**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.: +359876306495

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

Училище: Профилирана природо-математическа гимназия – гр. Ловеч

Клас: 12 „В“ клас

3. РЪКОВОДИТЕЛ:

Имена: Надежда Милкова Николова

Тел.: +359888196452

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 технети способности, когато те общуват с нас било то и от изолираното престрансто на личния си автомобил.

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

4.2. Основни етапи в реализирането на проекта:

* Проучване на проблема, работа на вече съществуващи системи и тяхната реализация;
* Поясняване на основните и на страничните функционалности на приложението;
* Подбиране на най- доброто място за поставяне на смартфона.
* Изучаване 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;
* Проучване възможностите на даден смартфон;
* Записване на различни видове данни, които да подпомогнат за разбиране състоянието на шофьора;
* Закупуване на OBD II адаптера;
* Имплементиране на алгоритмите за наблюдение над шофьора и над превозното средство;
* Запазване на информацията свързана с автомобила, геолокацията на мобилното устройство, състоянието на шофьора;
* Извличане на нужната информация от запазените данни;
* Създаване на мобилно приложение готово за експлоатация;
* Тестване и отстраняване на грешки и проблеми;
* Оптимизиране на работата на алгоритми;
* Планиране за 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 трябва да прихване набор от комплексни събития(заспиването на шофьора, нивата на подаване на гориво) в реално време за да редуцира предпоставките за ПТП;
* “Driverify” трябва да може да анализира и показва запазената информация;
* Запознах се с основите на езика Java който е главно използван в приложението.
* Трябваше да се въведа в машинното самообучение и да разбера как то работи;

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.

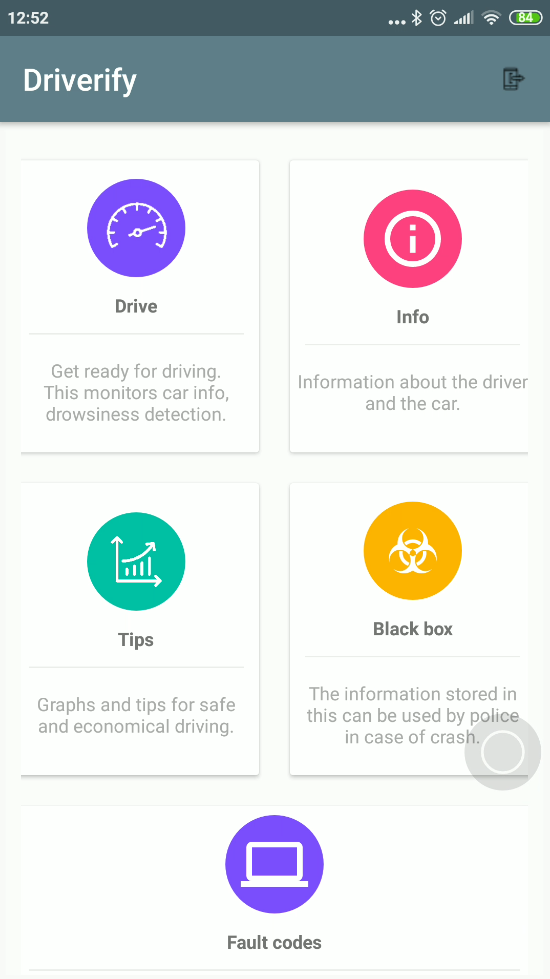
* Модули
* Хардуерен модул:
  + OBD II адаптер- Elm 327 v2.1
  + Смартфон
* Софтуерни модули:
  + “Drive”- основната страница в приложението, която се грижи за състоянието на шофьора;
    - ObdConnectionService- услуга следяща за връзката с превозното средство(приемане и предаване на информация);
    - 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.

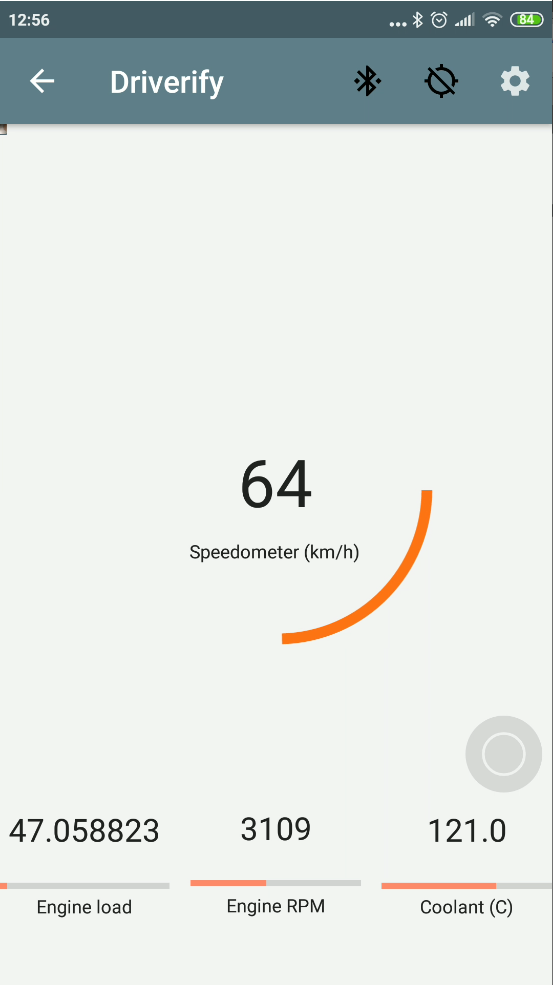
Условията за използването на смартфона са да е с „Android“ и версията на операционната система да е по- нова от „Android Jelly bean“. Устройството да разполага с предна камера(за бъдещи функционалности ще се изисква и задна камера), GPS, сензор за осветеност.

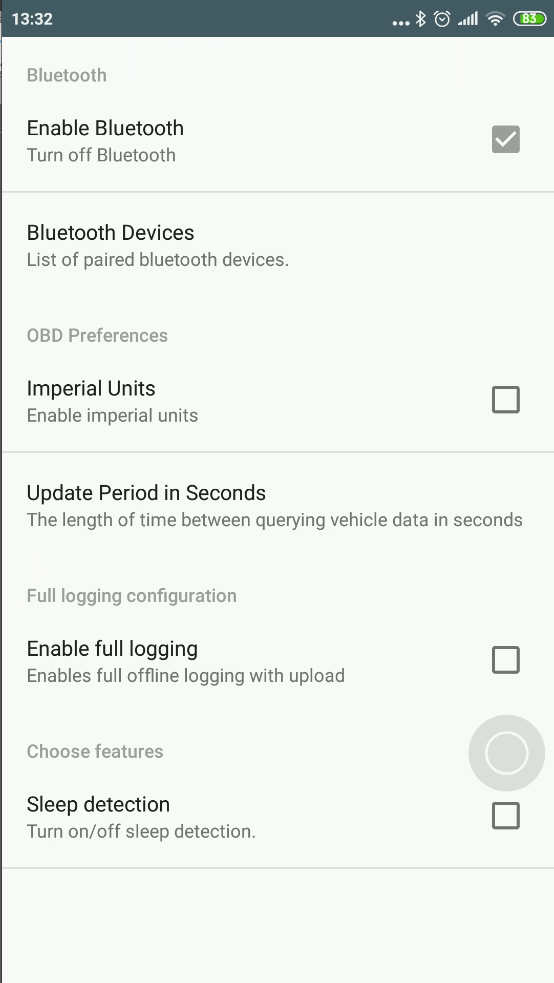
Архитектурата на приложението е изградена от отделни класове, които потребителят вижда- дейности(activities). Всяка активност, изпълнява свой „кръговрат на живота“, това са отделни методи извиквани при промяна на статуса на дейността(затваряне на приложението, кликване на бутона за връщане назад и тн.). „Движещата сила“ на програмния продукт са услугите(services). Те прихващат/ изпращат и запазват данните, които са извлекли.

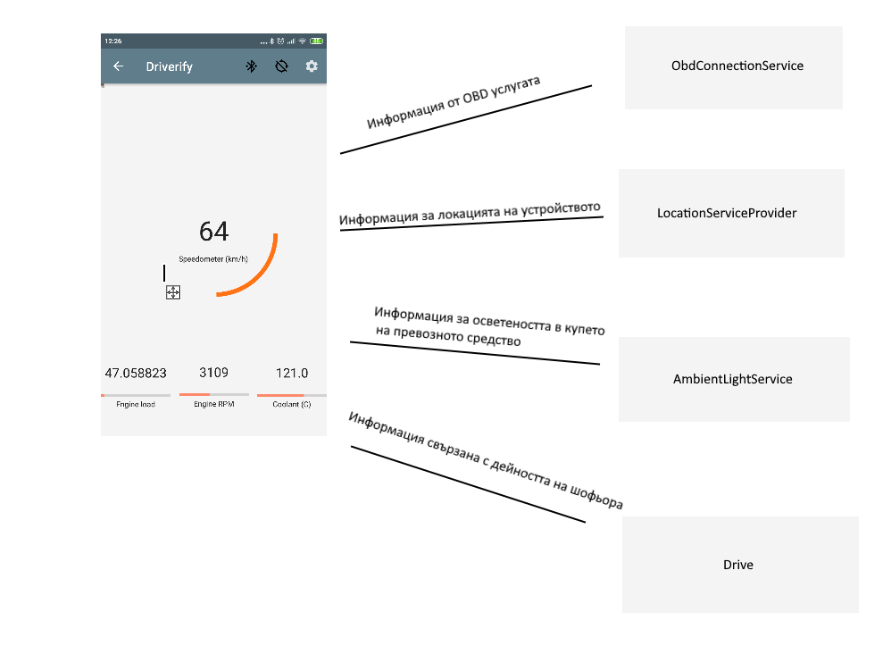
4.5. Реализация − обосновка за използвани технологични средства, алгоритми, литература, програмни приложения и др.

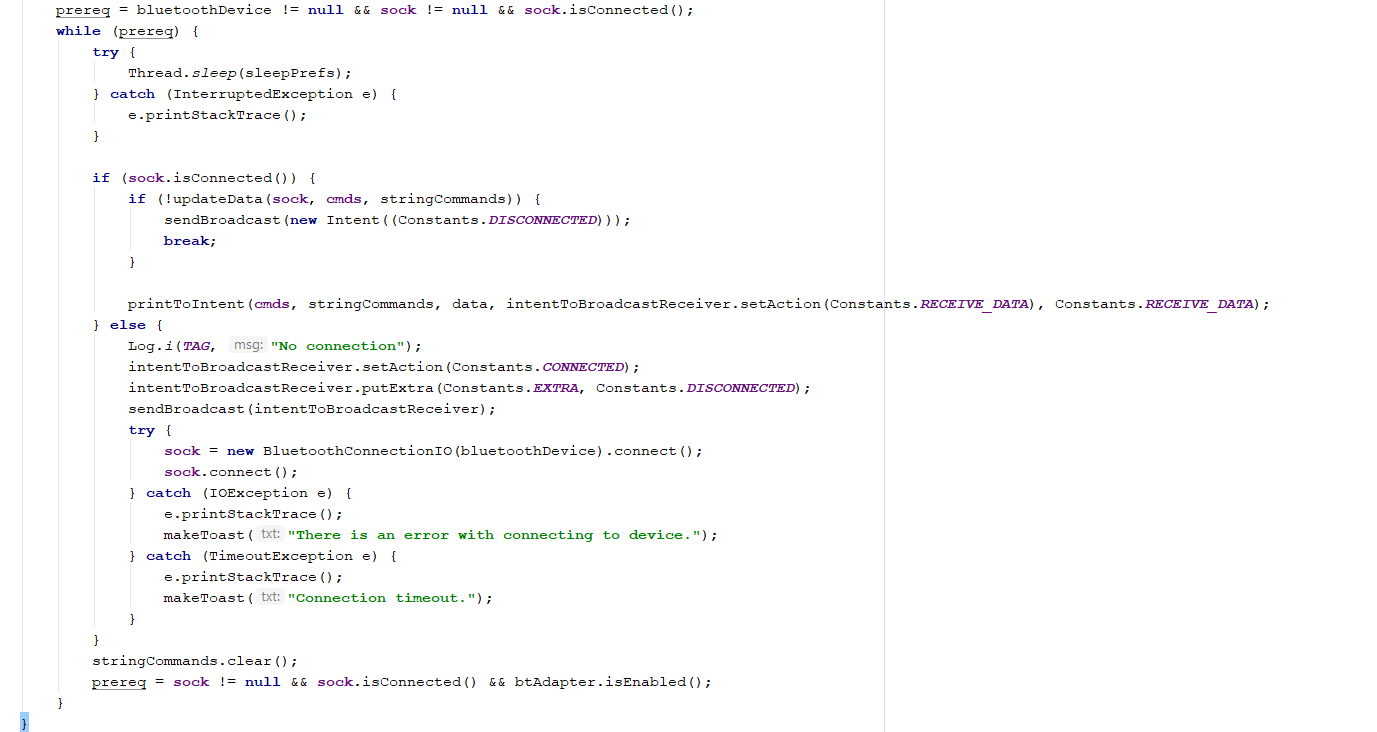
* Цялостното приложение е създадено чрез Android studio на програмния език Java. Защо точно този език? Защото е най- разпространен към момента;
* Алгоритъма за прихващане на мимиките на човека и извеждане на изводи дали той е заспал, буден или не е насочил вниманието си върху пътя е написан на Python чрез използване на Tensorflow Lite, имплементиран е чрез Google Management Services. Използвал съм GMS, защото имат много високи нива на оптимизация, а това за устройство с ниска изчислителна мощ играе огромна роля;
* Визуализирането на данните се случва чрез използване на:
* библиотеката AnyChart, защото е безплатна, лека, бърза и високофункционална;
* Google maps за визуализирането на пътя, по който е минало превозното средство;

*Фиг. 1. Началното меню на приложението*

*Фиг. 2. Drive activity*

*Фиг. 2. Опциите на Drive activity*

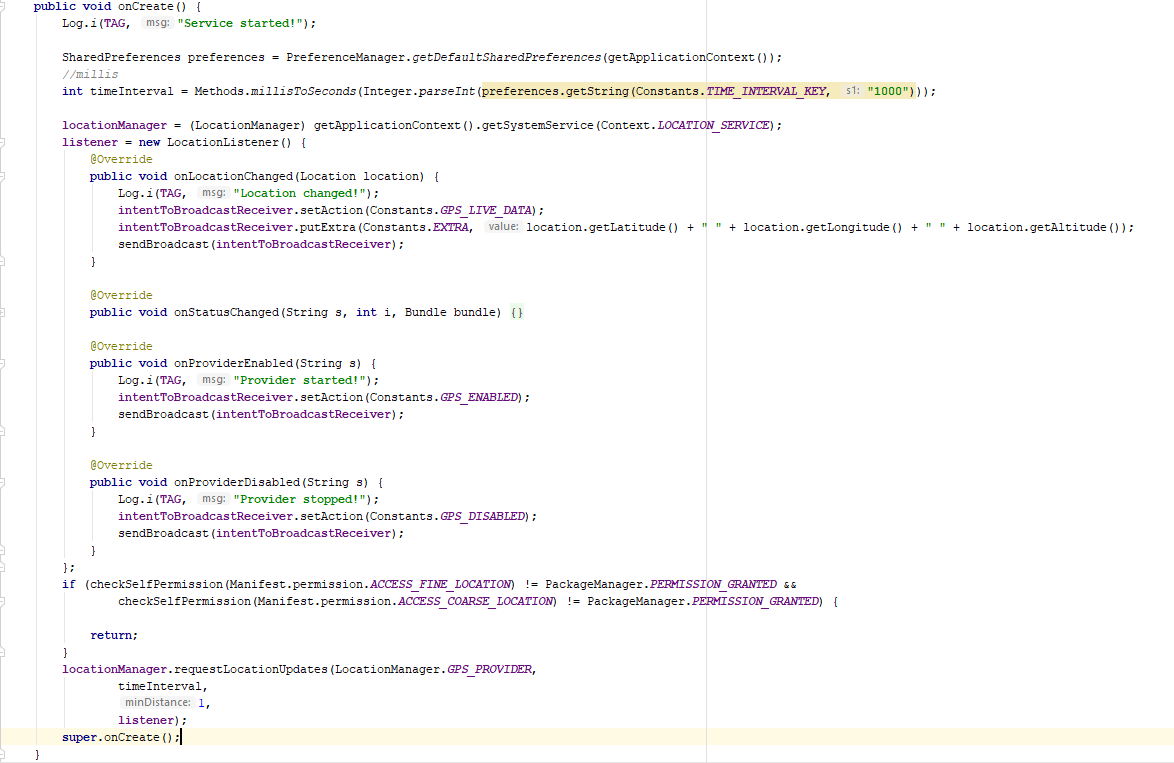
*Фиг. 3. Схема работната архитектура на дейността Drive*



*Фиг. 4. Изпращане на информация от ObdConnectionService*



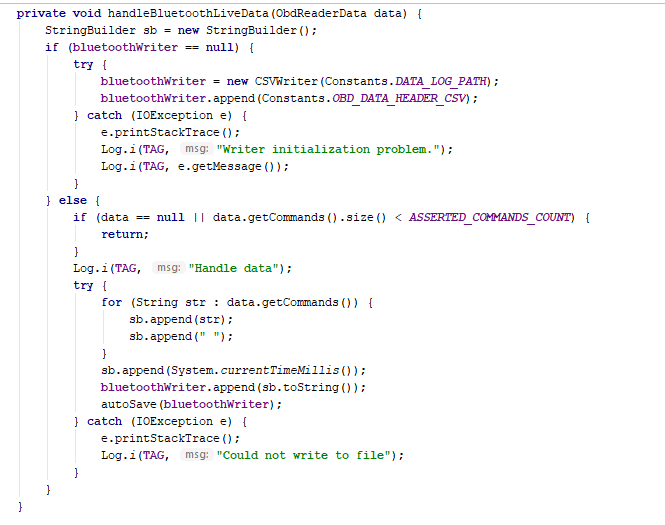
*Фиг 6. AmbientLightService*

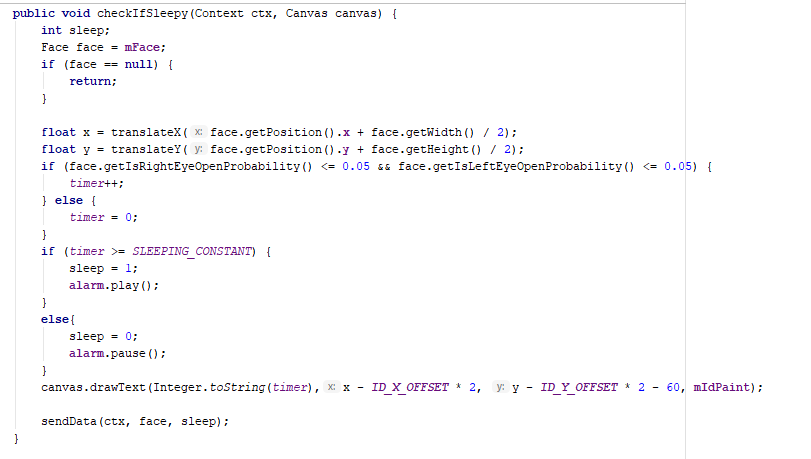


*Фиг. 5. LocationServiceProvider*

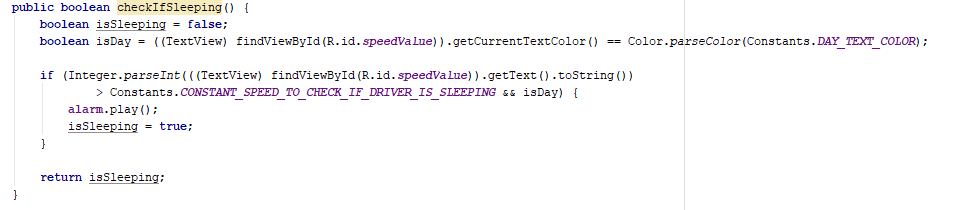


*Фиг. 5. Прихващане на информацията в Drive*

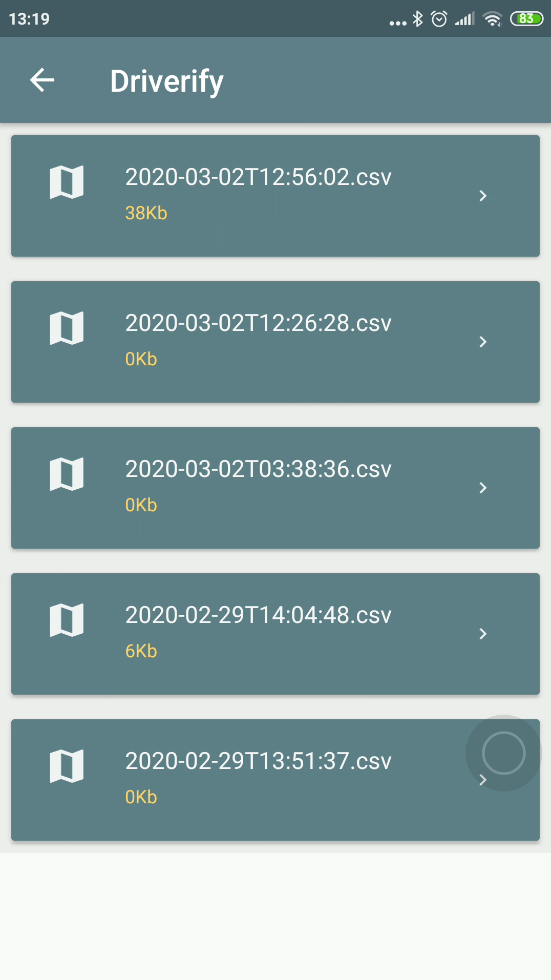
*Фиг. 6. Запазване на информацията от превозното средство(аналогично е и записването на другите идващи данни)*



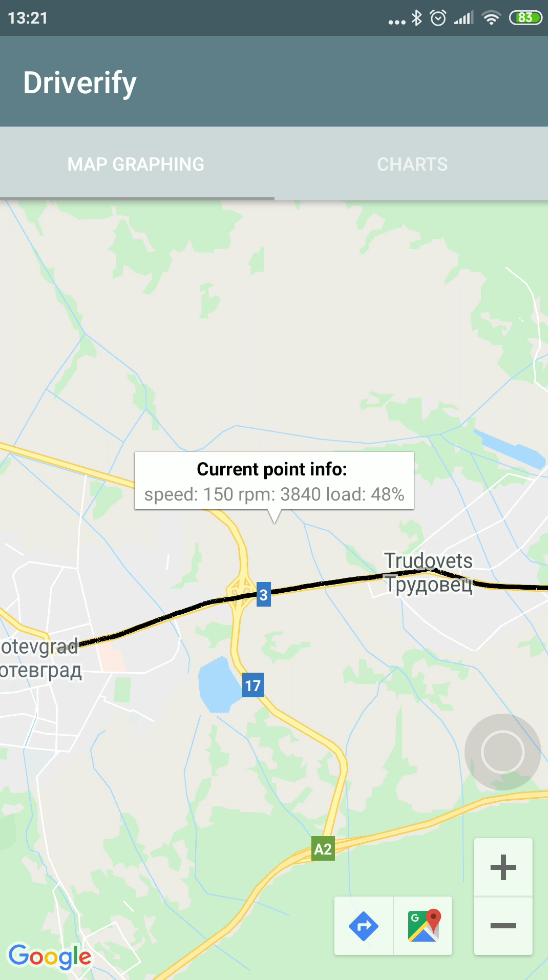
*Фиг. 7. Проверяване дали шофьора е заспал*

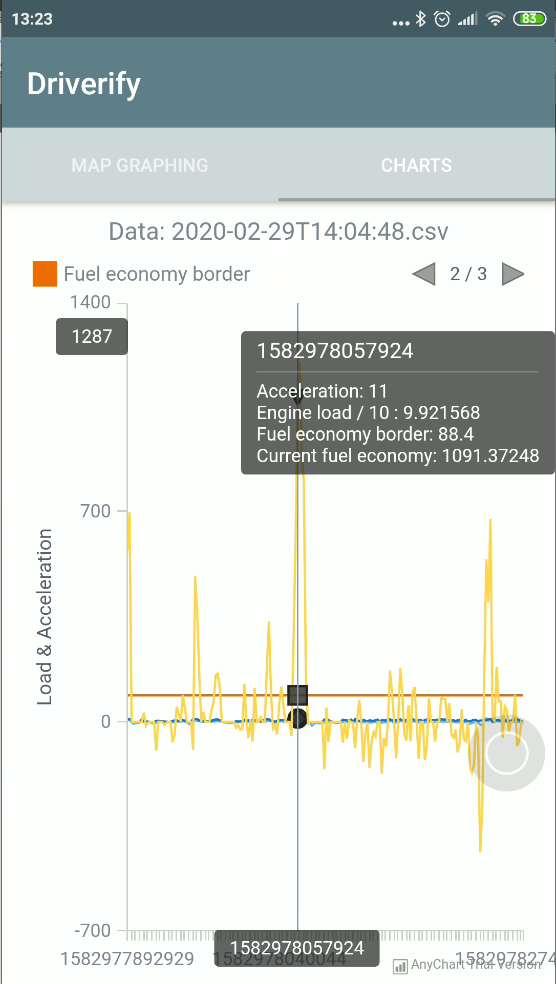


*Фиг. 8. Проверяване дали шофьора е насочил вниманието си върху пътя*



*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 не икономично шофиране.*

На съответното визуализиране на информацията се виждат 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 ускорението \* натоварването на двигателя
* Тъмносинята функция репрезентира ускорението на автомобила(ускорението е първата производна на графиката на скоростта. Графиката на скоростта е извлечена от превозното средство)
* Светлосинята функция репрезентира натовареността на двигателя

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