

Aumo - дигиталните касови бележки на бъдещето

Симо Александров Любо Любчев

Абстракт

Тонове касови бележки биват създадени и веднага изхвърлени, като за изработката им се използва химикал вреден за човека, заедно с дървесна маса, което означава, че хиляди декари гори биват отсичани годишно и мнозина се разболяват.

Aumo е мобилно приложение, придруженено с хардуерно устройство и уеб съвръп, което цели да премахне хартиените касови бележки, като ги замести с дигитални. Касовите апарати на магазини и заведения ще бъдат оборудвани с Aumo. Клиентите ще получат техните дигитални касови бележки при допира на тяхното мобилно устройство (през мобилното ни приложение) с Aumo чрез NFC (Near-Field Communication) технология. За мотив да се използва дигиталната касова бележка пред хартиения еквивалент, потребителите ще бъдат възнаграждавани с точки, всеки път когато клиентът предпочете Aumo пред традиционната касова бележка. Тези точки могат да бъдат използвани за бонуси под формата на намаления или материални награди, осигурени от търговския обект.

Проектът е с приложен характер, все още е в процес на разработка и е от сферата на информатика и информационни технологии. Идеята е измислена от Симо Александров, а е реализирана от двамата автори.

Tons of paper receipts are produced and then immediately thrown away, for the creation of which are used trees and a human toxic chemical, requiring thousands of decades of forests to be cut down and resulting in spreading of diseases.

Aumo is a mobile application, accompanied by a hardware device and a web server, which aims at removing paper receipts by replacing them with a digital equivalent. Receipt printers of shops and restaurants will be equipped with Aumo. Clients will take their digital receipts by approaching their phone (through our mobile application) to Aumo, establishing a connection via NFC (Near-Field Communication) technology. Incentive for using the digital receipt, as opposed to the paper alternative, will be points which users receive when choosing Aumo over the traditional option. Points can be exchanged for bonuses, which can either be discounts or physical items, provided by the shop or restaurant.

The project is of applicational nature, it is still under development and belongs to the IT field. The idea was conceived by Simo Aleksandrov and was realised by both of the authors.

Съдържание

Увод	3
Галерия	4
Функции	5
Как работи	5
Мотив (Геймификация)	5
Технологии	6
Backend	6
go	6
go-chi	6
MariaDB (MySQL)	6
GORM	6
Redis	7
Raspberry Pi	7
Docker	8
Frontend	8
React и React-Native	8
Етапи на развитие	9
Избор на тема	9
Проучване	10
Избиране на технологии и архитектура	10
Изработване	10
Описание на приложението	10
Заключение	10
Бъдеще и развитие	11

УВОД

За изработката на касови бележки се използват множество ресурси. Някои, от които включват:

- ВПА (Bisphenol A / Бисфенол А)
- Дърво

Първите, от които са токсични за човешката кожа, Бисфенол А, може да доведе до заболявания като рак, захарен диабет тип 2, наднормено тегло и други. Тонове дървета биват отсичани за създаването на хартията на касовите бележки. Статистики показват, че се отисчат 60 000+ декара гори годишно само от "Толемите 5 държави". Премахването на тези ресурси ще се подпомогне, както и на хората живуещи на нашата планета, така и на самата планета.

С тази разработка целим заменянето на хартиените касови бележки с дигитална алтернатива. За да постигнем тази цел трябва да бъдат решени следните задачи:

- Планиране на архитектура и подбор на правилните технологии
- Създаване на устройство, което ще играе ролята на посредник между касови апарати и мобилното приложение
- Разработка на backend сървър
- Оформяне на красив и лесен за използване графичен интерфейс
- Мобилно приложение за клиентите на заведения или магазини
- Мотив за потребителите (система за награди и точки - Gamification)
- Административен панел за добавяне на награди
- Рекламиране

Галерия



Фигура 1: Мобилното приложение на iPhone X



Фигура 2: Устройството



Фигура 3: Логото на Aumo

Функции

Приложението ни предоставя следните функции:

- Светкавична бързина
- Изпращане на информация (дигитална касова бележка) чрез NFC (Near-Field Communication)
- Виртуален магазин за промоции/награди предоставени от търговския обект
- Потребителски панел (Мобилно приложение)
- Административен панел за управление на виртуалния магазин
- История от всички касови бележки
- Създаване на списък с разходите извършени за деня/седмицата/месеца съответно

Как работи

Между всеки касов апарат и компютъра, свързан с него, ще бъде поставено по едно устройство - Auto. Auto представлява малък компютър (в случая Raspberry Pi), поставен в кутийка (изработена в случая от 3D принтер). Устройството ще играе роля на посредник и ще приема нужната информация от компютъра и в зависимост от избора на клиента, касовия бон ще бъде изпратен към принтера или към NFC модул (в случаите, когато клиента няма телефон, той може по да изисква своя касов бон по традиционния начин). Към този компютър - Auto е свързан с NFC модул, чрез който той получава информация за касовия бон от компютъра (POS системата), а мобилното приложение ще получава касовата бележка чрез допир до устройството по NFC. Потребителите ще могат да допрат тяхното мобилно устройство до Auto, използвайки мобилното ни приложение и касовия бон ще бъде добавен в техния профил.

Мотив (Геймификация)

При всяко използване на Auto клиента получава точки, те могат да бъдат обменени за бонуси, промоции или предметни награди предоставени от търговския обект, като по този начин клиентите биват мотивирани да използват нашето приложение.

Досега всичко е било “на хартия”. Този подход с дигитализация на нещо толкова битово ще изисква много усилия, тъй като ние хората обичаме да стоим в комфортната си зона и често не обичаме промяна. Затова ние решихме да вкараем геймификация в Auto, като по този начин клиентите ще бъдат мотивирани да използват дигитална касова бележка пред традиционната - хартиена, както и да посещават по често съответния търговски обект предлагаш услугата. Така усилията, които трябва да бъдат положени, както от страна на клиенти, така и от страна на собствениците на търговските обекти няма да се усещат и ще бъдат забавни.

Технологии

Backend

go

Като език за програмиране използвахме go, тъй като е бърз, гъвкав, лесен за писане и разбиране и може да се компилира към всички операционни системи - macOS, Linux, Windows.

go-chi

Като библиотека за HTTP сървър използвахме go-chi, поради факта че е тънък слой (wrapper) над стандартната библиотека на go - net/http. Предоставя лесна абстракция за създаване на REST API. Малък пример за сървър:

```
package main

import (
    "net/http"
    "github.com/go-chi/chi"
)

func main() {
    r := chi.NewRouter()
    r.Get("/", func(w http.ResponseWriter, r *http.Request) {
        w.Write([]byte("welcome"))
    })
    http.ListenAndServe(":3000", r)
}
```

MariaDB (MySQL)

Използвахме MySQL (или по-конкретно MariaDB) като база от данни, поради широкото ѝ разпространение в индустрията. Лесна е за използване и конфигуриране.

GORM

Като библиотека за свързване и абстракция от go към MySQL, се спряхме на GORM. Много бързо и лесно успяхме да създадем нашите модели. Само с няколко реда код, ние можем да имаме потребители в нашата база от данни.

```

type User struct {
    gorm.Model
    Name      string      `json:"name" gorm:"not null"`
    Email     string      `json:"email" gorm:"unique;not null"`
    Password  string      `json:"-" gorm:"not null" gob:"-"`
    Avatar    string      `json:"avatar" `
    Points   float64     `json:"points" gorm:"not null"`
    Orders   []ShopItem  `json:"orders" gorm:"many2many:user_shop_item;"`
    Receipts []Receipt   `json:"receipts"`
}

func (a *Aumo) CreateUser(u User) (User, error) {
    if err := a.db.Create(u).Error; err != nil {
        return nil, err
    }

    return u, nil
}

```

Redis

За сесии, кепиране за MySQL заявки използвахме Redis заедно с go-redis.

Raspberry Pi

За устройството използахме Raspberry Pi, поставено в 3D принтирана кутийка, направена в CAD системата Solidworks. Кутията беше принтирана в Русенския Университет, тъй като те разполагат с 3D принтер. (виж фиг. 4 и 5)



Фигура 4: Raspberry Pi



Фигура 5: Устройството

Docker

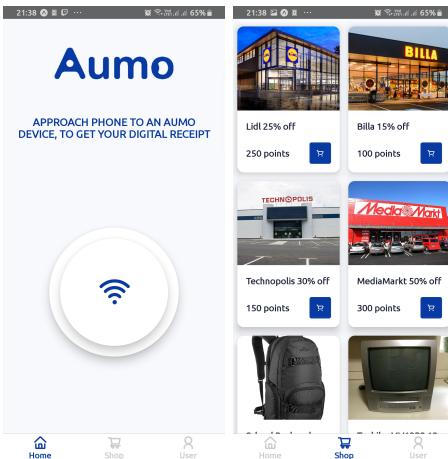
За deployment (публикуване) на нашото приложение, използвахме технологията за контейнери, по-конкретно - Docker. Той ни предоставя еднакъв environment, независимо от операционната система, дистробуция или други фактори. Също така ни улеснява scaling (скалиране) на много истанции.

Frontend

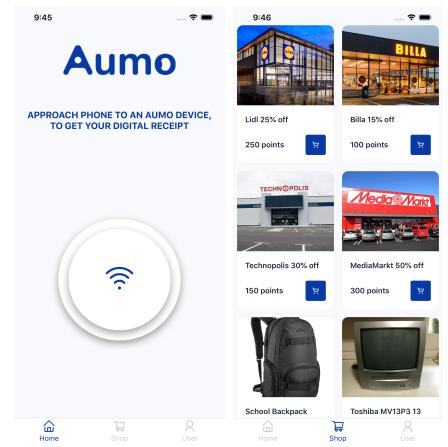
React и React-Native

За да създадем хубав и гъвкав интерфейс, заедно с reusable компоненти използвахме React - библиотека създадена от Facebook. За мобилното при-

ложение използвахме по-конкретно React-Native, тъй като можем да пишем един код за всички платформи - iOS и Android. (виж фиг. 6 и фиг. 7)



Фигура 6: Мобилното приложение на Android



Фигура 7: Мобилното приложение на iOS

Етапи на развитие

Избор на тема

Двамата автори бяхме в Стара Загора (участвахме на състезание), седнахме да обядваме. Докато Симо беше на опашката, забелязва, че зад касите стояха кофи за боклук преливащи от касови бележки, които така и не влизат в употреба а само се изхвърлят на вятъра един природен ресурс. Така решихме да измислим решение, с което можем да сложим край на този проблем възможно най-скоро - роди се идеята за Aumo.

Проучване

При установено проучване от нас - Любо и Симо, не успяхме да открием подобни решения, действащи в момента на пазара. Така се убедихме, че е време да започнем работа върху бъдещият ни продукт - Aumo.

Избиране на технологии и архитектура

През този етап ни минаха доста идеи относно подхода ни с технологиите, като се спряхме на вече гореспоменати. (Виж сек. Технологии).

Изработка

Започнахме работа върху проекта, по време на Русенския хакатон (TeenHack Ruse 2019) провел се в началото на октомври. - Тогава успяхме да създадем REST API написан на Go чрез, който извършаме CRUD операции. - Изградихме базова концепция за нашия графичен интерфейс - Figma - Създадохме нашето мобилно приложение с React Native - Свързахме мобилното приложение със сървъра ни

Описание на приложението

Сорс кода на проекта може да бъде намерен в GitHub¹. След като е изтеглен проекта, той може да бъде конфигуриран много лесно, благодарение на Docker.

```
docker build -t aumo .
docker run aumo
```

Мобилното приложение може да бъде стартирано чрез npm или yarn

```
npm run start
# or
yarn start
```

След като са стартирани, ще може да бъде намерено в Expo на вашето мобилно устройство, а сървъра на порт 3000.

Заключение

Aumo - дигиталните касови бележки на бъдещето ще спаси тонове хартия, като съответно ще бъдат запазени хиляди декари гори и спасяването на потенциално изчезващи живонски видове (тези, които живеят по дърветата). Предполага се, че търговските обекти ще започнат да печелят повече благодарение на геймификацията включена в Aumo. Хората ще спрат да губят своите касови бележки и ще могат лесно и бързо да си правят отчет за деня или седмицата къде какви пари отиват. Намалява се рискът за различни

¹<https://github.com/DeliriumProducts/aumo>

заболявания, като например рак, и диабет, тъй като ВРА вече не е фактор в този вид бележки.

Въпреки предизвикателността на поставените от нас задачи, ние успяхме да преодолеем почти всички. Подбрахме правилната за нуждите ни технология, която да може бързо да се справя с поставените от нас задачи.

Приложението е все още в процес на разработка, като към момента съществува само базов прототип, но се очаква до края на декември 2019 год. то да бъде почти завършено.

Бъдеще и развитие

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.