Софтуерни архитектури и разработка на софтуер
Домашна работа № 1
Вариант 1 – ExpenseBuddy

Стефан Велев, № 62537

Даниел Халачев, № 62547

Функционални изисквания	***Нефункционални изисквания***
№2, №3, №4, №5, №6, №7, №8, №9, №15	*№1 (Достъпност),* *№10 (Технологично ограничение),* *№11 (Сигурност),* *№12 (Наличност),* *№13 (Производителност),* *№14 (Производителност),* *№16 (Изменяемост)*

Изискване[1]	***Драйвер***	***Обосновка***
1. *Достъпът до системата трябва да може да се осъществява или през браузър, или чрез мобилен клиент за iOS и Android*	*Да*	*Достъпът до системата "чрез мобилен клиент" означава разработка на приложение, специфично за конкретна операционна система (в случая две ОС) - iOS и Android. Това предполага големи различия в кода за различните варианти на системата и съответно разделянето на два модула за кода на всяка мобилна версия и трети модул за браузър клиента. Тази декомпозиция на модулите би позволила възможно най-лесна промяна на кода във възможно най-малко на брой модули.*
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*	*Да*	*Достъпът до различната функционалност за различните видове потребители се определя от системата за администриране на акаунтите (част от нашата система, обособена в собствен модул). Предпоставка за съществуването на този модул не е само това изискване, а и изискването за регистрация и вписване (**3**).*
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.*	*Да*	*- Регистрацията и вписването са важни функционалности на софтуера, които осигуряват неговата сигурност, пълноценно потребителско изживяване и осъществяване на бизнес целите на проекта. Затова, с цел максимално разделение на функционалностите и максимална защита от злонамерени действия, е необходимо обособяването на модул за администриране на акаунтите (който от своя страна също би могъл да бъде разделен на два подмодула - за регистрация и вписване съответно).*

[1] Изискванията са запазени във формата, даден в условието, с цел прегледност и по-лесна проверка, въпреки че много от тях съдържат повече от едно изискване в себе си.

a. *Възможни външни системи са например Google и Facebook.* ***b.*** *Системата трябва да предоставя възможност за добавяне на допълнителни външни системи.*		*- В допълнение, използването на библиотеки от външни системи трябва да е отделено от собствения код на системата.* *- В заключение, изискването за добавяне на допълнителни външни системи също налага такава декомпозиция на модулите*
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*	*Не*	*- Така дефинирано, това изискване влияе пряко върху структурите от данни на системата и тяхното съдържание, но не и на архитектурата ѝ.* *- Влиянието върху архитектурата е косвено. Съхранението на тази информация ще се извършва в база от данни. Редно е системата да има модул, който отговаря за работата с нея. Освен това, базата данни трябва да е налична за всички клиенти. Подходящ архитектурен стил, който осигурява това свойство, е "Клиент-Сървър" архитектурата. За документирането на разположението на базата данни е необходима структурата на разположението (кой софтуер на коя хардуерна единица се намира), която пряко е обвързана със софтуерната архитектура.*
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.*	*Не*	*- Поддържането на два различни типа бюджети е функционално изискване, което не оказва пряко влияние върху архитектурата, т.е. не може да бъде определено като архитектурен драйвер.* *- Изпращането на имейл покана само по себе си не е достатъчно изискване и така формулирано не може да бъде обособено самостоятелно в отделен модул.*
6. *Всеки потребител може да създава бюджети или да се присъедини към създаден от друг потребител споделен бюджет*	*Не*	*Това изискване не води до добавянето на нов модул, а само до промяна на имплементацията на модула за управление на бюджети. Не е необходимо добавянето или изменянето на някой съществуващ архитектурен изглед.*
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 избрания от него бюджет* ***b.*** *чрез връзка с онлайн система за плащания (пример PayPal,*	*Да*	*- Това изискване оказва влияние върху софтуерната архитектура, защото функционалности като качване на снимка, обработка (разчитане) на снимката и плащане са сложни изчислителни операции (някои от които изискват технологии като изкуствен интелект) и затова е необходимо въвеждането на отделен модул за всяка от тях.* *- Комуникацията с други онлайн системи се осъществява обикновено чрез използването на вече съществуващи външни платформи. Лоша практика е смесването на собствени библиотеки с външни за приложението библиотеки. Необходимо е обособяването на отделен модул, а също и по-детайлно представяне в структурата на процесите.*

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 избрания 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.*	*Не*	*Модул за плащанията вече е обособен вследствие заради други (по-важни) функционални изисквания. Не се добавя нищо ново по отношение на архитектурата в сравнение с предишните изисквания.*
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.*	*Да*	*- Анализите на потребителите ще се извършват на основата на техните данни, които трябва да се съхраняват в база данни, поместена на определен хардуер. Това влияе на структурата на разположението. В допълнение, необходим е модул за СУБД. Така в бъдеще при добавянето на ново изискване (например за надеждност) ще може лесно да се осъществи измяната на този модул или дублирането му (например тактиката за предотвратяване на откази Active Redundancy).* *- Наличието на допълнителна секция Анализи , където се извършват сложни изчислителни операции, налага наличието на отделен модул за това в структурата на модулите.* *- Изпращането на известия на потребителите също е независима функционалност, за която е необходим отделен модул в структурите на модулите.*
10. *Възможни са опции за плащане с банкова карта, Revolut, Paypal и Skrill.*	*Не*	*Поради вече дефинирани архитектурни драйвери системата има модул, отговарящ за разплащанията. Приема се, че това изискване не е архитектурен драйвер, но със забележката, че е необходимо модулът за разплащанията да бъде декомпозиран на още няколко подмодула (поради тези нови възможности за плащане) и това да бъде отразено в представянето на Модулната структура.*
11. *Данните в системата трябва да бъдат защитени от нерегламентиран достъп. Особено важно е*	*Да*	*- За системата ExpenseBuddy сигурността е от изключително важно значение, поради съхранението на множество чувствителни данни. За пълноценната защита на тези данни е необходимо криптиране, както и*

комуникацията между системата и външните системи за разплащане да предоставя защита на пренасяните данни		*инсталиране на допълнителни защитни механизми, които трябва да бъдат предвидени в процеса на създаване на архитектурата.* *- Възможно е в процеса на разработка това изискване да отпадне като функционален драйвер, защото криптирането може да се имплементира уникално за всеки модул (разнородност в проектирането), а нерегламентиран достъп се предотвратява от системата за администриране на акаунтите*
***12.** Допуска се профилактика веднъж месечно в рамките на 6 часа. През останалото време, системата трябва да е 99,95% налична.*	*Да*	*Високият процент надеждност може да наложи дублирането на модули (Active/Passive Redundancy), което трябва да бъде отразено в декомпозицията на модулите и структурата на разположението.*
***13.** При извършване на плащане чрез външна система, известието за направеното плащане трябва да достига до потребителя в рамките на 30 сек.*	*Не*	*Количествените параметри на резултата "изпращане на известие" зависят от имплементацията на тази конкретна функционалност. Няма връзка с други функционалности и процеси в системата. Имплементацията конкретно на една функционалност не оказва влияние на архитектурата.*
***14.** С цел актуалност на информацията, генерирането на агрегираните анализи трябва да става до 3 секунди.*	*Не*	*Изискването за генериране на анализи обуславя наличието на модул, а не изискването генерирането да става за 3 секунди. Това зависи от имплементацията на този вече дефиниран модул.*
***15.** Всеки потребител трябва да има възможност да изтрива данни от системата, асоциирани с него по всяко време спрямо GDPR.*	*Не*	*Сигурността на личните данни означава внимателно складиране на информацията в системата и постоянно надграждане. Изтриването на информацията от потребител ще се имплементира в модула за управление на акаунтите, който е дефиниран в изискване **(3)**.*
***16.** Архитектурата трябва да позволява лесно добавяне на нови системи за плащания.*	*Да*	*Изискването налага да има отделен модул, обединяващ всички системи за плащания, за да могат промените да стават само в него.*