# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ

## ПРОЕКТ №234

## Driverify

### 1. ИМЕ НА ПРОЕКТА

„Driverify“, категория „Приложни програми“

### 2. АВТОР

Имена: Теодор Валентинов Христов

ЕГН: 0144123027

Адрес: гр. Ловеч, бул. “България” №81, вх. А

Тел.: +359 876 306 495

Имейл: [tedotedo771@gmail.com](mailto:kristianchok@gmail.com)

Училище: Профилирана природо-математическа гимназия – гр. Ловеч

Клас: 12 клас

### 3. РЪКОВОДИТЕЛ

Имена: Надежда Милкова Николова

Тел.: +359 888 196 452

Имейл: [nmnikolova@mail.bg](mailto:nmnikolova@mail.bg)

Длъжност: Учител по информатика в Профилирана природо- математическа гимназия – гр. Ловеч

### 4. РЕЗЮМЕ

#### 4.1. Цел и идея

Годината е 2020. Населението на земята е повече от 7 милиарда и половина. Технологиите се развиват, задоволявайки нуждите на потребителя и улеснявайки живота му. Но прекалените удобства водят до привикване. Статистиката доказва, че голяма част от автомобилните катастрофи са предизвикани от заспали шофьори зад волана – повече от 72 хиляди на година. Но ние хората сме социални същества и е в природата ни да се доверяваме на околните и техните способности, когато те общуват с нас – било то и от изолираното пространство на личния си автомобил.

Мобилното приложение “Driverify” и неговата основна цел – да следи състоянието на шофьора, нивата на агресия у него и начина му на шофиране.

#### 4.2. Основни етапи в реализирането на проекта:

* Проучване на проблема и изготвяне на план за решаването му;
* Проучване на работата на съществуващи системи и някои техники при реализацията им;
* Поясняване на основните и на страничните функционалности на приложението;
* Избор на най – подходящо място за поставяне на смартфона;
* Изучаване на архитектурата на OBD II адаптера;
* Симулиране на данните от компютъра на колата чрез друг смартфон;
* Изследване на комуникационните протоколи в автомобилите;
* Намиране на подходящо тестово превозно средство, върху което да бъдат тествани промените;
* Проучване на възможностите на дадени смартфони;
* Записване на различни видове данни, които да подпомогнат за установяването на състоянието на водача на превозното средство;
* Закупуване на OBD II адаптера;
* Имплементиране на алгоритмите за наблюдение над водача и над превозното средство;
* Запазване на информацията, свързана с автомобила, геолокацията на мобилното устройство, състоянието на шофьора и други данни;
* Извличане на нужната информация от запазените данни;
* Създаване на мобилно приложение, готово за употреба;
* Тестване и отстраняване на грешки и проблеми;
* Оптимизиране на работата на специфични алгоритми;
* Планиране на бъдещи актуализации с цел увеличаване на ефективността и функционалността на системата;

#### 4.3. Ниво на сложност на проекта − основни проблеми при реализация на поставените цели

* Проектът обхваща не само сферата на програмирането, но и разглежда комуникационната структура, използвана в автомобилите;
* Приложението има за цел да прихване набор от комплексни събития (заспиването на шофьора, нивата на подаване на гориво) в реално време, за да намали предпоставките за ПТП;
* “Driverify” прави анализ на извлечената информация и я показва по достъпен за потребителя начин;
* Запознаване в основите на езика Java, който стои в основата на създаването на мобилното приложението;
* Въведение в машинното самообучение и придобиване на основно понятие за приложението му;

#### 4.4. Логическо и функционално описание на решението – архитектура, от какви модули е изградено, какви са функциите на всеки модул, какви са взаимодействията помежду им и т.н.

##### 4.4.1 Модули

* Хардуерен модул:
  + OBD II адаптер – Elm 327 v2.1
  + Смартфон
* Софтуерни модули:
  + *Drive* – основната страница в приложението, която се грижи за състоянието на шофьора;
    - *ObdConnectionService* – услуга, следяща за връзката с превозното средство (приемане и предаване на информация);
    - *LocationServiceProvider*- услуга, следяща за промени в геолокацията на устройството;
    - *AmbientLightService* – услуга, следяща за промяната на осветеността в пространството превозното средство;
    - *DataControllerService* – услуга, отговаряща изцяло за записа на всичката информация, която бъде взета;
  + *Trip* – страница в приложението, която се грижи за пресмятане визуализиране на запазената информация;
    - *ChartTripAdapter* – фрагмент отговарящ за визуализацията на данните за икономично шофиране
    - *MapTripAdapter* – фрагмент отговарящ за изчертаването на картата и отбелязването на потенциално опасни зони от шофирането;

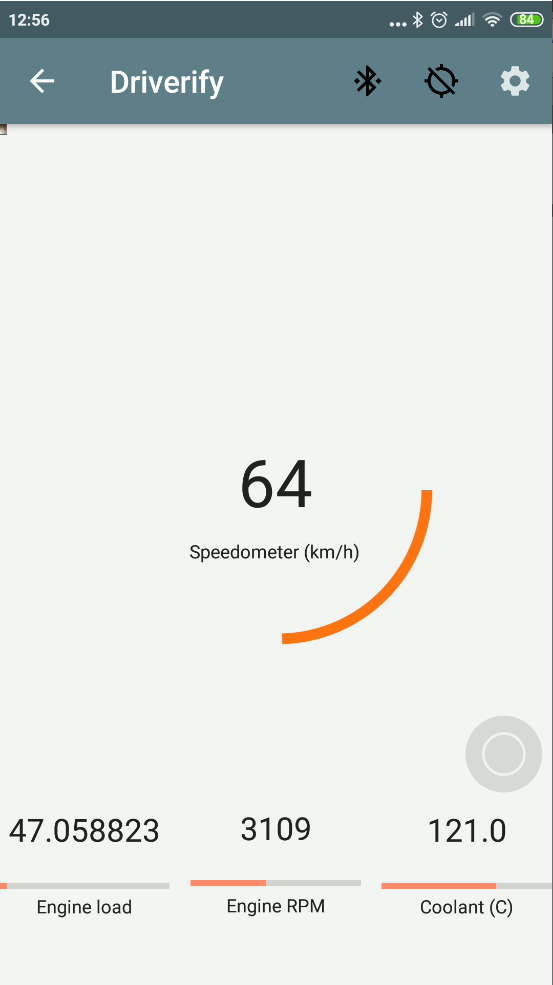
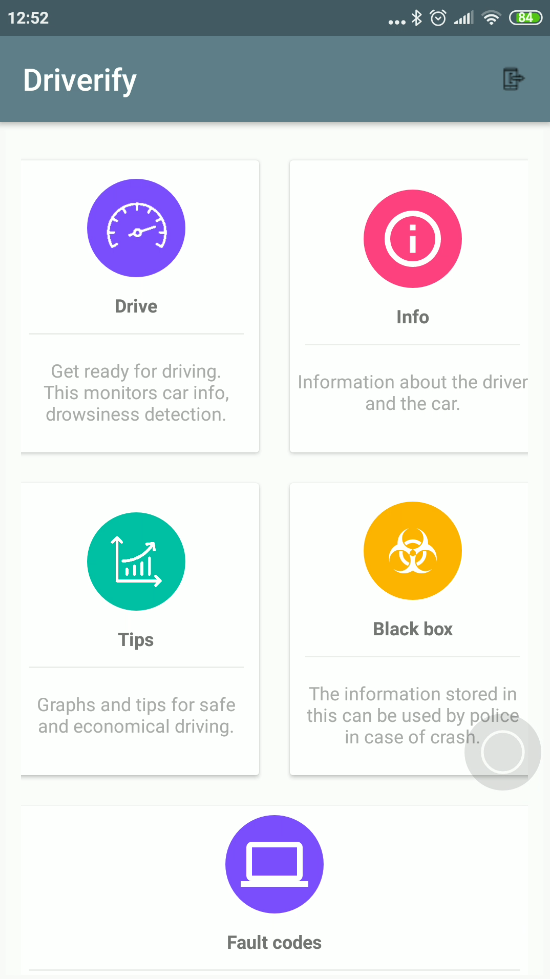
Мобилното приложение използва данни от превозното средство. Всеки автомобил след 2001 година при бензиновите и 2004 при дизеловите поддържа *OBD II (On-Board-Diagnostics)* протокол. *Elm 327* е устройството, което отговаря за извличането на нужната информация от автомобила. Устройството разполага с вграден мулти-протоколен интерпретатор, което помага за разчитането на суровите данни от превозното средство.

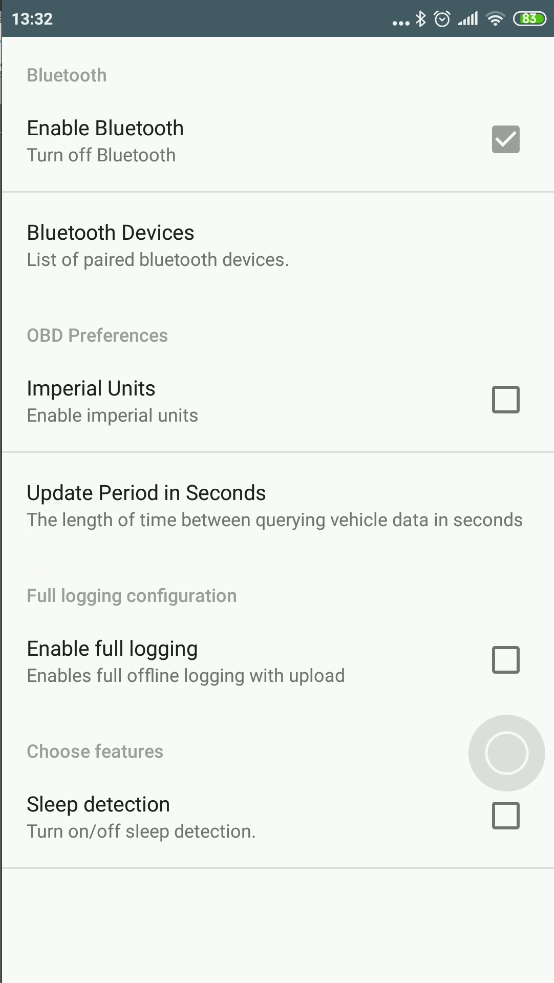
Условията за използването на смартфона са да е с *Android* и версията на операционната система да е по – нова от *Android Jelly Bean*. Устройството да разполага с предна камера (за бъдещи функционалности ще се изисква и задна камера), *GPS*, сензор за осветеност.

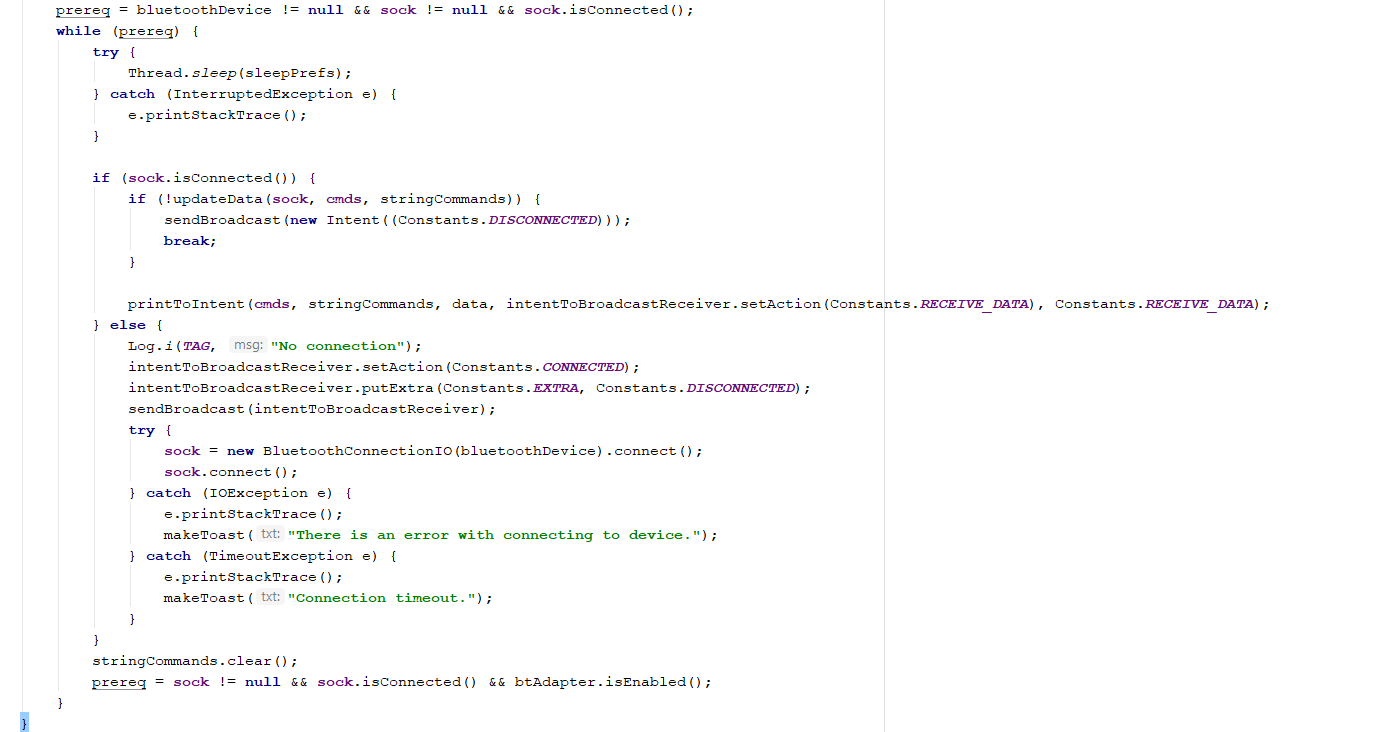
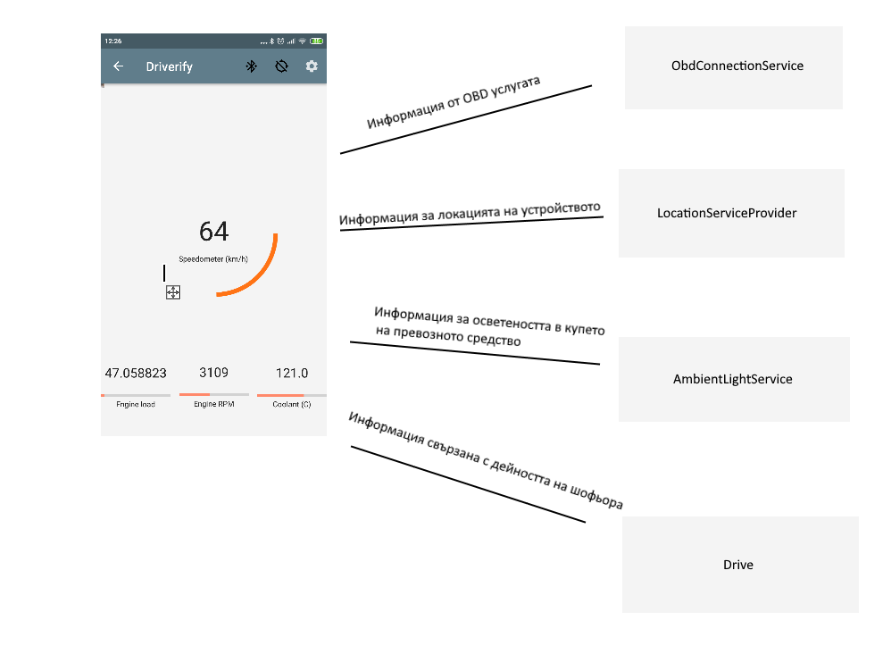
Архитектурата на приложението е изградена от отделни страници, които потребителят вижда, т.нар. *дейности* (*activities*). Всяка такава страница (*activity*), има свой „кръговрат на живота“, това са отделни методи, извиквани при промяна на статуса на дейността (затваряне на приложението, кликване на бутона за връщане назад и т.н.). Основна роля при реализацията на програмния продукт имат т.нар. *услуги* (*services*). Те прихващат/изпращат и запазват данните, които са извлекли.

#### 4.5. Реализация − обосновка за използвани технологични средства, алгоритми, литература, програмни приложения и др.

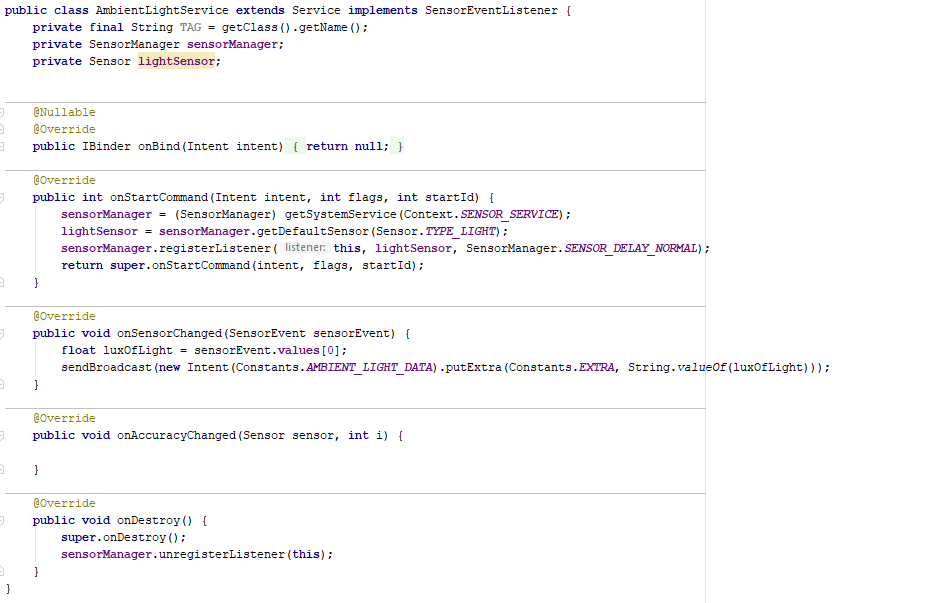
* Цялостното приложение е написано на програмния език *Java* като за реализацията му е използвана средата за разработка *Android Studio*. Основна причина за избора на тези технологии е широкото им разпространение и поддръжка.
* Алгоритъмът за прихващане на мимиките на човека и съставяне на изводи – дали той е заспал, буден или не е насочил вниманието си върху шофирането, е написан на *Python* чрез използване на *TensorFlow Lite*. Имплементиран е чрез *Google Management Services*. Използвани са *GMS*, защото имат много високи нива на оптимизация, а това от ключово значение при реализация с помощта на устройство с ниска изчислителна мощ.
* Визуализирането на данните се случва чрез използване на:
* *AnyChart –* безплатна библиотека, предлагаща широк спектър от функционалност;
* *Google Maps –* за визуализация на траекторията, по която се е движило превозното средство;

*Фиг. 1. Начално меню на приложението* *Фиг. 2.* ***Drive*** *activity*

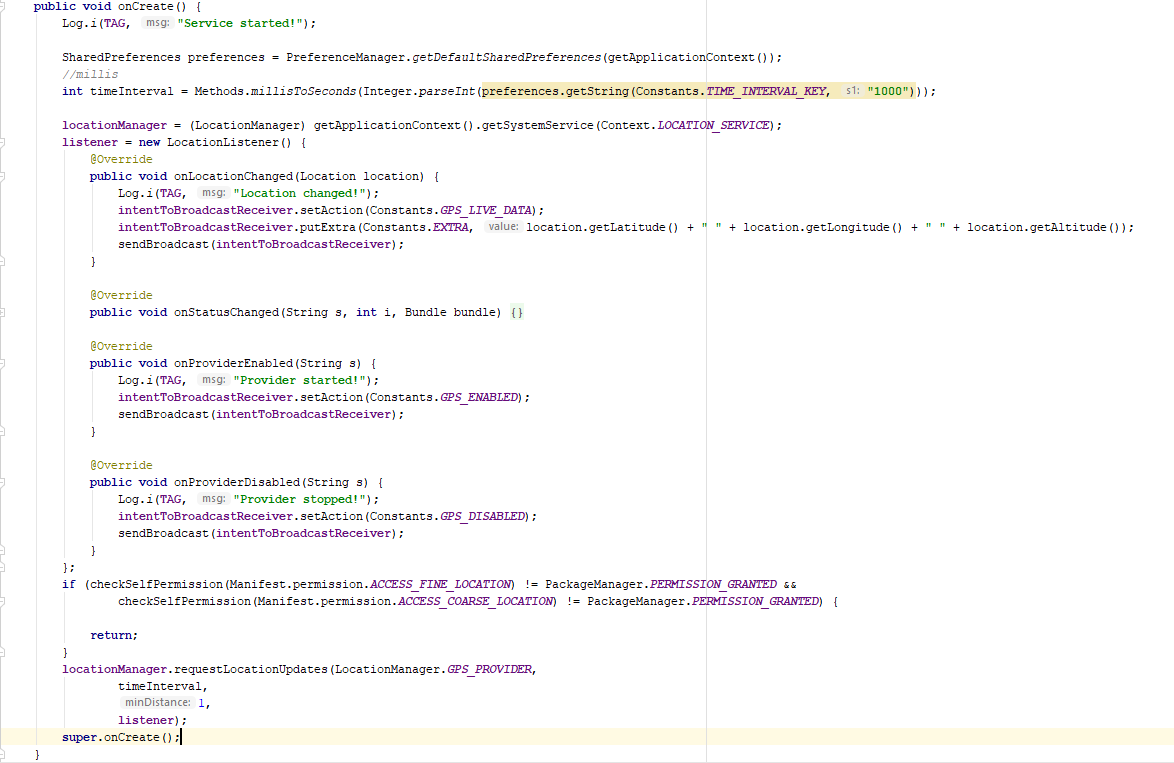
*Фиг. 2. Опциите на* ***Drive*** *activity*

*Фиг. 3. Схема на работната архитектура на дейността Drive*

*Фиг. 4. Изпращане на информация от ObdConnectionService*

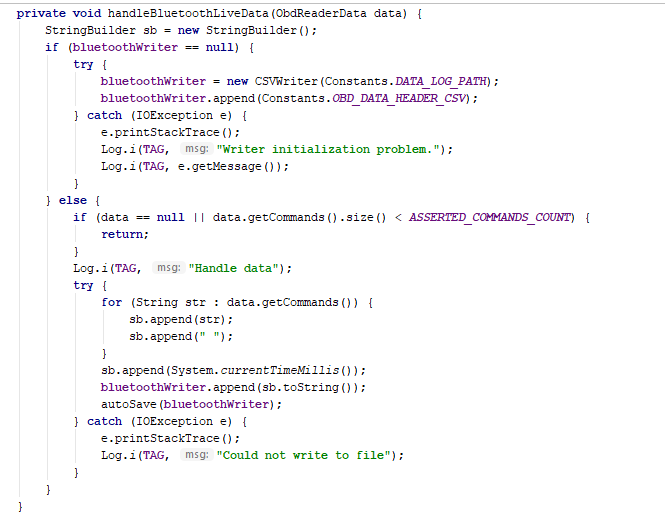


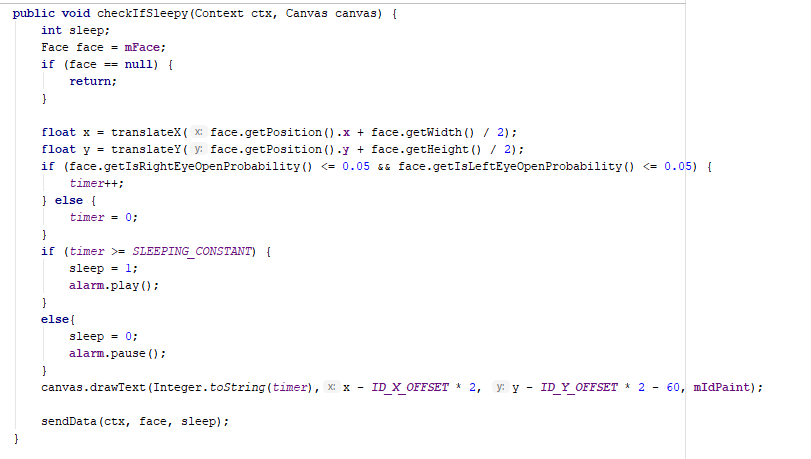
*Фиг 6. AmbientLightService*



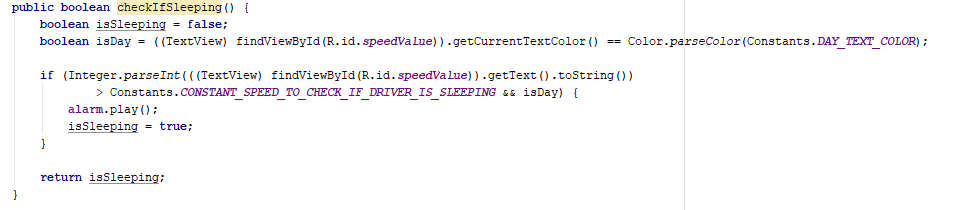
*Фиг. 5. LocationServiceProvider*

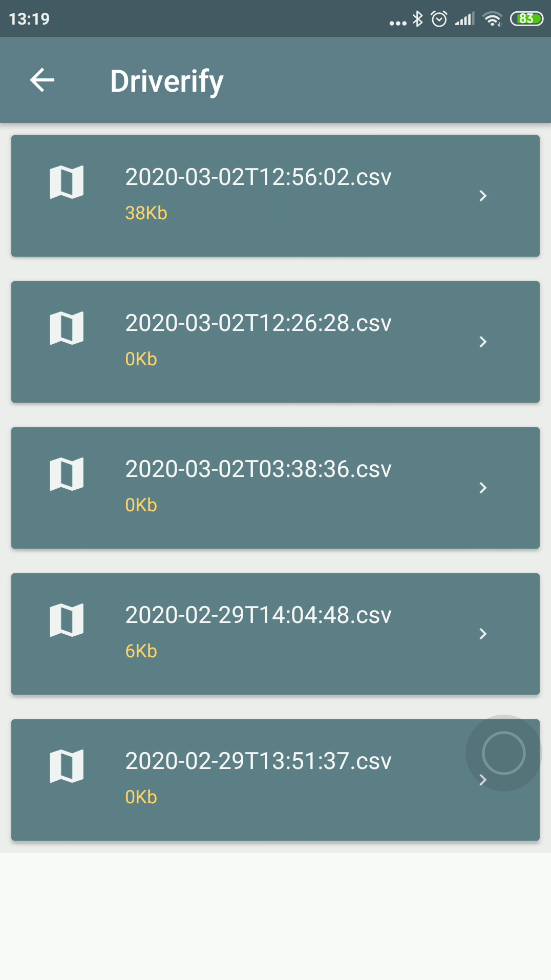
*Фиг. 5. Прихващане на информацията в Drive*

*Фиг. 6. Запазване на информацията от превозното средство(аналогично е и записването на другите идващи данни)*

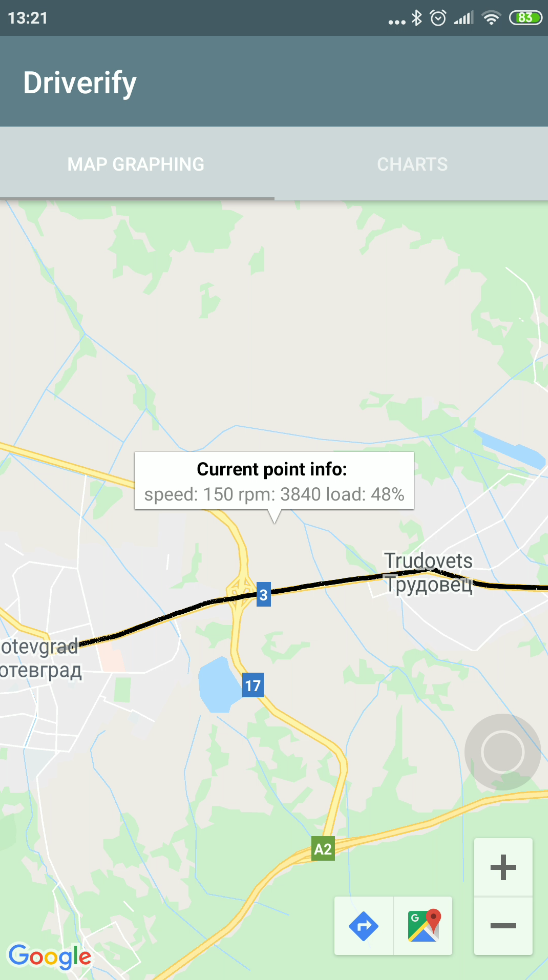


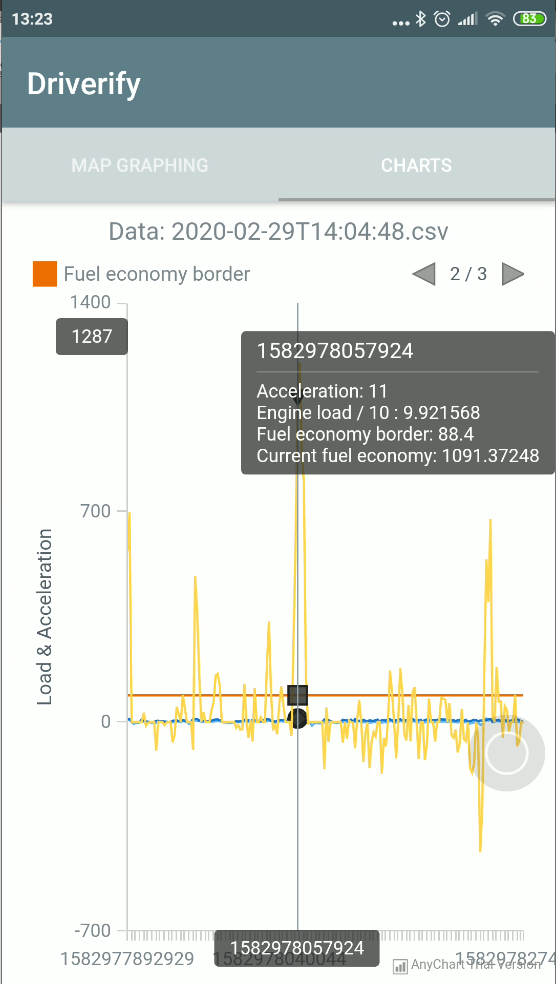
*Фиг. 7. Проверка дали водачът е заспал*

*Фиг. 8. Проверка дали шофьорът е насочил вниманието си върху движението*



*Фиг. 9. Trip activity – визуализиране на всички записи*

*Фиг. 10. Изчертаване на пътя и показване на данните от автомобила във всеки записан момент.*



*Фиг. 11. Изчертаване на графика, свързана с икономичното шофиране. Всичко по – голямо от f (current fuel economy) е неикономично шофиране.*

На съответната визуализация се наблюдават 4 функции.

* **Червената** функция е константата на икономичното шофиране. Тази стойност е получена от тестови записи на шофиране, предоставени от опитни шофьори с дълъг стаж зад волана.
* **Жълтата** функция е
* **Тъмносинята** функция показва ускорението на автомобила (Ускорението е първата производна на скоростта по времето. Графиката на скоростта е извлечена от превозното средство.).
* **Светлосинята** функция дава информация за натовареността на двигателя.

*//TODO: Да добавя снимки на кода на визуализирането на файловете*