъдещото нарастване на броя на потребителите и обемите на транзакциите?*
6. *Има ли механизми за обратна връзка от потребителите и докладване на системни грешки?*

Процедура по инспектиране на кода.

Кога се обхожда изцяло от множество потребители(тестери) като се правят проверки по контролните списъци за наличие на функционалности и проблеми!

Списък с откритите дефекти

1. Установени проблеми

- Приложението не винаги успява да зареди запазения потребител в локалната памет от първия път
- При излизане от системата в някой от страниците, приложението спира работа.
- Приложението не винаги показва възникналите грешки по време на изпълнение на операции свързани с зареждане на информация от базата данни.

2. Предвидени решения на откритите проблеми

- Преработване на системата за управление и локализиране на запазен потребител в локалната памет.
- Добавяне на система за глобално визуализиране на възникналите грешки

Основен метод за изчисление на крайната цена за билет

```
export function calculateTicketPrice(  
  basePrice: number,  
  travelTime: string,  
  hasSeniorCard: boolean,  
  isTravelingWithChild: boolean,  
  hasFamilyCard: boolean  
): string {  
  // Convert travel time to minutes  
  const hour = parseInt(travelTime.split(':')[0]);  
  const minute = parseInt(travelTime.split(':')[1]);  
  const timeInMinutes = hour * 60 + minute;  
  
  // Determine if it is peak time  
  const isPeakTime =  
    (timeInMinutes >= 450 && timeInMinutes <= 570) || (timeInMinutes >= 960 && timeInMinutes <= 1170);  
  
  // Calculate base discount based on time  
  let discount = isPeakTime ? 0 : 0.05;  
  
  // Additional discount rules  
  if (hasSeniorCard && !isPeakTime) {  
    discount = 0.34; // Overrides the time-based discount if any  
  } else if (isTravelingWithChild) {  
    if (hasFamilyCard) {  
      discount = 0.50; // Overrides other discounts  
    } else {  
      discount = 0.10; // Overrides the time-based discount if any  
    }  
  }  
  
  // Calculate final price  
  const finalPrice = basePrice * (1 - discount);  
  return finalPrice.toFixed(2);  
}
```

Контролен списък за преглед на кода

1. Всички функции и променливи ясно ли са наименувани и дефинирани?
2. Кодът следва ли принципа DRY (не повтаряйте себе си)?
3. Има ли достатъчно коментари и документация?
4. Стабилно и изчерпателно ли е обработването на грешки?
5. Кодът отговаря ли на установените стандарти и указания за кодиране?

Самостоятелен преглед на кода по използване на контролния списък

1. Всички променливи са описателно именувани както и спазват стандарта "Camel case" за изписването си.
2. Кога се придържа към DRY принципа с някои леки изключения умишлено поставени за по добра четимост на кода.
3. Кога се старая по-добре да бъде обяснителен сам по себе си, но в необходимите моменти са налични достатъчно коментари които да обяснят принципа на действие и целта
4. Има какъвто да се желае от обработката на грешки.
5. Кога отговаря на установените стандарти и указания за писане на лесно четим и производителен код.