

ЗАДАНИЕ ЗА КУРСОВА РАБОТА

Дисциплина: *Софтуерни архитектури и разработка на софтуер*

I. Формат

Курсовата работа (КР) по дисциплината „Софтуерни архитектури и разработка на софтуер“ има за цел да проектирате и документирате софтуерната архитектура на система, изискванията за която са дадени в Раздел II настоящия документ. Предаването на готовите КР става през сайта на курса по САРС в Moodle. Предадени по електронната поща КР няма да се приемат. Нека всеки предаде самостоятелно курсовата работа независимо, че тя представлява работа в екип. Предава се само **pdf** версия на документа във файл с име **КРxx_2022_ffff_name**, където **xx** е номерът на избрания вариант, **ffff** е факултетния номер, а **name** е името на студента. За улеснение на изготвянето на КР в Moodle ще бъде качен шаблон със основните секции, които трябва да присъстват в КР, както и критериите за оценяване. **Не архивирайте предадения файл!**

Дейностите по проектирането на архитектурата са предмет на екипна работа, като за целта студентите се разделят на екипи от по **2 човека**. Презентации на курсовите работи ще се проведат по време на упражнения в последните две седмици на семестъра. Присъствието по време на презентациите за съответната група е задължително за всички екипи (т.е. не се допуска някои колеги да си тръгнат, след като са представили своята КР).

Описаната в КР архитектура трябва да съдържа документация на структурите, като е необходимо документирането на **три** структури – декомпозиция на модулите и още **две** по избор. Като част от текста на курсовата работа, задължително се дава аргументирана обосновка на избора на двете допълнителни структури.

II. Описание

Да се проектира софтуерната архитектура на софтуерната система *ExpenseBuddy*, предназначена за управление на разходи. Системата позволява на потребителите да създават индивидуални или споделени бюджети за от 2 до 5 души. Основната цел на системата е потребителите да имат възможност да съхраняват информация за направени разходи за определен период от време и да получават анализи на база направените разходи.

III. Изисквания към системата

1. Достъпът до системата трябва да може да се осъществява или през браузър, или чрез мобилен клиент за iOS и Android.
2. Системата поддържа 2 типа потребители:
 - Обикновени потребители, които могат да използват ограничен набор от функционалностите на системата.
 - Привилегировани потребители, които имат достъп до всички функционалности на системата
3. Потребител може да се регистрира в системата чрез имейл, потребителско име и парола, или чрез връзка с външна система. Потребител може да се впише в системата чрез имейл или потребителско име и парола, или чрез външна система.
 - a. Възможни външни системи са например Google и Facebook.
 - b. Системата трябва да предоставя възможност за добавяне на допълнителни външни системи.
4. Бюджетът представлява съвкупност от разходи. Всеки разход съдържа следната информация:
 - a. списък на закупените продукти/услуги и цена за всеки от тях
 - b. търговец
 - c. дата на разхода
 - d. категория
 - e. начин на плащане
 - f. опционално: при споделен бюджет се вписва автоматично името на потребителя, въвел разхода
 - g. опционално: бележки от потребителя
5. Системата трябва да поддържа два типа бюджет:
 - a. Индивидуален - право на достъп има само потребителят, който е създал бюджета.
 - b. Споделен - създава се от един потребител и има възможност да добавя към него до 4 нови потребителя.
 - i. Добавяне на потребител към споделен бюджет се случва чрез изпращане на имейл покана.

6. Всеки потребител може да създава бюджети или да се присъедини към създаден от друг потребител споделен бюджет.
7. Нов разход може да бъде добавен по следните начини:
 - a. чрез снимка на касов бон: Потребител може да направи или качи снимка на касов бон и системата автоматично ще попълни формата за разход с данни от касовия бон, а на база на търговеца приел плащането, системата определя към коя категория той принадлежи. След потвърждение от страна на потребителя, разходът бива добавен към избория от него бюджет
 - b. чрез връзка с онлайн система за плащания (пример PayPal, Revolut, Skrill, банкови системи) - Потребител може да свърже бюджета си с онлайн системи за плащания. В този случай при извършване на плащане, потребителят ще получава известие или имейл, чрез който може да потвърди плащането и да го добави нов разход към бюджета на база плащането. Категорията на разхода се определя от системата на база търговеца получил плащането.
 - c. чрез ръчно въвеждане на разход - Потребителят може да попълни форма за информация за разхода и да го добави към избория от него бюджет.
8. Обикновените потребители ще имат достъп до секцията Отчети, в която ще могат да виждат всичките си бюджети и наличните разходи за тях. Обикновен потребител може да премине към Премиум при заплащане на месечен абонамент.
9. Премиум потребителите ще имат достъп до допълнителна секция Анализи. В тази секция са налични анализи за разходите по даден бюджет за даден период:
 - a. Премиум потребители могат да избират период за анализ.
 - b. Премиум потребителите могат да филтрират по категория разход и/или начин на плащане. Наличните категории могат да са: храна, услуги, развлечение, разни и др.
 - c. Премиум потребителите ще получават известия, ако има необичайна активност спрямо разходите за даден бюджет. (Пример: разходите в категория храна са увеличени с 30% спрямо предходния месец.)
10. Възможни са опции за плащане с банкова карта, Revolut, Paypal и Skrill.
11. Данните в системата трябва да бъдат защитени от нерегламентиран достъп. Особено важно е комуникацията между системата и външните системи за разплащане да предоставя защита на пренасяните данни.
12. Допуска се профилактика веднъж месечно в рамките на 6 часа. През останалото време, системата трябва да е 99,95% налична.
13. При извършване на плащане чрез външна система, известието за направеното плащане трябва да достига до потребителя в рамките на 30 сек.

14. С цел актуалност на информацията, генерирането на агрегираните анализи трябва да става до 3 секунди.
15. Всеки потребител трябва да има възможност да изтрива данни от системата, асоциирани с него по всяко време спрямо GDPR.
16. Архитектурата трябва да позволява лесно добавяне на нови системи за плащания.