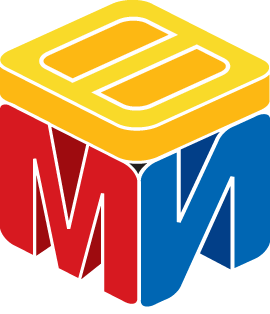
  
 **Пловдивски университет „Паисий Хилендарски”**  
 Факултет по Математика и Информатика   
 Катедра „Компютърни технологии“

  
Дипломна работа

Тема  
Мобилно приложение - Дневник за автомобила „InfoCar“

|  |  |
| --- | --- |
| **Дипломант:**  *Желязко Георгиев Георгиев*  *фак. № 1601261076*  *специалност: „Информатика“* | **Научен ръководител:**  *гл. ас. д-р Мая Стоева* |

**Пловдив, 2020 г.**

# Съдържание

[Съдържание 2](#_Toc526146652)

[Списък с фигури 5](#_Toc526146653)

[Речник на използваните термини](#_Toc526146654) 6

[Увод 7](#_Toc526146655)

[Цели на дипломната работа 8](#_Toc526146656)

[Структура на дипломната работа 9](#_Toc526146657)

[Глава 1. Съществуващи платформи в уеб пространството 10](#_Toc526146658)

[1.1 Drivvo 11](#_Toc526146659)

[1.2 Pinterest 12](#_Toc526146660)

[1.3 Polycount 13](#_Toc526146661)

[1.4 Сравнителен анализ 14](#_Toc526146662)

[Глава 2. Технологии и средства за разработка 16](#_Toc526146663)

[2.1 HTML 5 16](#_Toc526146664)

[2.2 CSS 3 17](#_Toc526146665)

[2.3 JavaScript 17](#_Toc526146666)

[2.4 jQuery 18](#_Toc526146667)

[2.5 Ajax 18](#_Toc526146668)

[2.6 PHP 18](#_Toc526146669)

[2.7 phpMyAdmin 19](#_Toc526146670)

[2.8 Composer 19](#_Toc526146671)

[2.9 Laravel 20](#_Toc526146672)

[2.10 Eloquent ORM 21](#_Toc526146673)

[2.11 Apache 21](#_Toc526146674)

[2.12 MySQL 21](#_Toc526146675)

[2.13 Atom 21](#_Toc526146676)

[2.14 Adobe Photoshop CC 2018 22](#_Toc526146677)

[Глава 3. Описание на системата 23](#_Toc526146678)

[3.1 Изисквания 23](#_Toc526146679)

[3.1.1 Ефикасност и интуитивна използваемост 23](#_Toc526146680)

[3.1.2 Модерен и атрактивен дизайн 23](#_Toc526146681)

[3.1.3 Адаптивност 23](#_Toc526146682)

[3.1.4 Бързина 24](#_Toc526146683)

[3.1.5 Гъвкавост и възможност за разширение 24](#_Toc526146684)

[3.1.6 Оптимизация за търсачките 24](#_Toc526146685)

[3.1.7 Оптимизирана база данни 24](#_Toc526146686)

[3.2 Същност на приложението 25](#_Toc526146687)

[3.2.1 Системна среда 25](#_Toc526146688)

[3.2.2 Навигационно меню 26](#_Toc526146689)

[3.2.3 Начална страница 27](#_Toc526146690)

[3.2.4 Страница за регистрация 31](#_Toc526146691)

[3.2.5 Страница за вход 33](#_Toc526146692)

[3.2.6 Страница с информация за определено произведение 34](#_Toc526146693)

[3.2.7 Страница с категории 37](#_Toc526146694)

[3.2.8 Страница с албуми 38](#_Toc526146695)

[3.2.9 Страница с информация за определен албум 39](#_Toc526146696)

[3.2.11 Потребителска страница 43](#_Toc526146697)

[3.2.12 Страница за настройки на профила 44](#_Toc526146698)

[3.2.13 Администраторска страница 46](#_Toc526146699)

[Глава 4. Ръководство за програмиста 49](#_Toc526146700)

[4.1 Архитектура на системата 49](#_Toc526146701)

[4.2 Архитектура на базата данни 51](#_Toc526146702)

[4.2.1 Albums 51](#_Toc526146703)

[4.2.2 Album\_images 52](#_Toc526146704)

[4.2.3 Comments 53](#_Toc526146705)

[4.2.4 Images 53](#_Toc526146706)

[4.2.5 Migrations 54](#_Toc526146707)

[4.2.6 Roles 55](#_Toc526146708)

[4.2.7 Role\_user 55](#_Toc526146709)

[4.2.8 Taggables 55](#_Toc526146710)

[4.2.9 Tags 56](#_Toc526146711)

[4.2.10 Users 56](#_Toc526146712)

[4.2.11 Votes 57](#_Toc526146713)

[4.2.12 EER диаграма и релации 58](#_Toc526146714)

[Глава 5. Анализ, SEO, домейн име, хостинг, тестване, критични точки и усъвършенстване 59](#_Toc526146715)

[5.1 Анализ 59](#_Toc526146716)

[5.2 SEO 60](#_Toc526146717)

[5.3 Домейн име 62](#_Toc526146718)

[5.4 Хостинг 63](#_Toc526146719)

[5.5 Тестване 63](#_Toc526146720)

[5.6 Критични точки 64](#_Toc526146721)

[5.7 Усъвършенстване 65](#_Toc526146722)

[Авторска справка 66](#_Toc526146723)

[Заключение 67](#_Toc526146724)

[Използвана литература 68](#_Toc526146725)

# Списък с фигури

[Фигура 1. Drivvo 12](#_Toc526146726)

[Фигура 2. Drivvo 13](#_Toc526146727)

[Фигура 3. Simply Auto 14](#_Toc526146728)

[Фигура 4. Simply Auto 18](#_Toc526146729)

[Фигура 5.](file:///C:/Users/Bozhidar/Desktop/Diploma/1401261107_Bozhidar_Kabzamalov_IRO_260918.docx" \l "_Toc526146730) Fuelio 21

[Фигура 6. Пример за медийна заявка 25](#_Toc526146731)

[Фигура 7. Use case за нерегистриран потребител 26](#_Toc526146732)

[Фигура 8. Use case за регистриран потребител 26](#_Toc526146733)

[Фигура 9. Use case за администратор 27](#_Toc526146734)

[Фигура 10. Навигационно меню за нерегистриран потребител 28](#_Toc526146735)

[Фигура 11. Навигационно меню за регистриран потребител 28](#_Toc526146736)

[Фигура 12. Навигационно меню за администратор 28](#_Toc526146737)

[Фигура 13. Адаптивен дизайн на навигационното меню 28](#_Toc526146738)

[Фигура 14. Падащо меню 28](#_Toc526146739)

[Фигура 15. Начална страница за нерегистриран потребител 29](#_Toc526146740)

[Фигура 16. Начална страница за регистриран потребител 29](#_Toc526146741)

[Фигура 17. Функция за гласуване 30](#_Toc526146742)

[Фигура 18. Филтър по време - днес 31](#_Toc526146743)

[Фигура 19. Switch който връща променлива с изображения 31](#_Toc526146744)

[Фигура 20. Адаптивен дизайн на началната страница 31](#_Toc526146745)

[Фигура 21. Страница за регистрация 32](#_Toc526146746)

[Фигура 22. Функция за регистрация 33](#_Toc526146747)

[Фигура 23. Страница за вход 34](#_Toc526146748)

[Фигура 24. Функция за вход 34](#_Toc526146749)

[Фигура 25. Адаптивен дизайн](file:///C:/Users/Bozhidar/Desktop/Diploma/1401261107_Bozhidar_Kabzamalov_IRO_260918.docx" \l "_Toc526146750) 35

[Фигура 26. Страница с информация за определено произведение 36](#_Toc526146751)

[Фигура 27. Функция за изтриване на изображение 37](#_Toc526146752)

[Фигура 28. Функция за нов коментар 37](#_Toc526146753)

[Фигура 29. Страница с категории 38](#_Toc526146754)

[Фигура 30. Адаптивен дизайн на страница с категории 39](#_Toc526146755)

[Фигура 31. Страница с албуми 39](#_Toc526146756)

[Фигура 32. Адаптивен дизайн на страница с албуми 40](#_Toc526146757)

[Фигура 33. Страница с информация за определен албум 41](#_Toc526146758)

[Фигура 34. Функция за изтриване на албум 42](#_Toc526146759)

[Фигура 35. Страница за качване на файлове 43](#_Toc526146760)

[Фигура 36. Функция за качване на единичен файл 44](#_Toc526146761)

[Фигура 37. Потребителска страница 45](#_Toc526146762)

[Фигура 38. Страница за настройки на профила 46](#_Toc526146763)

[Фигура 39. Функция за актуализация на профила 47](#_Toc526146764)

[Фигура 40. Администраторска страница 48](#_Toc526146765)

[Фигура 41. Функция за създаване на категория 49](#_Toc526146766)

[Фигура 42. Функция за изтриване на категория 49](#_Toc526146767)

[Фигура 43. Взаимодействията в MVC модела 50](#_Toc526146768)

[Фигура 44. База данни 52](#_Toc526146769)

[Фигура 45. Таблица albums 53](#_Toc526146770)

[Фигура 46. Таблица album\_images 54](#_Toc526146771)

[Фигура 47. Таблица comments 54](#_Toc526146772)

[Фигура 48. Таблица images 55](#_Toc526146773)

[Фигура 49. Таблица migrations 55](#_Toc526146774)

[Фигура 50. Таблица roles 56](#_Toc526146775)

[Фигура 51. Таблица role\_user 56](#_Toc526146776)

[Фигура 52. Таблица taggables 57](#_Toc526146777)

[Фигура 53. Таблица tags 57](#_Toc526146778)

[Фигура 54. Таблица users 58](#_Toc526146779)

[Фигура 55. Таблица votes 58](#_Toc526146780)

[Фигура 56. EER диаграма и релации на базата данни 59](#_Toc526146781)

[Фигура 57. Google Analytics първа част 60](#_Toc526146782)

[Фигура 58. Google Analytics втора част 61](#_Toc526146783)

[Фигура 59. GTmetrix първа част 62](#_Toc526146784)

[Фигура 60. GTmetrix втора част 62](#_Toc526146785)

[Фигура 61. GTmetrix трета част 63](#_Toc526146786)

[Фигура 62. Browserstack 65](#_Toc526146787)

# Речник на използваните термини

**Google I/O (Google Input/Output или Google In the Open) –** Годишна конференция за програмисти, провеждана от Google в Mountain View, Калифорния.

**MVC (Model-View-Controller)** – Модел-Изглед-Контролер архитектурен модел за изграждане на приложения;

**SEO** – Процесът на подобряване на видимостта на едно уеб приложение в резултатите на търсачки като Google;

**ASO –** Процесът на подобряване на видимостта на едно мобилно приложение в резултатите на търсачките в магазините за мобилни приложения като Apple App Store и Google Play Store;

**Hashing** – Еднопосочна, математическа функция, която служи за преобразуване на входна последователност в изходен низ с фиксирана дължина;

**Id** –Идентификационен номер на запис в базата данни;

# Увод

„Дневник за автомобила“ е платформа, проектирана и разработена съобразно налагащите се тенденции и технологии за разработката на мобилни приложения. Мотивът на автора за разработването на това приложение идва от това, че голям набор от хора, притежаващи някакъв вид автомобил, допускат грешката да забравят до кога им важат сроковете за дадени задължителни разходи по нея(данък моторно превозно средство, гражданска отговорност, годишен технически преглед, автокаско). Идеята на това приложение е да има опция за въвеждане на тези задължителни разходи и при наближаване на крайната датата, седем дни преди това, потребителят да бъде известяван. Приложението ще има лесен за използване интерактивен интерфейс с минимални отделни полета и страници, за да бъде възможно най-удобен и бърз за работа за потребителя. Някои от основните характеристики на описаното приложение са:

* наличие на интерактивен и интуитивен интерфейс;
* дизайн, съобразен със съвременните изисквания;
* бързина при работа със приложението;
* гъвкаво с възможност за разрастване;
* функционалност, която позволява редакция, добавяне, изтриване на дадени полетата.

“InfoCar” е приложение, което ще дава възможност на потребителя да следи в града, в който се намира, къде има бензиностанции и какви са цените на горивата в при тях. Ще има страница, в която ще може да намери къде са автосалоните в града, за закупуване на нов автомобил или за завеждане на автомобила си за ремонт в дадените оторизирани сервизи. Същото така потребителите ще имат възможност за въвеждане на разходите си по автомобила – разходи за гориво, ремонти, данъци и т.н. За реализацията на проекта е използван програмният език – Kotlin и за база данни е използвано Firebase Realtime Database, която работи на основа NoSQL.

Мобилното приложение е достъпно за мобилни телефони с операционна система Андроид в Play Store - // да сложа линк, ако/като го кача в play store-a.

# Цели на дипломната работа

Основната цел на потребителите е да не се притеснят за това дали не изпускат някое важно заплащане по автомобила си и ще имат възможност за следене на разходите по автомобила си. Това доведе до идеята за разработване на дипломна работа, която е мобилно приложение, „Дневник за твоята кола“, което ще даде възможност за дефинирането на следните задачи за изпълнение:

1. Избор на подходящи технологии и средства за разработка, които да осигурят лесна, ефективна и интуитивна работа със системата;

2. Разработване на собствена визия на приложението;

3. Изграждане на адаптивен и улеснен за потребителя дизайн на приложението, което да е съвместимо с различните на големина видове дисплей;

4. Осигуряване на възможност за „разрастване“ на приложението от към функционалност;

5. Осигуряване на бързина при работа в приложението.

# Структура на дипломната работа

Дипломната работа се състои от увод, 5 глави, заключение и използвана литература.

* **Увод** – представя въведение в тематиката, целите и предназначението на дипломната работа;
* **Първа глава –** В „**Съществуващи приложения**“ се описва за вече създадените и подобни приложения, които потребителите вече използват. Ще се коментират предимствата и недостатъците спрямо дипломната работа;
* **Втора глава –** „**Използвани технологии и средства**“ разглежда използваните технологии и средства нужни за реализацията на дипломната работа:

Kotlin, Firebase, Firebase Database, Illustrator.

-----------------------------------------------------------------------------------------------

* **Трета глава** –В **описание на системата** се дефинират изискванията към платформата, основните модули и тяхната функционалност. Всичко това е съпроводено с детайлен снимков материал.
* **Четвърта глава** – **Ръководство за програмиста** представя платформата като я описва детайлно чрез пълна диаграма на базата данни.
* **Пета глава** – **„Анализ, SEO, домейн име, хостинг, тестване, критични точки и усъвършенстване“** дава подробна информация за това как ще бъде изпълнен всеки един от изброените в заглавието процеси.
* **Заключение** – Предоставя обобщена информация за цялото уеб приложение и посочва перспективите за бъдещото развитие на проекта.
* **Използвана литература** – Упоменават се източниците, цитирани в дипломната работа и тези, използвани за разработката на уеб платформата.

# Съществуващи приложения

Приложенията, които ще бъдат разгледани в тази глава, биват CRUD системи, които позволяват управление и работа със съдържание. Четирите основни функционалности, които CRUD системата позволява са:

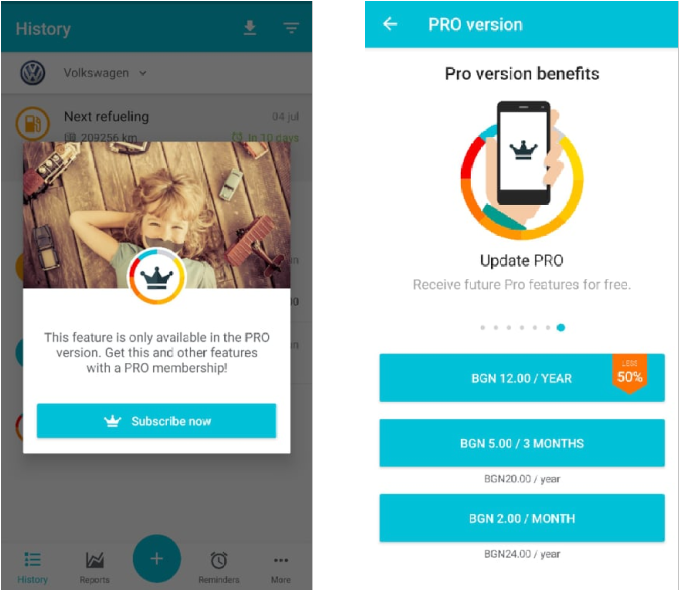
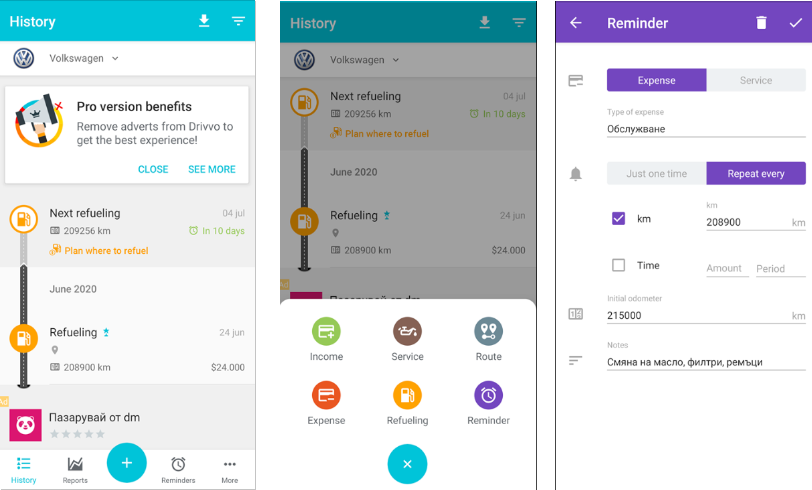
* Създаване (Create);
* Четене (Read);
* Актуализиране (Update);
* Изтриване (Delete).

Авторът е избрал тези мобилни приложения поради факта, че са използвани от най-много потребители и имат най-висок рейтинг. Ще бъдат разгледани възможно най-подробно тези приложения и ще се обсъдят предимствата и недостатъците им от гледна точка на потребител.

## Drivvo

## „Drivvo“ е едно от най-сваляните приложения в Google Play Store (Android) и Apple App Store (IOS). Приложението е създадено Юни месец, 2014 година в Бразилия. „Drivvo“ има над 1.5 милиона тегления в онлайн магазините за приложения, които идват в два вида в зависимост от операционната система, с която ви разполага телефона. В Google Play Store, приложението е оценено с 4.6 рейтинг от максимални 5.

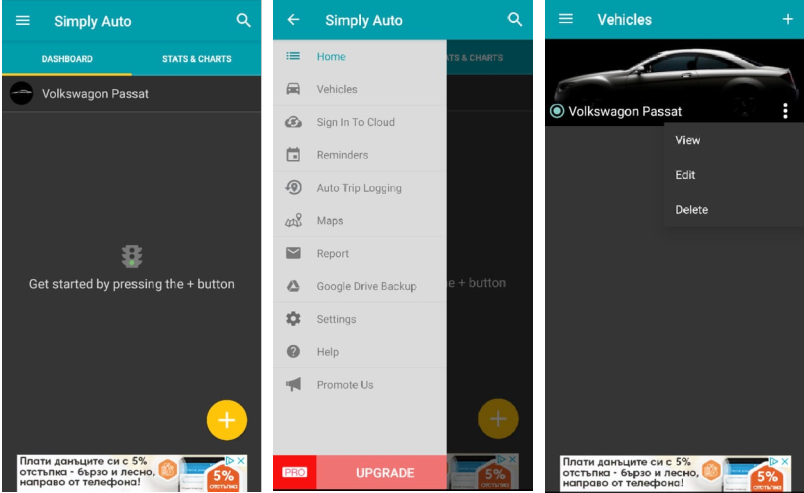
„Drivvo“ разполага с много опции, които са удобни и улеснени за потребителя. Минусите на приложението не са много и са повече от страна на потребител отколкото на разработчик. Има реклами, които можеш да махнеш, но трудно, и понякога стоят закачени на няколко прозореца. Един от начините да ги премахнеш е да станеш PRO потребител. От там идва и друг дискомфорт. Когато натиснеш за опция да станеш PRO потребител ти изскача един pop-up, който ти не можеш да махнеш освен, ако не натиснеш „Subscribe now“, което те праща на екран за плащане и след това да дадеш бутон за назад. Незначително, но все пак неудобно. Макар и опция „PRO“ си има и своите плюсове, те са по-скоро за хора притежаващи служебни автомобили отколкото лични.

Фигура 1. **Drivvo**

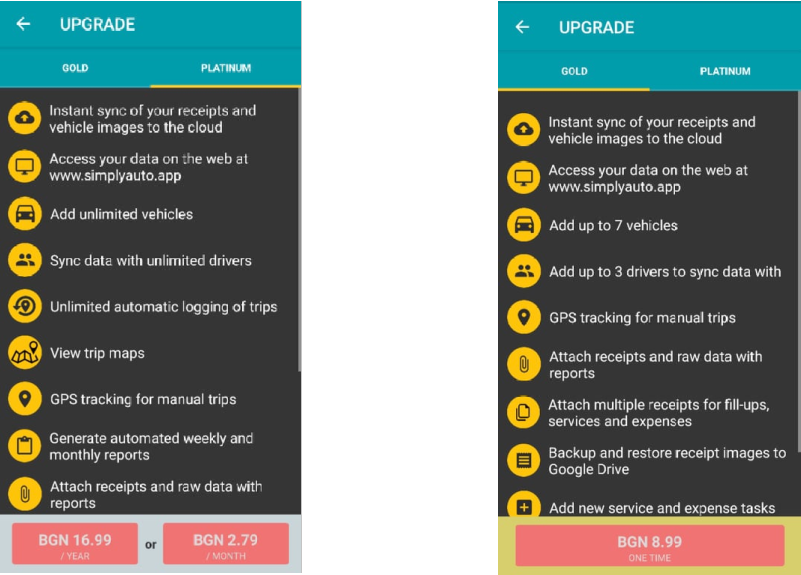
**Фигура 2. Drivvo**

## Simply Auto

„Simply Auto“ е също едно от най-сваляните приложения в Google Play Store (Android) и Apple App Store (IOS), над 500 хиляди сваляния, но не е толкова популярно колко другите две, които са изброени тук. Рейтинга на приложението за двете операционни системи е 4.5 от максимални 5. Приложението е създадено от Mrigaen Kapadia, през 2014 година.

Фигура 3. Simpy Auto

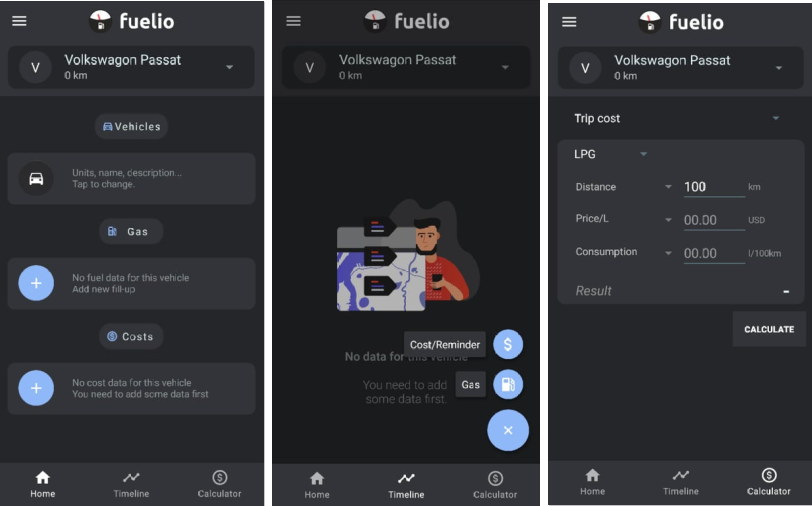
За разлика от „Drivvo“, „Simply Auto“ няма толкова много опции, които биха били полезни за една част от потребителите ѝ. Приложението има много ненужни допълнителни стъпки, които е необходимо да направиш, за да стигнеш до едно определено място. Тези излишни стъпки карат потребителя да отделя и губи повече време в самото приложение. „Simply Auto“, както и всяко друго приложение от този тип, трябва да осигурява бърз, лесен и с минимални стъпки достъп до дадени екрани. Това не се наблюдава тук. Както и други приложения, това също има заделено място за рекламни цели в него, които могат да бъдат премахнати само и единствено чрез закупуване на, така наречените, „PRO“ версии, които идват в два вида – Gold, Platinum. Като се има предвид, разбира се, че „Platinum“ версията ти дава повече и неограничени възможности в някои опции. Версия „Gold“ има по-малко опции и някои от тях са с ограничение. „PRO“ версиите са повече за хора, които притежават повече от един автомобил и използват през повечето време автомобила си.

**Фигура 4. Simpy Auto**

## Fuelio

„Fuelio“ е приложение, което върви наравно с „Drivvo“. Има над 1.5 милиона сваляния от Google Play Store (Android) и Apple App Store (IOS). Има рейтинг от 4.6 от максимални 5. „Fuelio“ е създадено през 2011 година, а от 2015 година насам, приложението е в партньорство със Sygic. Това е компания, която е фокусирана върху автомобилни приложения, навигации и приложения свързани с пътувания.

„Fuelio“ e напълно безплатно приложение. Няма добавени „PRO“ версии, няма заплащания в самото приложения. Лесен е за употреба и има бърз достъп с минимални стъпки до опциите, които искаш да погледнеш.

Фигура 5. Fuelio

## Сравнителен анализ на системите

Разработеното мобилно приложение притежава голям брой сходства, но и различия, в сравнение с някои от по-горе споменатите известни приложения.

Основните прилика, в сравнение с „**InfoCar**“ са:

* Напълно безплатно мобилно приложението;
* Възможност за създаване на един или повече автомобили;
* Възможност за следене на годишен финансов разход;
* Възможност за следене на разход на гориво.

Основните различия, в сравнение с „**InfoCar**“ са:

* Локализация на оторизирани сервизи в града, в който се намира потребителя;

Основни сходства, в сравнение с „**Drivvo**“ са:

* Възможност за следене на разходите по автомобила;
* Възможност за налагане на известия, ако принадлежат ремонти, обслужвания и т.н.;
* Възможност за въвеждане на един автомобил.

Основните различия, в сравнение „**Drivvo**“ са:

* **Drivvo** позволява на потребителя да въведе повече от един автомобил, ако е заплатена „PRO“ версията. **InfoCar** е напълно безплатно и можеш да въвеждаш повече от един автомобил без заплащане;
* **Drivvo** има опция за платено приложение с повече опции. **InfoCar** е напълно безплатно
* **Drivvo** ти дава възможност за изтегляне на pdf файл пълен с разходите по автомобила. **InfoCar** няма тази опция.

Основни сходства, в сравнение с „**Simply Auto**“ са:

* Възможност за следене на разходите по автомобила;
* Възможност за налагане на известия;
* Възможност за въвеждане на повече от един автомобил.

Основните различия, в сравнение „**Simply Auto**“ са:

* **Simply Auto** дава възможност за следене на пътуванията, които потребителят е правил. **InfoCar** я няма тази възможност
* За разлика от **InfoCar, Simply Auto** има възможност за синхронизиране на данните от мобилното приложение заедно с уеб апликацията им;
* За разлика от **Simply Auto, InfoCar** е напълно безплатно приложение.

Основни сходства, в сравнение с „**Fuelio**“ са:

* Възможност за налагане на известия;
* Възможност за въвеждане на неограничен брой автомобили;
* Напълно безплатно приложение;
* Възможност за въвеждане на разходите по автомобила си.

Основните различия, в сравнение „**Fuelio**“ са:

* За разлика от **InfoCar, Fuelio** има опция за бекъп като използва Google Drive или DropBox;
* **Fuelio** дава възможност за категоризиране на разходите па групи, които потребителят е изхарчил;
* **Fuelio** дава възможност за следене на разхода на гориво чрез диаграма.

# Технологии и средства за разработка

В тази глава ще се разгледат всички технологии и средства, използвани за разработката на мобилното приложение. За създаването на едно напълно функциониращо, съвременно и интерактивно мобилно приложение е необходимо един набор от различни технологии – програмен език, база данни, програма за създаване на икони и дизайн.

За разработката на мобилното приложение „**InfoCar**“, авторът е използвал програмният език „Kotlin“. За база данни е използвано „Firebase“ и „Firebase database“. За създаване на иконите – „Adobe Photoshop“.

## Android Studio

Android Studio е официалната интегрирана среда (integrated development environment (IDE)) за разработка на операционната система Android на Google, изградена върху софтуера на IntelliJ IDEA на JetBrains и разработена специално за разработка на Android приложения.

Android Studio е пригодено за работа върху Windows, macOS, Linux операционни системи. Използва се като заместител на Eclipse Android Development Tools (E-ADT) като главна интегрирана среда за разработка на Android апликации.

Интегрираната среда, Android Studio, бе обявена на 16 май, 2013 година на конференцията на Google I/O. Първата нестабилна версия на е била 0.1 пусната през май на 2013година. След това се пуска бета версията – 0.8 през юни, 2014 година. Първата стабилна версия е пусната през декември 2014 година като започва от версия 1.0.

Android Studio е създаден от езиците Java, C++ и Kotlin. Последната стабилна версия досега, която е излезнала на пазара е 4.0.1, на 14 юли, 2020 година. Програмата е създадена от разработчиците на Google и JetBrains.

## Kotlin

Kotlin е крос-платформа, статично типизиран език за програмиране с общо предназначение с изводи за тип. Kotlin е проектиран да работи съвместно с Java, а JVM версията на стандартната му библиотека зависи от библиотеката на класове Java, но извода на типа позволява синтаксиса му да бъде по-кратък.

На 7 май, 2019 година, Kotlin заменя Java като предпочитан език от Google за разработване на мобилни приложения.

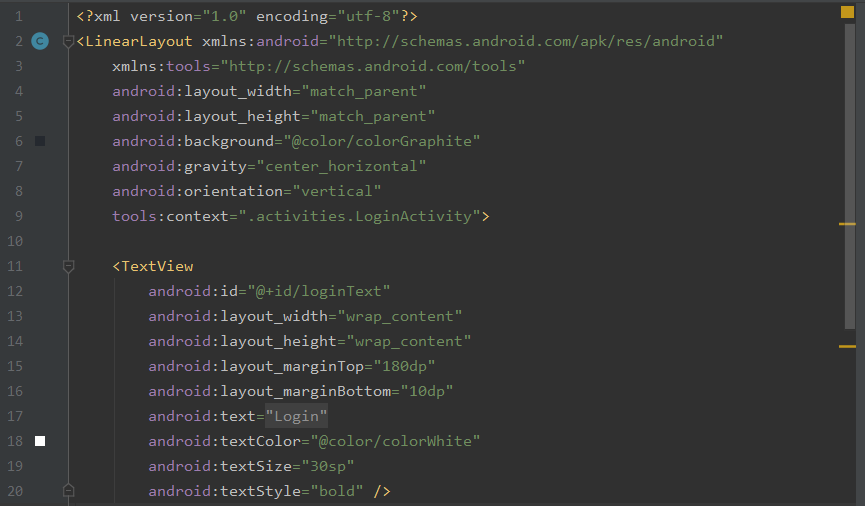
JetBrains работи върху за създаването на нов език за програмиране, с главен програмен дизайнер, Андрей Бреслав, който да е съвместим с Java Virtual Machine (JVM). През юли, 2011 година, след дълги години работа, JetBrains пускат езика **Kotlin**. Една от главните цели на Kotlin е да може да се компилира толкова бързо колкото Java. Езикът е кръстен на острова Kotlin, който се намира близо до Санкт Петербург, Русия.

Първата стабилна версия на Kotlin e v1.0, която е изкарана на 15 Февруари, 2016 година. А през 2017 година, Google обяви първокласна поддръжка за Kotlin на Android операционна система. Последната версия до днешна дата е v1.4.

## XML

XML(Extensible Markup Language) е език, който дефинира набор от правила за кодиране на документи във формат, който може да се чете както от човек, така и от компютърната машина. Създаден е през 1996 година, като се пуска на пазара на 10 февруари 1998 година. Последната версия е 1.1, пусната на 29 септември 2006 година.

XML е създаден и разработван от Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, Francois Yergeau, John Cowan.



## Firebase, Firebase Database

Firebase е платформа за разработка на мобилни и уеб приложения. Firebase е създаден през септември, 2011 година в Сан Франциско, Калифорния, САЩ от двамата ѝ основателя – Джеймс Тамплин и Андрю Ли, а през 2014 година е придобита от Google.

За изработването на мобилното приложение съм използвал функциите Firebase Authentication и Firebase Realtime Database.

**Firebase Authentication** се използва за вписване в дадено приложение, което изисква login метод. То позволява вписването да се извърши чрез един или повече методи като например – чрез email и парола, телефонен номер, чрез други социални мрежи, и т.н.

**Firebase Realtime Database** се използва за складиране и синхронизиране на информацията като самата база данни е на принципа на NoSQL база данни.

## Adobe Illustrator

Adobe Illustrator е редактор на векторна графика разработен от Adobe Inc. Първоначално Illustrator е разработен с идеята да функционира на Apple Macintosh, както и прави през първите си версии – 1.0, 1.1, 88. От версия 2.0 Adobe Illustrator има възможност за работа и на Windows. Adobe Illustrator е пуснат на пазара през януари 1987г. Програмата е разработена на езика за програмиране – C++. Последната издадена версия на Adobe Illustrator е на 16 юни 2020г. Версия CC 24.2.

# Описание на системата

Преди започването на разработката на мобилното приложение, трябва да се формулират основните изисквания, към мобилното приложение.

## Изисквания

* Мобилното приложение трябва да бъде лесно, ефикасно и интуитивно за ползване.
* Мобилното приложение трябва да има модерен и атрактивен дизайн.
* Мобилното приложение трябва да има адаптивен интерфейс.
* Мобилното приложение трябва да е гъвкаво, с възможност за разширение.
* Мобилното приложение трябва да работи бързо.
* Мобилното приложение трябва да има добре изградена база данни

### Ефикасност и интуитивна използваемост

Мобилното приложение трябва да е лесно, ефикасно и интуитивно за работа. Това се постига като следваме набор от прости правила при създаване на дизайна, като например разположение и достъпност на елементите. Мобилното приложение не трябва да изглежда объркващо и потребителите трябва да могат да достигнат до крайната цел без затруднения.

### Модерен и атрактивен дизайн

Мобилното приложение трябва да има модерен и атрактивен дизайн. Дизайнът трябва да е изчистен и не трябва да е претрупан с елементи, които биха объркали или подвели потребителя.

### Адаптивност

Мобилното приложение трябва да има адаптивен дизайн, което позволява на оформлението на различните страници да се адаптира към устройства с различна разделителна способност.

### Бързина

Мобилното приложение трябва да притежава бързодействие. **Тука трябва да разгледам да видя кво ми трябва.**

### Гъвкавост и възможност за разширение

Мобилното приложение трябва да може да се усъвършенства в бъдеше. **Това е постигнато с помощта на използвания MVC архитектурен модел. Чрез него, функционалността бива разделена на модули, а бизнес логиката на слоеве. Това помага на безпроблемното разширяване на платформата.**

### Оптимизирана база данни

Базата данни трябва да е добре структурирана и правилно индексирана. Една добре структурирана база данни:

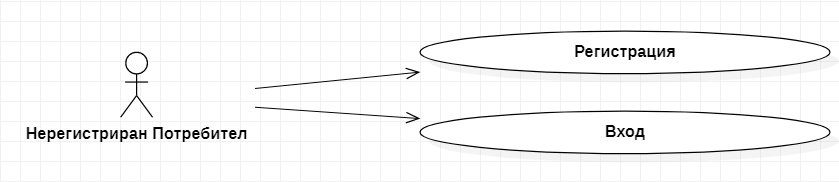
* Спестява дисково пространство, като елиминира излишните данни;
* Поддържа точността и целостта на данните;
* Позволява по-бързо извличане на данни.

## Същност на приложението

В тази част ще бъде разгледана всяка една основна част от уеб приложението и ще бъдат описани основните му функционалности. Всичко това е съпроводено с детайлен снимков материал.

### Системна среда

В мобилното приложение съществуват две роли, но има възможност за добавяне в бъдеще на още поне една. Всяка от тях има достъп до различни функционалности на мобилното приложение, например единствено регистриран потребител би могъл да използва всичките функции на мобилното приложение. Тези роли биват: нерегистриран потребител, регистриран потребител.

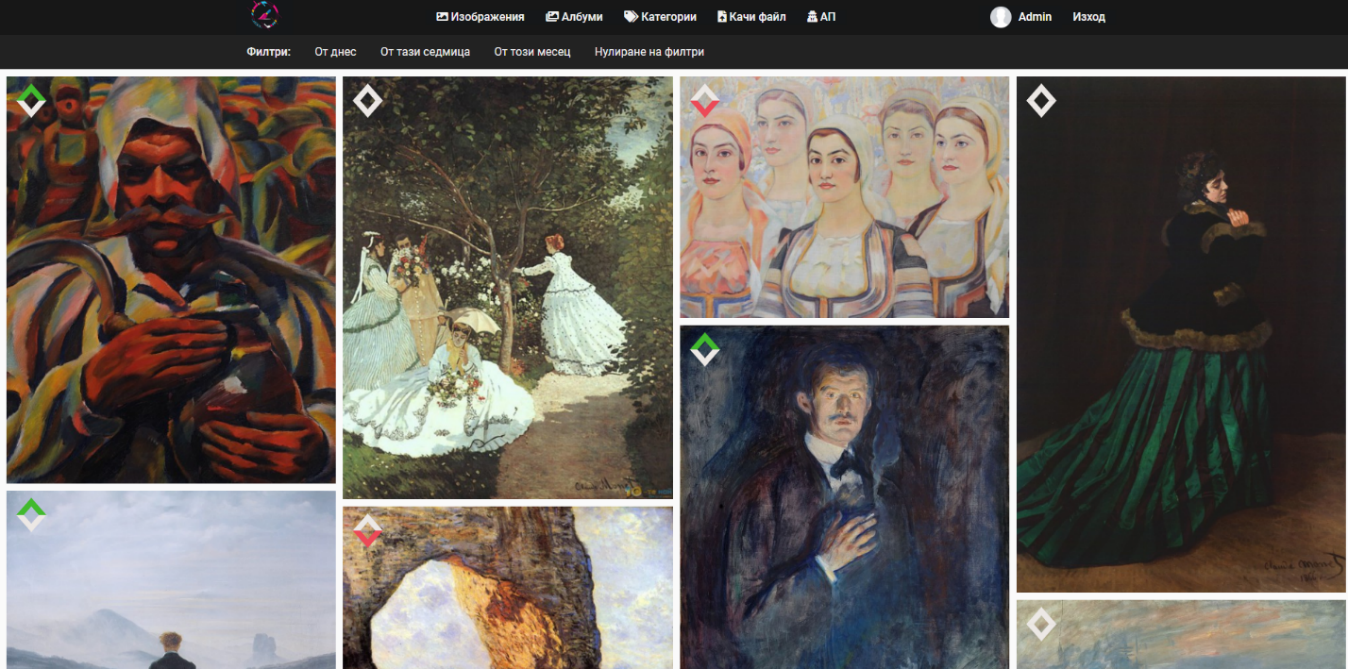
Фигура 6. Use case за нерегистриран потребител

Фигура 7. Use case за регистриран потребител

### Начална страница

Първата страница, на която попада потребителя при зареждане на приложението е началната страница. На нея са разположени всички бутони, които навигират потребителя към нужните страници и функции на приложението.

На тази страница потребителите имат възможността да отиват в различните страници – „моят профил“, „моите разходи“, „настройки“, „сервизи и магазини“, „закони“, „бензиностанции наоколо“.

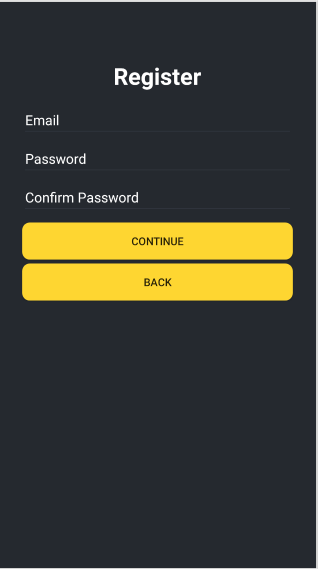


Фигура 8. Начална страница за регистриран потребител

### Страница за регистрация

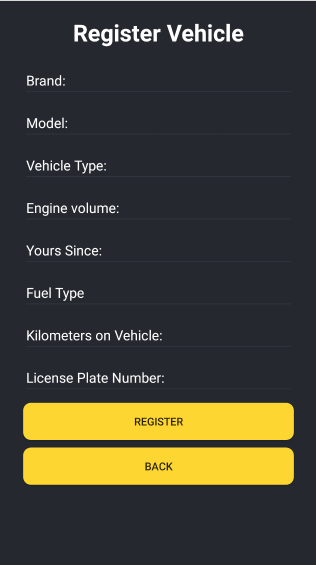
Страницата за регистрация служи за добавяне на нови потребители. При регистрация потребителят трябва задължително да въведе данните в двете страници. На тази страница са визуализирани три полета за въвеждане на информация:

* Парола;
* Имейл;
* Потвърждение на имейл.

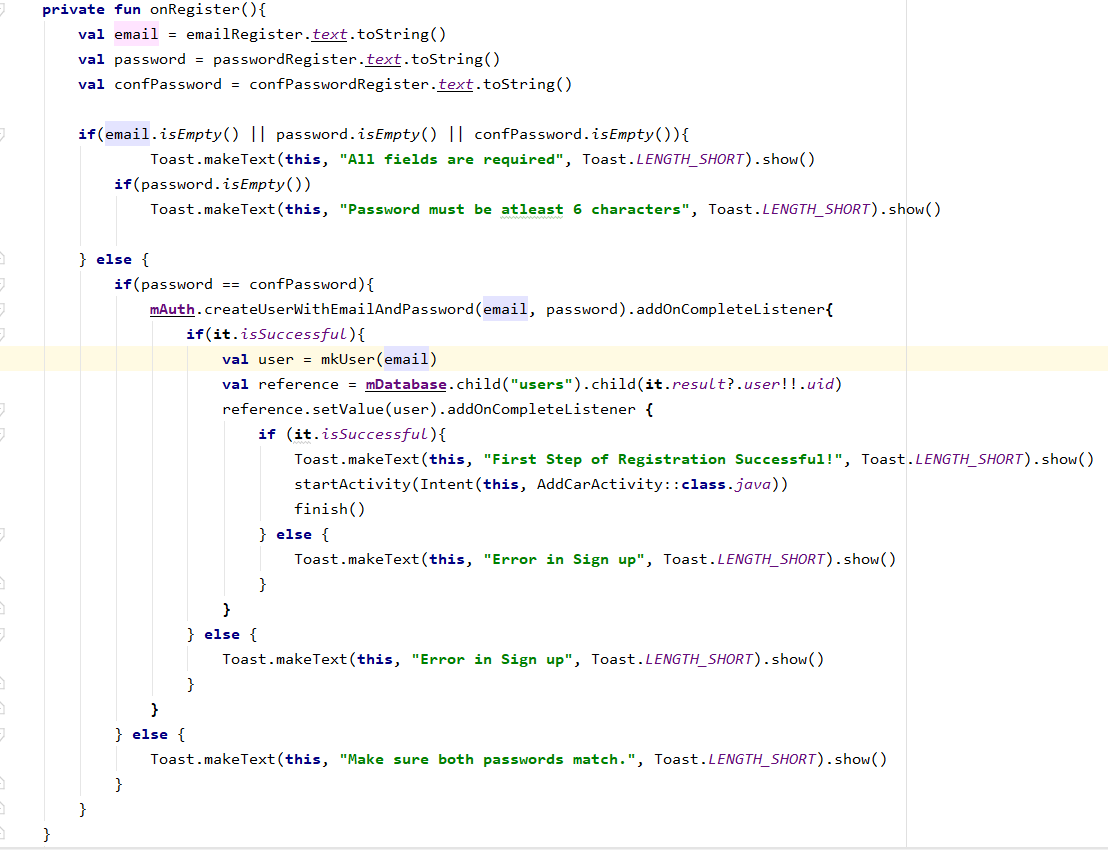
Фигура 9. Страница за регистрация

След попълване на първата страница от регистрацията, за да може да се продължи напред към втора стъпка, приложението минава през валидация, за да се види дали имейлът, с който потребителят ще се регистрира е валиден и дали паролата отговаря на нужните изисквания. След успешно премината валидация, приложението препраща бъдещия потребител към втора стъпка от регистрацията. На тази страница потребителят трябва задължително да въведе данните за автомобила си. На страницата са визуализирани осем полета за попълване:

* Марка на автомобила;
* Модел на автомобила;
* Вид на автомобила;
* Обем на двигателя / конски сили;
* Дата на регистрация;
* Тип гориво;
* Изминати километри на автомобила;
* Регистрационен номер на автомобила.

**Фигура 10. Страница за регистрация - втора стъпка**

Преди информацията да бъде запазена в базата данни, тя минава през валидация на сървърната част. Валидацията изисква всички полета да са попълнени според изискванията, които са ѝ предоставени. Ако някое от тези изисквания не е покрито, сървърът ще върне съобщение за грешка и информация за нея. Накрая потребителя се запаметява в таблиците Users и CarInfo на базата данни.

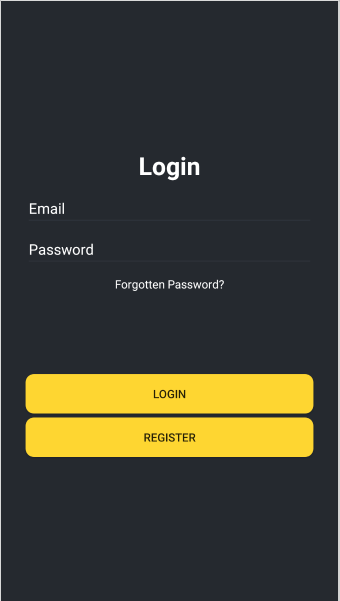
Фигура 11. Валидация на първата страница от регистрацията

**Фигура 12. Функция за регистрация втора стъпка**

### Страница за вход

Тя служи за вход на вече регистрирали се потребители. На тази страница има 2 полета за въвеждане на информация:

* Имейл;
* Парола;

Фигура 13. Страница за вход

След като информацията бъде въведена, тя минава през валидация на сървърната част. Валидацията изисква и двете полета да са попълнени. След това функцията проверява дали съществува запис с този имейл и парола в базата данни и в зависимост от това връща съобщение за грешка, или позволява на потребителя да достъпи своя профил.

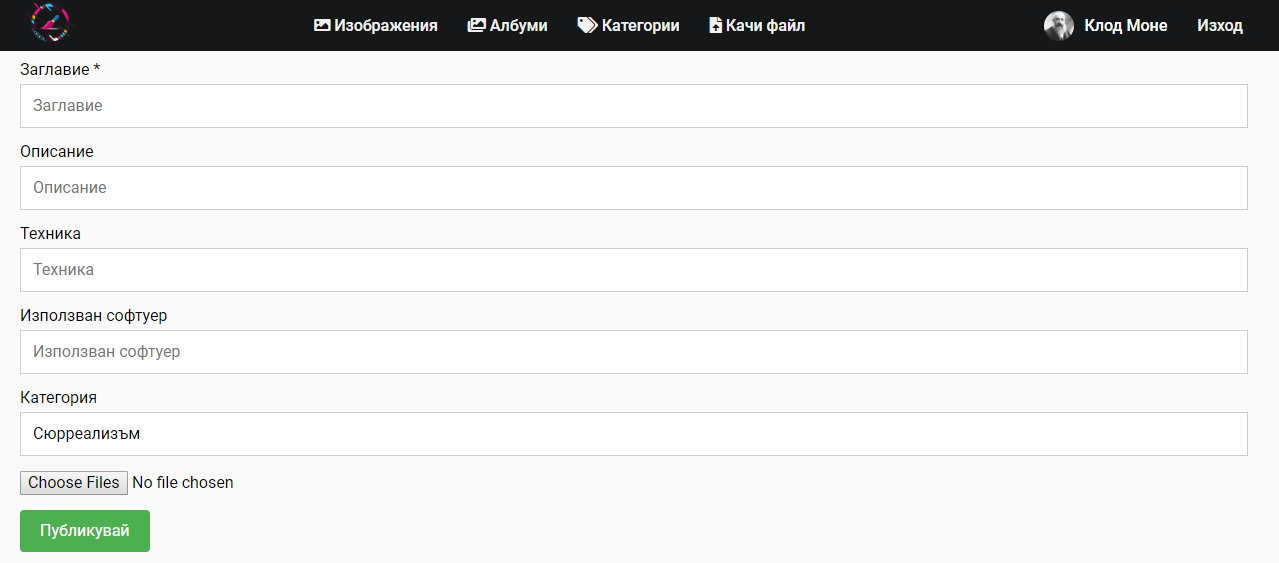


Фигура 14. Функция за вход

* + 1. Страница за добавяне на нов автомобил

Тази страница служи за добавяне на нов автомобил като вече има поне един съществуващ в акаунта на потребителя. На тази страница има десет задължителни полета за попълване:

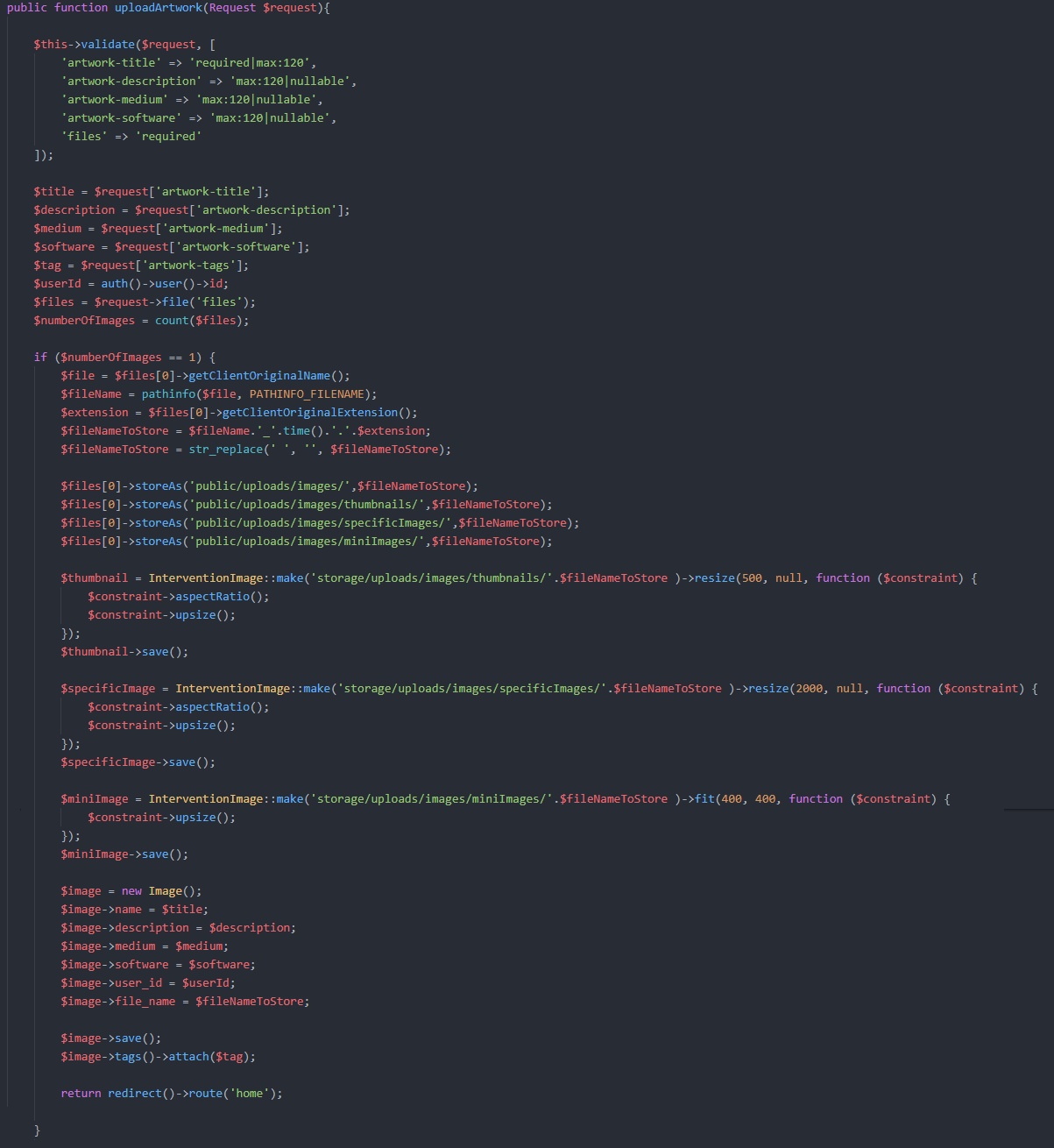
* Марка на автомобила
* Модел на автомобила
* Тип автомобил
* Обем на двигателя на автомобила
* Година на купуване
* Вид на горивото
* Километри на автомобила
* Регистрационен номер на автомобила
* Дата на купуване на гражданска отговорност
* Дата на успешно преминат годишен технически преглед



Фигура 15. Страница за качване на файлове

Преди информацията да бъде обработена и запазена в базата данни, тя минава през валидация на сървърната част. Валидацията изисква всички полета да са попълнени с нужните условия. Ако някое от тези изисквания не е покрито, сървърът ще върне съобщение за грешка и информация за нейния тип.

След успешно премината валидация цялата информация се запазва в базата данни и се визуализира в профила на потребителя. До самата визуализация потребителят ще може да изтрива или коригира нужната информация, ако след време се промени някоя от данните или вече този автомобил не е наличен при него.

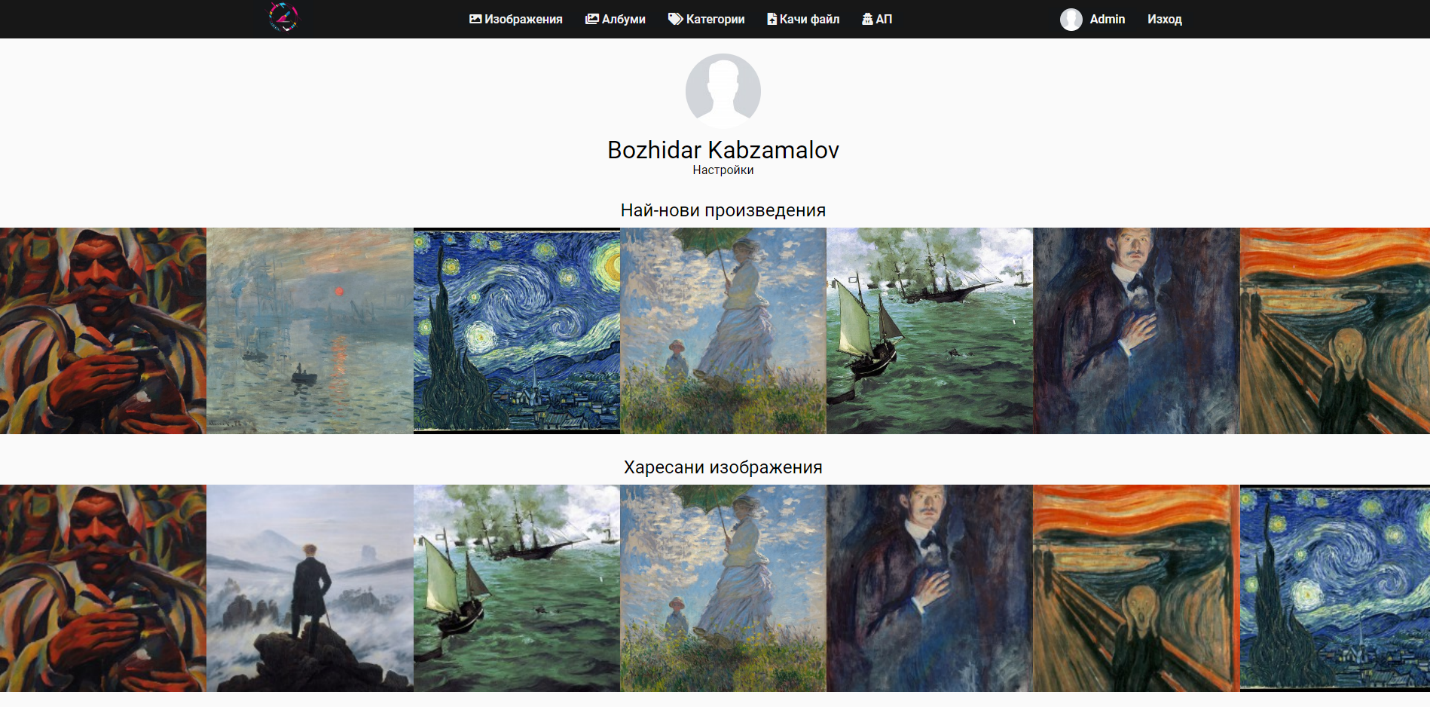


Фигура 16. Функция за качване на единичен файл

### Потребителска страница

На потребителската страница може да се види основната информация за автомобила/автомобилите на потребителя.

Визуализиран е бутон „Настройки“, който седи в менюто, което е позиционирано най-долу на приложението. Този бутон води към страницата за настройки на профила, от която потребителят може да прави промени в профила си.



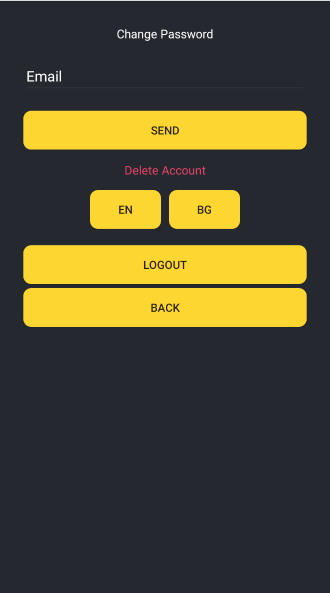
Фигура 17. Потребителска страница

### Страница за настройки на профила

Страницата за настройки на профила позволява на потребителите да правят промени по своите секции.

На тази страница са показани три полета за промяна на профила на потребителя

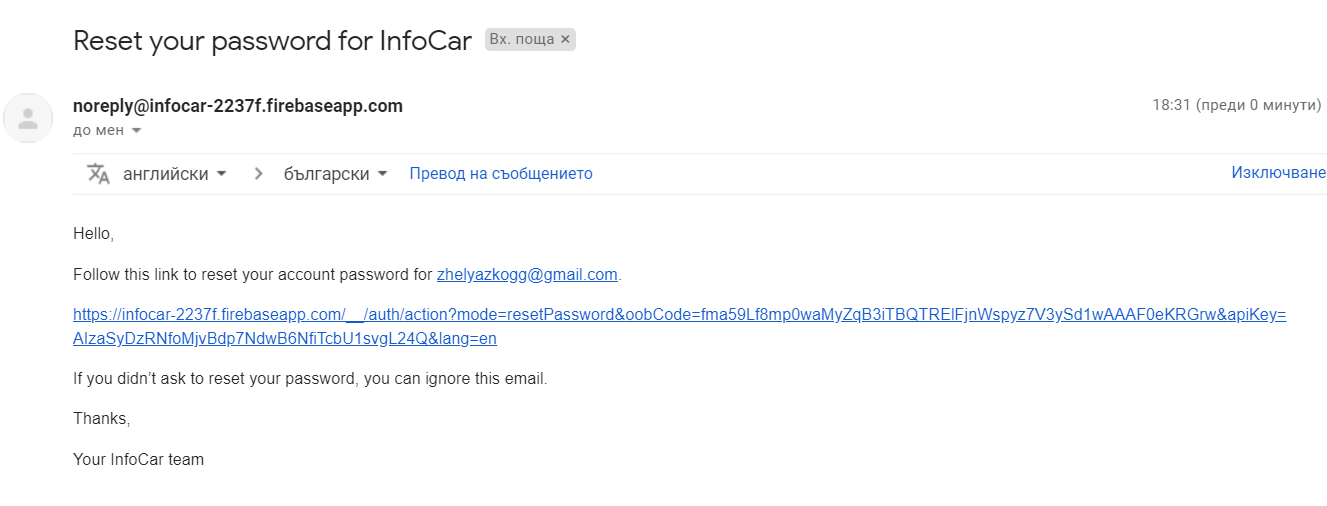
* Промяна на паролата;
* Изтриване на акаунт;
* Смяна на езика от български на английски и обратно;
* Изход от приложението;
* Бутон за връщане към начална страница.



Фигура 18. Страница за настройки на профила

Потребителят ще може да смени езика на профила си от български на английски и обратно в зависимост от неговото желание, може да променя паролата си, може и да изтрие акаунта си завинаги. За промяна на паролата, на потребителя ще му се покаже едно поле, в което трябва да въведе имейла, с който е регистриран профила му, за да му бъде изпратено съобщение с линк към страница, в която да смени паролата си с нова по негово желание.

След успешно въведена нова парола, сървърът извлича информацията от заявката и актуализира данните на потребителя.

Фигура 19. Имейл за подновяване на стара парола.

Фигура 20. Функция за актуализация на профила

В тази глава бяха представени всички основни секции от мобилното приложение. Описана е каква функционалност предоставят, как да бъдат използвани и е обяснен използвания код, реализиращ логиката на системата. Всичко това бе съпроводено с детайлен снимков материал. В следващата глава ще разгледаме архитектурата на системата и базата данни, необходими за бъдещо развитие на системата.

# Ръководство за програмиста

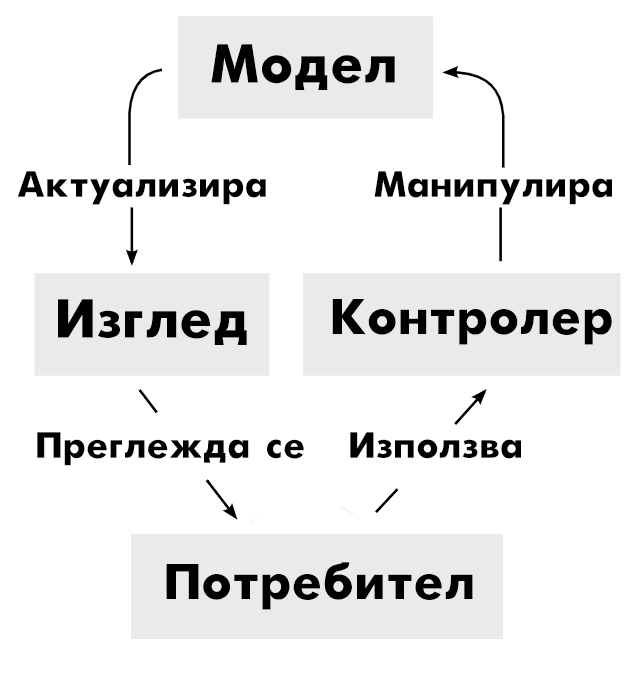
В тази глава е разгледана по-отблизо архитектурата на системата и на базата данни.

Използваната база данни е на Firebase - Realtime Database, която се води NoSQL, а за удостоверяване на потребителите е използвано Firebase Authentication. В нея информацията се съхранява в таблици като всеки един потребител има уникално Id, което се възпроизвежда автоматично при регистрация.

## Архитектура на системата

Model-View-Controller (MVC) е архитектурен модел използван за разработка на потребителския интерфейси, чрез който приложението бива разделено на три взаимосвързани части.

Това позволява обособяването на бизнес логиката от графичния интерфейс и данните в приложението. По този начин се осигурява повторно използване на код и ефективното бъдещо развитие на системата. Тази архитектура е добър избор при проектиране и разработка на уеб, мобилни, десктоп и други приложения.



Фигура 21. Взаимодействията в MVC модела

Целите на MVC модела е да позволи на програмистите, дизайнерите на базата данни и дизайнерите на потребителския интерфейс да работят паралелно по отделните модули без да си пречат един на друг. Например, програмистите могат да проектират структурата на данните и начинът, по който потребителят взаимодейства с тях, без да се налага потребителският интерфейс да е напълно завършен.

Втората цел на MVC модела е да позволи създаването на компоненти, които да са независими един от друг. Това позволява повторна употреба на всеки един компонент и в други приложения. При това по един лесен и бърз начин.

Част от предимствата на MVC са:

* По-бърз процес на развитие;
* Възможността за паралелна работа по различните модули;
* Възможността един модел да има повече от един изгледа;
* Поддръжка на асинхронна техника което позволява разработката на бързи приложения.

Част от недостатъците на MVC са:

* Повишена сложност и абстракция;
* Необходими са познания за множество технологии;
* Нужда от множество програмисти;
* Недостатъчна ефективност на достъпа до данни в изгледите.

Трите взаимосвързани части на MVC са:

Model (Модел) – Моделът представлява част от приложението, което реализира домейн логиката, още известна като бизнес логика. Домейн логиката обработва данните, които се предават между базата данни и потребителския интерфейс. Обикновено това са данните от реалния свят, които се моделират и над които се работи: въвеждане, промяна, показване и т.н.. Често моделът съхранява и извлича официална информация в базата данни;

View (Изглед) – Изгледът отговаря за показването на данните от модела към потребителя. Даден модел може да има няколко изгледа, което позволява информацията да бъде представена по различен начин във всеки един изглед.;

Controller (Контролер) – Контролерите обработват постъпващите заявки, въведени от потребителя, изпълняват подходящата логика, манипулират данните, и чак след това ги изпращат към изгледа.

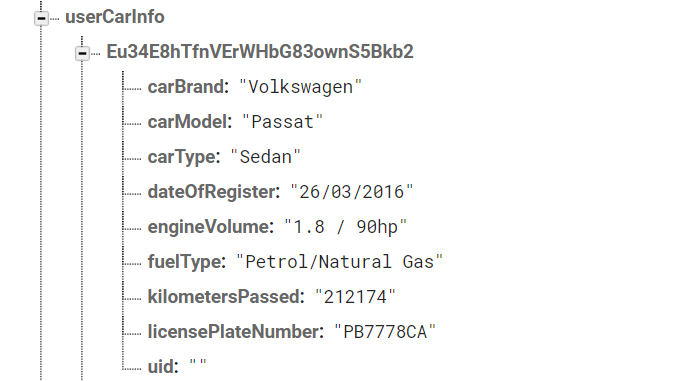
## Архитектура на базата данни

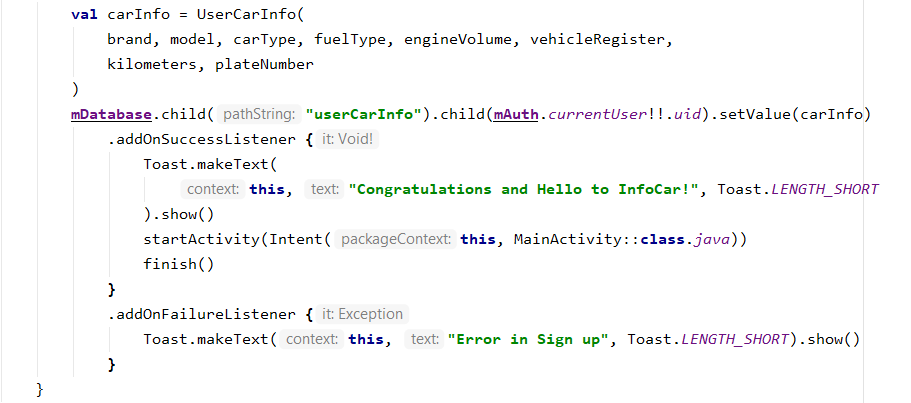
Използваната база данни за това приложение е NoSQL базата данни – Firebase Realtime Database. Приложението използва 4 таблици, за сега, като по-долу ще разгледаме всяка една детайлно.

### UserCarInfo

В таблицата **UserCarInfo** се съхраняват всички данни, които потребителят въвежда при втората стъпка от регистрацията си. Чрез тази таблица данните за автомобила му се възпроизвеждат в профила му. Тази таблица съдържа данни за:

* Автоматично генерирано уникално Id, което съвпада с това на потребителя;
* Марка на автомобила;
* Модел на автомобила ;
* Категория на автомобила;
* Дата на регистрация на автомобила от потребителя;
* Обем на двигателя;
* Вид гориво, с което разполага автомобила;
* Изминати километри на автомобила;
* Регистрационен номер на автомобила.

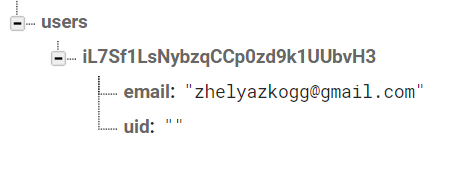
Фигура 22. Таблица UserCarInfo

**Фигура 23. Метод за създаване и записване в таблица UserCarInfo**

### Users

В таблицата **Users** се съхраняват имейлите на регистрираните потребители:

* Автоматично генерирано уникално Id, което съвпада с това на потребителя;
* email на регистрирания потребител.

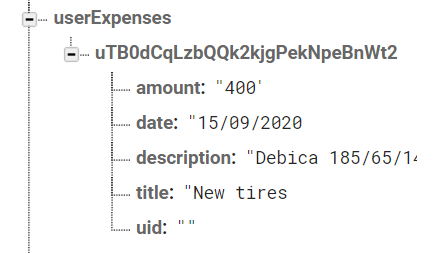
Фигура 24. Таблица users

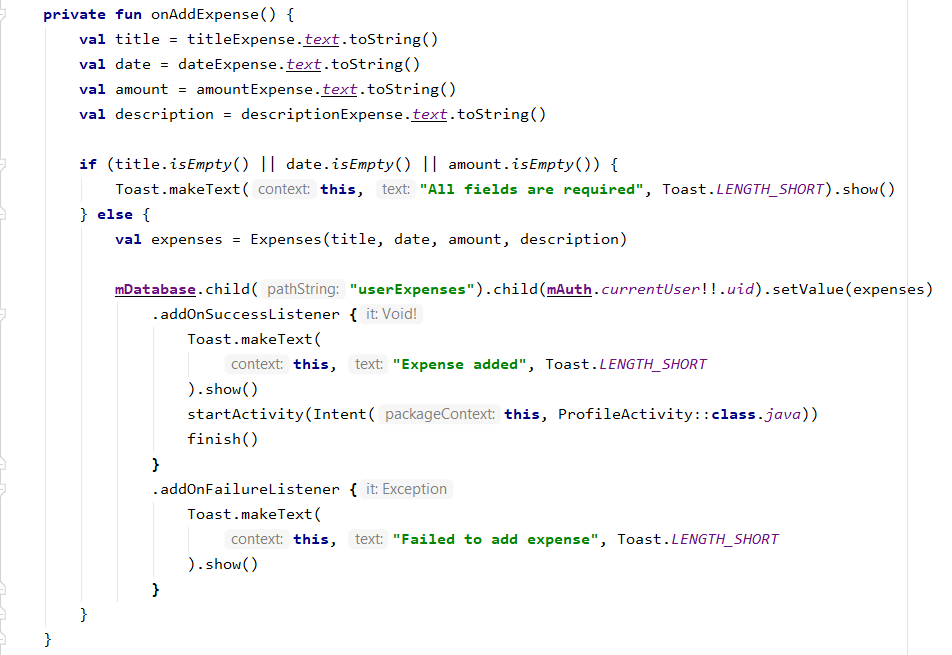
Фигура 25. Метод за създаване и записване в таблица users

### UserExpenses

В таблицата се съхраняват всички разходи, които потребителят е въвел свързани с автомобила си:

* Автоматично генерирано уникално Id, което съвпада с това на потребителя;;
* Заглавие на разхода направен от потребителя;
* Дата на направения разход;
* Сумата изхарчена по разхода;
* Описание на разхода.

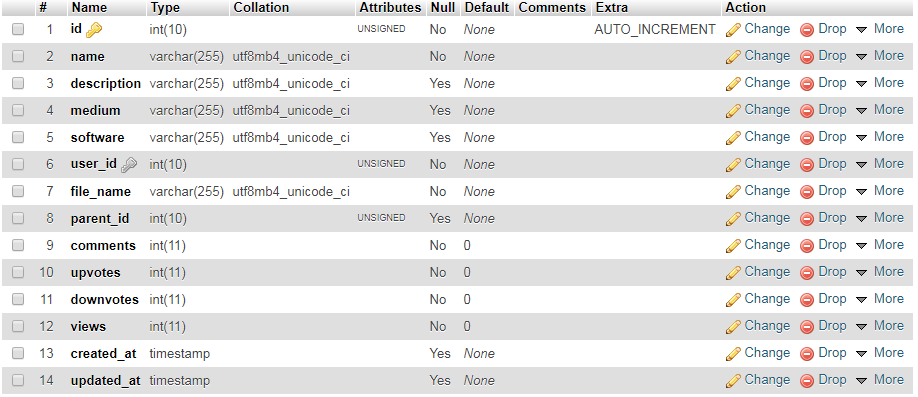
Фигура 26. Таблица Expenses

Фигура 27. Метод за създаване и записване в таблица userExpenses

### Reminders

В таблицата **Reminders** се съхраняват данни за всички известия, които потребителят е въвел и иска да бъде известяван в бъдеще за тях:

* Автоматично генерирано уникално Id, което съвпада с това на потребителя;
* Заглавие на известието;
* Описание на известието;
* Дата на известяване.



Фигура 28. Таблица Reminders

# SEO и ASO, тестване, критични точки и усъвършенстване

## SEO and ASO

Според последни проучвания, 27% от потребителите откриват приложения, докато използват Интернет търсачките през мобилните си устройства. Това прави приблизително една трета от приложенията, които биват открити извън самите магазини за приложения (Google Play Store за Android и Apple App Store за iOS).

SEO (Search Engine Optimization) или Оптимизация за търсещите машини, представлява процесът на подобряване на видимостта на мобилните приложения в резултатите на уеб търсачките. Алгоритъмът, който Google използва, за да предлага и класира приложения в страниците с резултати от уеб търсачката (Search Engine Result Page (SERP)), е различен от този, използван от магазините за приложения, така че класирането на ключови думи за приложението няма да бъде еднакво за уеб търсенето и магазините за мобилни приложения. Най-често срещаният резултат от търсене на мобилно приложение в уеб пространството е чрез единичен фрагмент. Той съдържа име на приложението, броят на дадените оценки заедно със средната оценка и тестов фрагмент, който описва приложението, последвано от бутон за инсталиране. Този фрагмент се нарича още Single Snippet.

Оптимизация на магазините за мобилни приложения (App Store Optimization(ASO)) e процесът за подобряване на видимостта на приложенията в магазините за приложения и увеличаване на процента на преобразуване на приложения. За да се класира на по-високо ниво в резултатите от търсене в магазините за приложения, ASO използва и честотата на посещение на приложението (Click-through rate (CTR)). Това означава, че трябва да убедите потребителите да натиснат върху приложението ви в магазина за приложения след като го намерят. Това може да се постигне чрез оптимизиране на името на приложението, логото на приложението, екранните снимки на приложението и същевременно средната оценка на приложението. След като потребителите вече са посетили страницата на вашето приложение, в магазина за приложения, трябва да ги убедите да си свалят самото приложение или да го закупят в зависимост от приложението, което е разработено. Тази част на ASO е позната като Оптимизация на степен на конверсия (Conversion Rate Optimization (CRO)).

Разликата между ASO и SEO Оптимизацията е, че ASO се използва за оптимизация в магазините за мобилни приложения, а SEO се използва за оптимизация в търсачките на уеб пространството. Другата главна разлика между ASO и SEO са факторите за класиране. SEO факторите за уеб търсачките, като Google например, включва повече от 200 аспекта, докато за ASO е много по-кратко.

ASO и SEO (Search Engine Optimization) имат прилики в техниките за оптимизиране и класиране, така че дадено приложение, ако се класира високо в магазините за приложения, то най-вероятно ще се класира добре и за подобни търсения в мрежата.

Една от важните причини за използване на оптимизация на приложенията в магазините са приложение, е че общият брой на приложенията в Google Play Store (Android) и Apple App Store (iOS) надхвърля 5 милиона. Разработчикът трябва така да оптимизира и рекламира приложението си, че да може то да излиза на по-горни позиции в класацията.

## Критични точки

Една от критичните точки е възможността за хакерска атака върху приложението. Мерките, които са взети биват:

* Password hashing на паролите в базата данни. В случай че базата данни е компрометирана, хакерите биха намерили произволно изглеждащи низове от знаци, в които паролите са криптирани и това позволява за малко по-голяма защита над паролите на потребителите.
* Firebase Authentication е разработена с цел скриване и криптиране на паролите на потребителите.

Друга критична точка е потенциалното претоварване на сървъра и надхвърляне безплатния лимит за памет на базата данни. За да не се случва това трябва съответно да се заплати на Firebase, за да може да се използват по-голяма памет за приложението.

## Усъвършенстване

Няколко от подобренията подготвени за в бъдеще биват:

* Подобрен дизайн на приложението;
* Добавяне на повече от един автомобил за даден потребител;
* Добавяне на функция за изтегляне на всички разходи, които са записани от потребителя, под формата на pdf файл;
* Добавяне на функция за свързване с GPS-a на телефона, за да може потребителят да разглежда за даденият град, в който се намира, къде има бензиностанции, сервизи и автомагазини;
* Добавяне на функция, която сортира цените на бензиностанциите, така че потребителят да може да разбере къде са най-ниските/скъпите цени на горивата в града, в който се намира;
* Добавяне на страница, която автоматично обновява законите за движения по пътищата в дадената държава, в която се намира потребителят;
* Добавяне на функция за смяна на езика на приложението (от английски на български и обратно);
* Разработване на платформата и като изцяло на уеб базирано приложение.

# 

# Авторска справка

В настоящата дипломна работа е представено мобилното приложение „Дневник за автомобила“. Описани са вече съществуващи мобилни приложения с подобна цел, използваните технологии и средства за разработката и дизайна на приложението.

Основните приноси към дипломната работа са:

* Създаване на напълно функциониращо мобилно приложение за следене и записване на важни разходи и напомняния за превозните средства на потребителя.
  1. Проектирането и разработването на базата данни представени в параграф 4.2;
  2. Разработване и описание на пълната функционалност на мобилното приложение в Глава 3;
  3. Проектиране на дизайна на всички уеб страници на приложението чрез употреба на CSS в параграф 3.2;
  4. Имплементацията на защити върху паролите на потребителите от хакерски атаки, описани в параграф 5.2;
  5. ASO оптимизиране на мобилното приложение представено   
     в параграф 5.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Принос** | **Вид принос** | **Цел** | **Задачи** | **Параграф** |
| **1** | **Приложен** | **1** | **2, 6** | **4.2** |
| **2** | **Приложен** | **1** | **1, 2, 5, 6** | **3.2** |
| **3** | **Приложен** | **1** | **3, 4** | **3** |
| **4** | **Приложен** | **1** | **8** | **5.6** |
| **5** | **Приложен** | **1** | **7** | **5.1** |
| **6** | **Приложен** | **1** | **7** | **5.2, 5.6** |

# Заключение

Целта на дипломната работа е постигната и всички поставени пред автора задачи са завършени. Полученият изходен продукт „Дневник за автомобила“ е функциониращо мобилно приложение.

Благодарение на основните използвани технологии Kotlin, XML, Firebase Authentication и Database, платформата притежава бързодействие, лесна е за работа и нейната функционалност безпроблемно може да бъде развита в бъдеще.

Дизайнът на страниците на мобилното приложение взима под внимание това да бъде лесен за използване и навигиране. С времето, дизайнът ще бъде адаптиран все повече към модерните тенденции в мобилното разработване като имплементация на адаптивен дизайн

Авторът предвижда за в бъдеще добавянето на няколко допълнителни модула, с цел разширение на функционалността. Няколко от тези разширения биват - добавяне на повече от един автомобил за даден потребител, добавяне на функция за изтегляне на pdf файл за разходите, които потребителят е записал и добавяне на опция за смяна на език на приложението.

# Използвана литература

1. YouTube: <https://youtu.be/Fs9Yerlclrk>, *последно посетен на 01.09.2020 г.*
2. YouTube: <https://youtu.be/w2j7HxkVdiI>,*последно посетен на 01.09.2020 г.*
3. Udemy: https://www.udemy.com/course/kotlin-programming, *последно посетен на 01.09.2020 г.*
4. TheTool: <https://thetool.io/2017/seo-for-mobile-apps>,*последно посетен на 01.09.2020 г.*
5. App Radar: <https://appradar.com/academy/aso-basics/what-is-app-store-optimization-aso>,*последно посетен на 01.09.2020 г.*
6. Android Developers: <https://developer.android.com/>,*последно посетен на 01.09.2020 г.*
7. Antonio Leiva, „Kotlin for Android Developers: Learn Kotlin the easy way while developing an Android App“, първо издание 2016, ISBN-10: 1530075610
8. David Karlins, „Adobe Illustartor CC For Dummies“, 2020, ISBN-10: 1119641535
9. Pinterest: https://www.pinterest.com/, *последно посетен на 01.09.2020 г.*
10. Fuelio: https://fuel.io/, *последно посетен на 01.09.2020 г.*
11. Drivvo: https://www.drivvo.com/en, *последно посетен на 01.09.2020 г.*
12. Simply Auto: https://simplyauto.app/, *последно посетен на 01.09.2020 г.*
13. Kotlin: https://kotlinlang.org/, *последно посетен на 01.09.2020 г..*
14. XML: https://www.w3schools.com/, *последно посетен на 01.09.2020 г.*
15. Firebase: https://firebase.google.com/, *последно посетен на 01.09.2020 г.*
16. Image Color Picker: <https://imagecolorpicker.com/>, *последно посетен на 01.09.2020 г.*