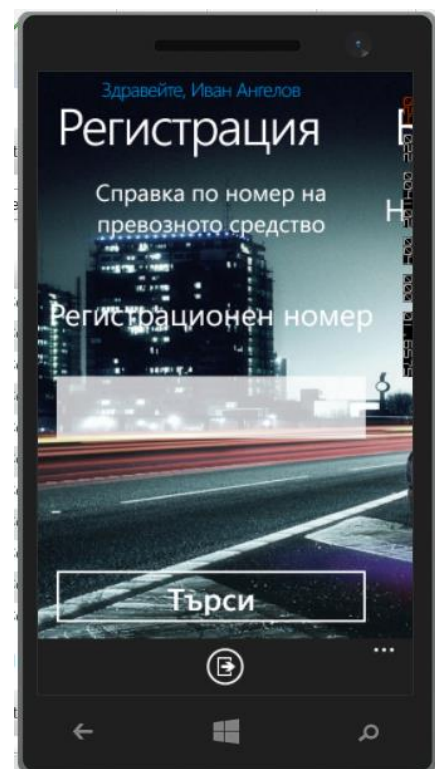
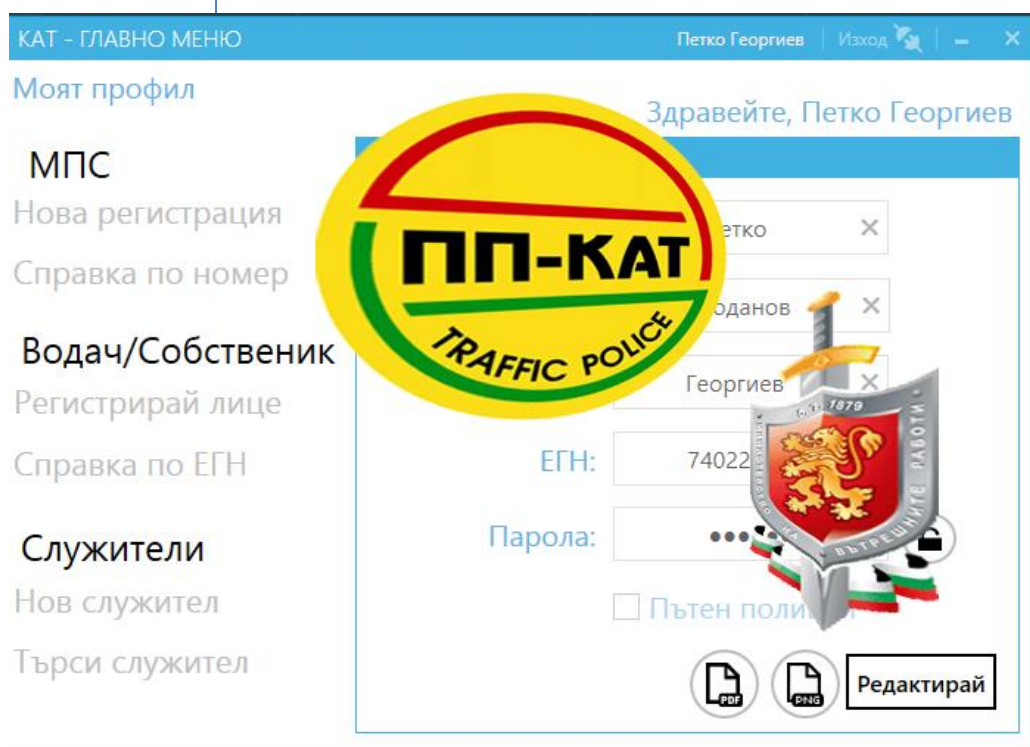


Програмни среди & Програмиране за мобилни у-ва

Проект

Информационна система за нуждите на КАТ

Да се проектира и реализира програма, която да съхранява и управлява информация за регистрираните автомобили. Да поддържа информация за извършените нарушения (засечени от камери, фишове, актове). Да се правят справки за неплатени глоби, за установяване собственост на автомобил, други.



Ръководител:

гл. ас. д-р Антония Ташева

ас. инж. Боян Петров

ФКСУ, гр. 42, Фак № 121213099

Изготвил:

Георги Челенков

ФКСУ, гр. 42, Фак № 121213099

ТУ-София

18.05.2015 г.

Съдържание

Увод	1
Анализ на съществуващи разработки	2
• Officer™ Traffic	2
• CrimeStar™	3
• eForce™	5
Проектиране (накратко)	6
Проектиране (в детайли)	7
• Кой ще използва продукта?	7
• Какви данни ще се използват?	8
• Достъп до функционалностите	10
○ Концептуален дизайн - десктоп	10
○ Концептуален дизайн – мобилна версия	21
Реализация	26
• База данни	26
• Десктоп архитектура	29
• Мобилна архитектура	30
Потребителско ръководство (резултати)	33
• Desktop use-cases	33
• Mobile use-cases	37
Заклучение	39
Използвани източници	40
Приложение	41

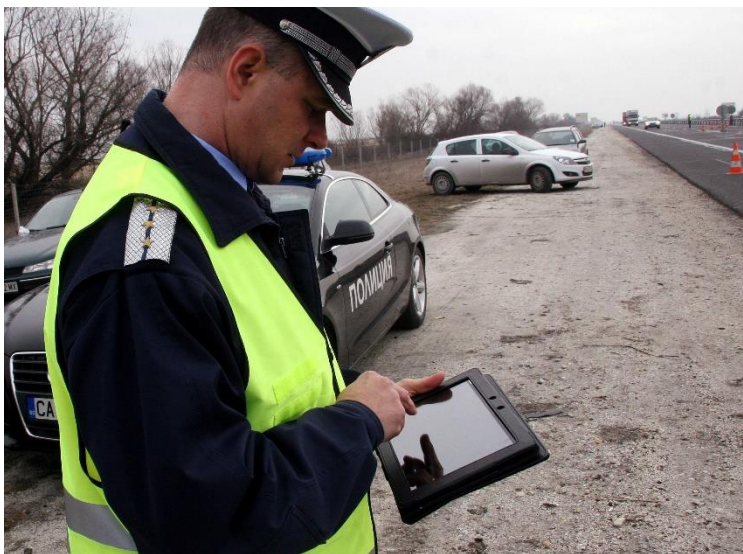
Увод

В основата на съвременния свят се намира *‘информацията’* и ключът за успеха се крие в правилната и обработка. Това твърдение важи в пълна сила за всички държавни структури и институции.

В структурите на КАТ се работи с огромни бази данни, чиито ефективно използване е критично за функционирането на цялата институцията. Информацията е силно динамична и подлежи на промени постоянно. Голяма част от настоящата обработка представлява ръчно въвеждане на голям обем данни под формата на текст от упълномощено лице. Работата по въвеждането на информацията, нерядко, се свежда, дори, до ръчното изписване на информацията на хартиен носител и последващото и въвеждане в база данни. Сам по себе си, този подход е твърде неефективен, тъй като неколкотократно влияние на човешкия фактор значително увеличава потенциалната възможност за възникване на грешка, както и необходимото време за извършване на дадена задача.

Основно средство при работа с подобен тип информация са т. нар. *‘справки’*, чрез които се проверява валидността на данните и се обработват потребителски заявки. Обикновено, упълномощено лице (наричано още *‘Администратор’*) в даден момент обработва заявка направена от един-единствен конкретен потребител (Пример: Администратор прави справка за неплатени глоби на даден водач/отбелязва като платена дадена глоба). Настоящото решение цели да реши проблема с последователната обработка на данни, като направи възможна паралелната, сигурна и бърза обработка на данни, при едновременното обслужване на няколко потребители от едно лице. Централизираното управление, чрез приложението, помага за повишаването на консистентността на данните и ефективното им използване.

В допълнение към администраторското десктоп приложение, налично е и мобилно приложение, което цели да улесни и , до голяма степен, автоматизира дейността на полевите (пътни) полицаи при генерирането и регистрирането на документи за нарушения, съставени на място, както и да опрости значително справките относно автомобилна регистрация, данни за водач, застрахователна информация и др.



Анализ на съществуващи разработки

По отношение на сигурността на пътя, съществуват множество, както десктоп, така и мобилни системи. Взаимодействието и колаборацията между органите на реда и съвременните технологии повишава значително ефективността на работата. Сред по – популярните системи са:

- [Officer™ Traffic](https://www.gtechna.com/product/officer-traffic-suite/) - <https://www.gtechna.com/product/officer-traffic-suite/>



Фиг. 1 - Officer™ Traffic

Продуктът представлява сложна система, която според създаделите – увеличава значително качеството на услугите, свързани със сигурността, като същевременно намалява разходите. Сред потребителите на приложението се нареждат огромен брой полицейски управления предимно в САЩ.

Софтуерът е изграден на модулен принцип, като голяма част от модулите са платени. Дадено полицейско управление може да се снабди с определен модул, ако има реална нужда от него.

Продуктът таргетира, както десктоп среди, така и мобилни устройства с цел – по – голяма полева гъвкавост.

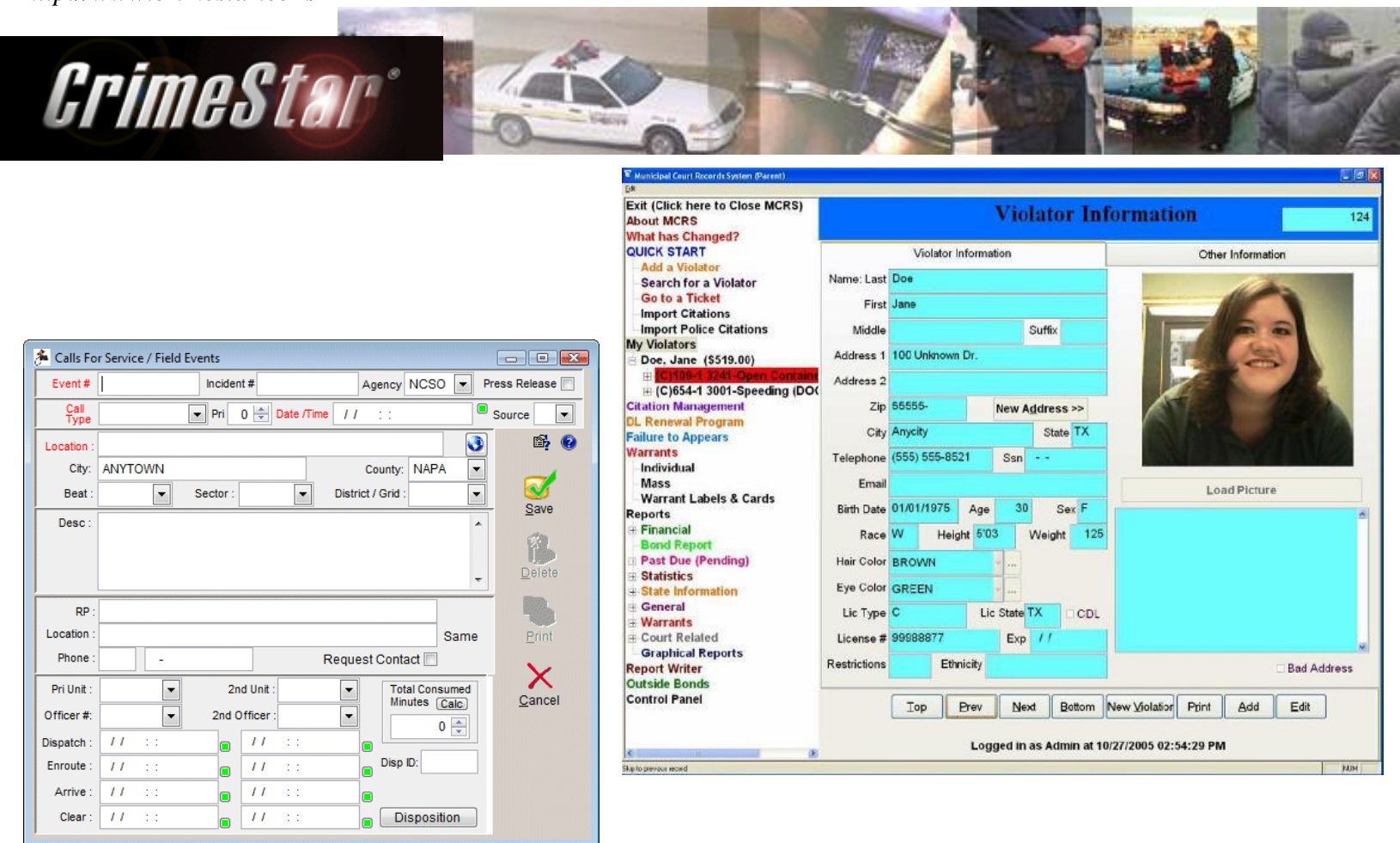
Сред **плюсовете** на **Officer™ Traffic** системата се нареждат:

- От 15 % до 30 % по – бърза обработка на фишове/актове/глоби;
- Ограничаване на грешките при съставяне на документи;
- Паралелна обработка на множество нарушения;
- Повишаване на събираемостта на средствата от глоби;
- Автоматично сканиране на регистрационни номера и съставяне на заявки;
- Обединява отдел “Пътна полиция” с отдел “Паркинги” чрез интегрираната ‘pay-per-ticket’ (плащаш това, което ползваш) система.

Слабости в системата, на този етап, се забелязват в:

- Способността за гъвкаво адаптиране на системата при вече установени работни процеси, различни от типичните за **Officer™ Traffic**;
- Висока цена на отделните модули;
- Сложен и труден за ориентиране интерфейс на мобилните системи;
- Фокусирана предимно върху нарушения при паркиране и сигнализирането им;
- Невъзможност да се правят детайлни справки за личните данни на водачите.

Друг популярен продукт в сферата на пътната сигурност и редовност е системата **CrimeStar™** - <http://www.crimestar.com/>



Фиг. 2 - CrimeStar™ система

CrimeStar™ е предпочитан от традиционалистите продукт, разполагащ с богата функционалност, фокусирана предимно върху полеви операции, съставяне на сложни текстови документи и анализи свързани със задържания и арести.

Продуктът е на пазара от 1989 година, като не се наблюдават особени промени по интерфейса на системата и отделните модули.

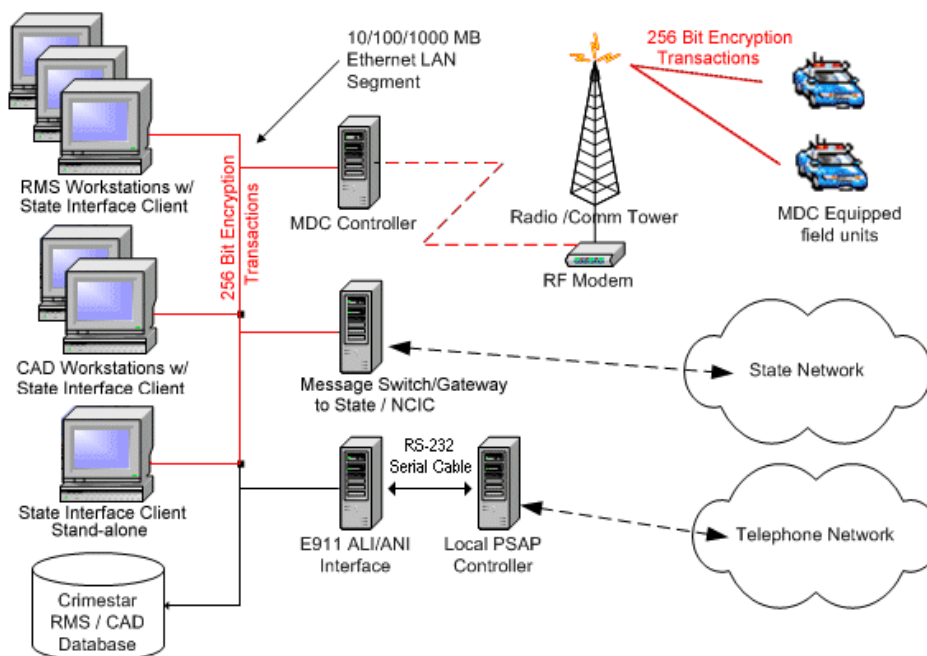
Предлага се като една основна система, към която впоследствие е възможно да се закупи някои от наличните модули – Подробни карти за САЩ (PMF Module – Premium Map Files), MDC (Mobile Digital Communicator), CAD (Computer Aided Dispatch) и още няколко.

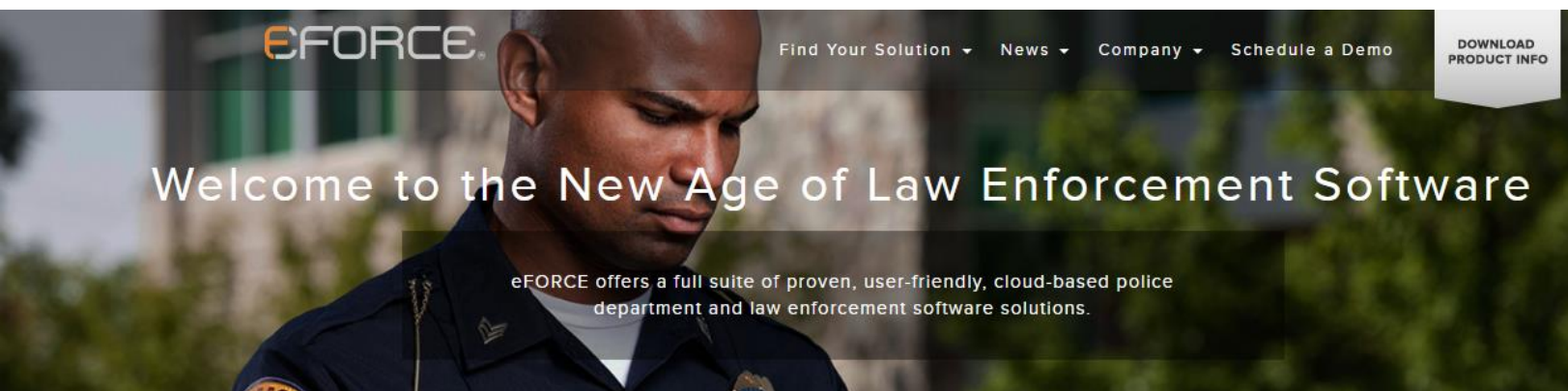
Основни **предимства**, които правят системата привлекателна за специалистите в сферата, въпреки остарелия дизайн са:

- Сложна Search система, на базата на местоположения, превозни средства, документи и др.;
- Над 200 изградени шаблона за генериране на документи в различни формати и готови за принтиране;
- Контрол на достъп до отделните модули и документи;
- Лесна комуникация и споделяне на данни с други агенции;
- Възможност за изграждане на системата като пълноценна **LAN** система (*Professional Edition* версията на софтуера) или **Client/Server** архитектура (*Enterprise Edition* версията).

Сред слабите черти на **CrimeStar™** са:

- Морално остарелия интерфейс от края на миналия век;
- Липса на мобилна технология;
- 32 битова архитектура на приложението;
- Необходимо е да бъде преминато обучение, за да бъде използван пълноценно продукта. (сложна работа със софтуера);
- Техническата поддръжка е в сила единствено за територията на САЩ.





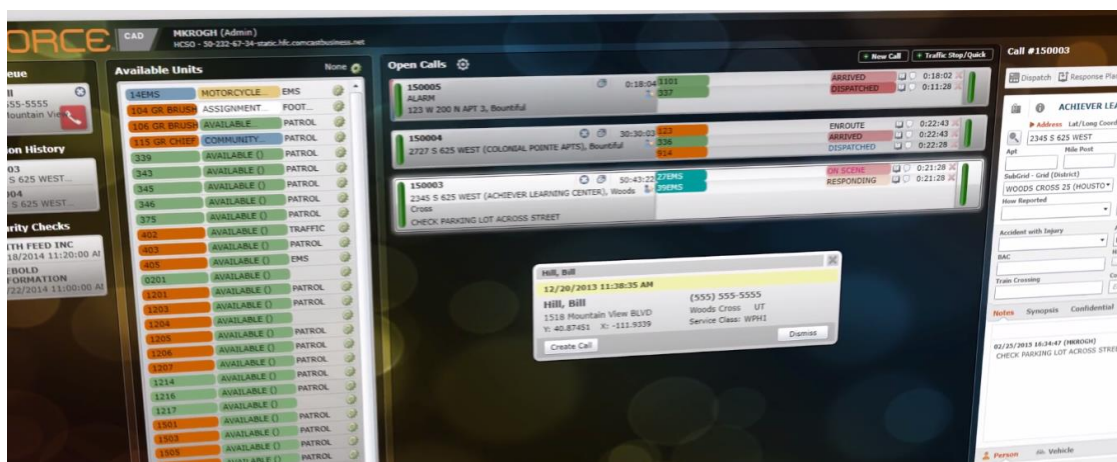
Фиг. 3 - eForce™

eFORCE® Public Safety Software Suite е пълноценно портфолио от уеб-базирани решения, включващи:

- Computer-Aided Dispatch (CAD) функционалност;
- Records Management Software (RMS) функционалност;

Бързонабиращата скорост **eFORCE®** система притежава следните **предимства** пред другите системи в този клас:

- Computer-Aided Police Dispatch;
 - Records Management Software (RMS);
 - Civil Process Software;
 - Transit Fare Enforcement Software;
 - Municipal Court Software;
 - Mobile Police Software – iOS (Iphone и Ipad) операционна система;
 - Jail Management Software (JMS);
 - Automatic Vehicle Locator (AVL) – автоматично проследяване на автомобили;
 - e-Citations Software.
- Високи такси на абонаментни планове;
- Липса на гъвкавост при интеграция със съществуващи системи.



Проектиране (накратко)

Системата разполага със съвременен интерфейс, таргетиращ Windows-базирани платформи, както и мобилни операционни системи (Android, iOS, Windows Phone). Високата степен на портативност е постигната благодарение на платформата – *Xamarin*, фокусирана върху изграждането на мултиплатформени (*cross-platform*) приложения. Тъй като приложението е все още **Бета** версия, е изграден единствено Windows Phone UI, като съвсем скоро се очаква, на базата на вече създадената мултиплаформена архитектура (с помощта на *MVVM Cross* фреймуърка) да бъде изработен UI за Android и iOS.

За създаването на десктоп версията на приложението са използвани типични библиотеки, принципи и похвати, присъщи на програмните модели на продуктите на *Microsoft*. Графичният интерфейс се базира на **WPF (Windows Presentation Foundation)**.

От архитектурна гледна точки, както десктоп версията, така и мобилната версия са изградени на базата на MVVM (Model-View-ViewModel) фреймуърк, като съществуват различия, в зависимост от конкретно-използвания шаблон за правилно изграждане на MVVM архитектурата. (Пр. MVVM Cross Framework – при мобилното приложение.)

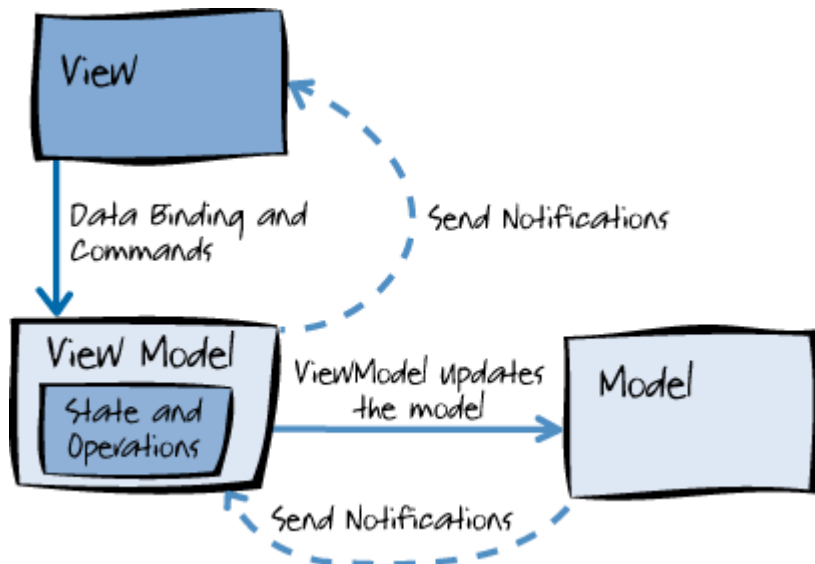
Основната мотивация за използване на Model-View-ViewModel многослойната архитектура в настоящото приложение се изразява в:

- Осигуряване на намалена свързаност (Tight coupling), ясно разделение между бизнес логика, логика на приложението и графичен интерфейс, повишаване възможността за преизползваемост на кода;
- MVVM е типичен шаблон за Xaml платформите, тъй като позволява ефективно и гъвкаво ‘свързване на данните’ с графичния интерфейс, посредством ‘Data Binding’ механизмите;
- Улеснено последващо тестване на приложението, поради ясното разделение на интерфейса от останалите компоненти.

Продуктът представлява сложна система, която според създаделите – увеличава значително качеството на услугите, свързани със сигурността, като същевременно намалява разходите. Сред потребителите на приложението се нареждат огромен брой полицейски управления предимно в САЩ.

Софтуерът е изграден на модулен принцип, като голяма част от модулите са платени. Дадено полицейско управление може да се снабди с определен модул, ако има реална нужда от него.

Продуктът таргетира, както десктоп среди, така и мобилни устройства с цел – по – голяма полева гъвкавост.



Фиг. 4 – Опростен архитектурен (MVVM) модел на приложенията

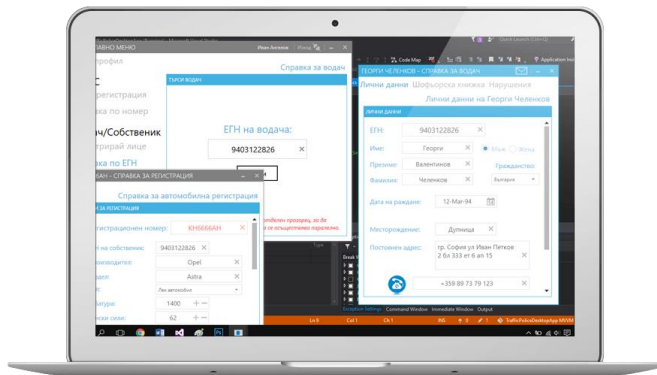
От гледна точка на съхранение на данни и модел, приложенията функционират с помощта на **MySQL** база данни, намираща се на отдалечен сървър.

Комуникацията се осъществява чрез **WCF** (Windows Communication Foundation) приложение, което представлява междинния слой. (между базата и самото приложение)

Проектиране (в детайли)

- *Кой ще използва продукта?*

Десктоп версията на приложението е проектирана да бъде изцяло в полза на служителите от отдел “Пътна полиция” в дадено районно управление. Всеки служител ще разполага със **служебен номер** и **парола**, чрез които получава достъп до системата. След успешен вход, служителят ще разполага с набор от инструменти, спомагащи за осъществяване на различни справки (подробна информация за водач, друг служител, регистрационен номер). Предвижда се да има и различни нива потребители, които разполагат с различна степен на достъп до общите ресурси. За момента налични са два основни типа служители – пътни полицаи и административни служители.



Мобилната версия е пригодена за полевите служители (пътни полицаи). Тя разполага със семпъл интерфейс, в който са въплътени основните функционалности, необходими на един полицаи да извърши пълноценно ежедневните си служебни задължения, или иначе казано – да прави справка за водач, регистрация и да съставя документи за извършено нарушение.

- *Какви данни ще се използват?*

Модела на приложението е изграден от няколко основни модула, описващи:

- Потребител (Полиций/Административен служител);
- Водач;
- Автомобилна регистрация;
- Нарушение.

За пълноценното функциониране на всеки от модулите, е необходимо детайлното му описание.

При осъществяване на операциите – ‘Справка за служител’ и ‘Регистрация на служител’ се използват следните описания (*‘Служител’*):

- Служебен номер на служителя;
- Име, презиме и фамилия;
- Информация за типа на служителя (полицей/администратор);
- Парола за вход в системата.

За функционалността свързана с обработката на данни за водач са предвидени данните(*‘Водач’*):

- Лични данни:
 - ЕГН;
 - Име, презиме и фамилия;
 - Пол;
 - Гражданство;
 - Дата на раждане;
 - Месторождение;
 - Постоянен адрес;
 - Телефонен номер;
 - Email.
- Данни свързани със статуса на ‘шофьор’ на лицето
 - Дата на издаване на шофьорска книжка;
 - Дата на изтичане на валидността на документа;
 - Орган, издал документа;
 - Налични контролни точки;
 - Данни за притежавани от лицето категории (*използван е шаблона на реална шофьорска книжка*)
 - Дата на издаване на категорията;
 - Дата на изтичане на валидността на категорията;
 - Ограничение;

13.	9.	10.	11.	12.
	AM	08.08.07		
	A1		
(14.)	A2		
	A		
	B1	08.08.07		
	B	08.08.07		
	C1		
	C		
	D1		
	D		
	BE		
	C1E		
	CE		
	D1E		
	DE		
	T1M		
12.	T1T		

Фиг. 5 – 'Категории' ('гръб' на документа)

(10) – Дата на издаване

(11) – Дата на изтичане

(12) - Ограничение

За пълноценната работа на продукта, е изграден и модел на '**Регистрация**'. Той е изцяло съобразен с данните от големия талон на превозното средство. Негови параметри са:

- Регистрационен номер;
- ЕГН на водач-собственик;
- Дата на първа регистрация на превозното средство;
- Дата на настояща регистрация;
- Технически х-ки на превозното средство
 - Производител;
 - Модел;
 - Цвят;
 - Тип на превозното средство;
 - Кубатура;
 - Мощност;
 - VIN.
- Застрахователна информация
 - Винетка
 - Наличие на винетка;
 - Валидност.
 - Застраховка 'Гражданска отговорност'
 - Наличие;
 - Дата на сключване;
 - Дата на изтичане;
 - Застраховател
 - Застраховка 'Автокасco' (аналогични данни на Застраховка 'Гражданска отговорност')

Структурата включва и 'Нарушение', като данните са съобразени с наличните и използвани шаблони за съставяне на документи за нарушение.

- Служебен номер на нарушението;
- Служебен номер на служителя – съставител;
- ЕГН на извършителя;
- Дата на извършване на нарушението;
- Дата на регистрация на нарушението;
- Местопроизшествие (Координати – географска ширина и дължина);
- Описание;
- Възражение.

Широкия набор от функционалности на системата е постигнат, благодарение на базата на няколко основни *бизнес процеса*:

- Логване на потребител (полицай/администратор) в системата;
- Извличение на данни за потребител;
- Регистрация на водач/собственик;
- Справка за водач/собственик;
- Добавяне на нова ‘Регистрация’ (по смисъла на закона) на превозно средство;
- Подробна справка за ‘Регистрация’;
- Съставяне на нарушение на водач;
- Обработка на нарушението (Справка за нарушение/премахване/генериране на документ);
- Изпращане на уведомление до водач (под формата на имейл).

Някои от процесите са типични за десктоп платформата с таргет група – *служители* , докато други са характерни за мобилната платформ, подпомагащи дейността на пътните полицаи (Справка за регистрация, за водач, съставяне и регистриране на нарушение на водач).

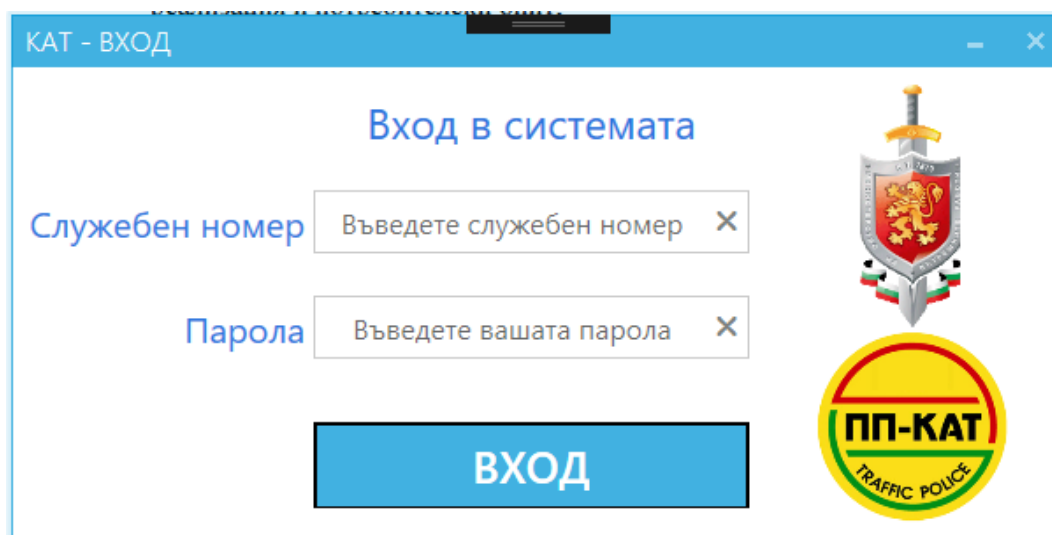
- *Достъп до функционалностите*

Системата разполага със съвременен и интуитивен интерфейс изграден на базата на WPF (за десктоп) и WindowsPhone Silverlight 8.1 (за Windows Phone). Предстои мултиплатформената Xamarin архитектура да намери своята имплементация и за Android и iOS (посредством Xamarin.iOS и Xamarin.Android).

Следващите фигури представят в детайли концепцията на потребителския интерфейс, както за десктоп, така и за мобилни устройства.

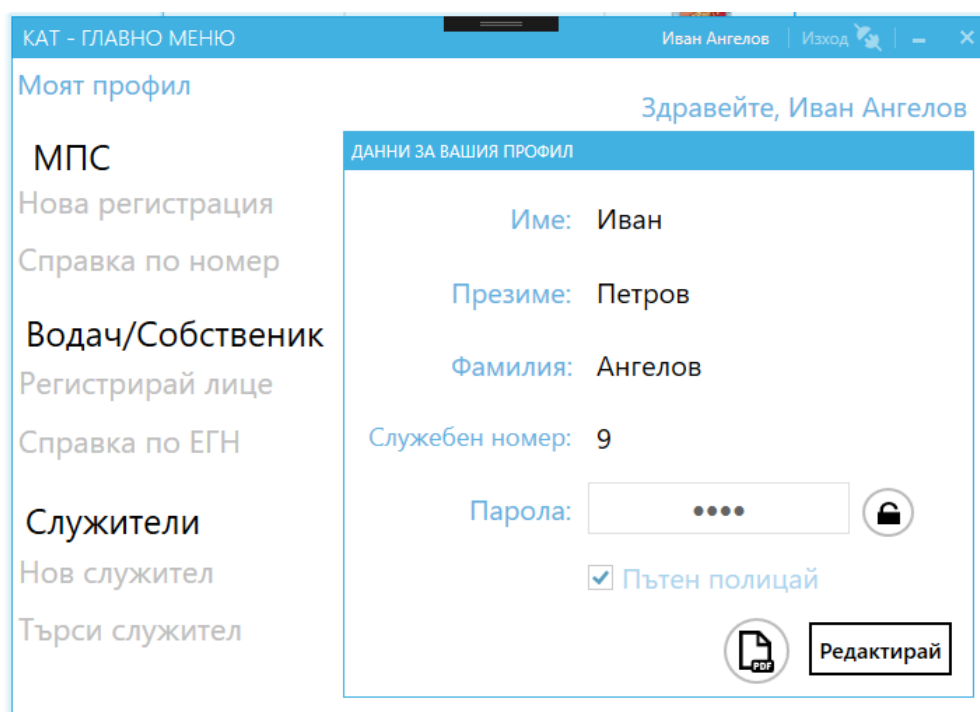
- *Концептуален дизайн - десктоп*

Системата разполага със съвременен и интуитивен интерфейс изграден на базата на WPF (за десктоп) и WindowsPhone Silverlight 8.1 (за Windows Phone).



Фиг. 6 – Вход в системата

На фиг. 6 е представена стартовата отправна точка на десктоп приложението, или иначе казано – **Login** прозореца. Сам по себе си е достатъчно описателен и не се нуждае от пояснения.



Фиг. 7 – Главно меню

След успешен вход в системата, потребителят бива препратен към основния екран/меню – изграден с помощта на **TabControl** (чийто дизайн е дело на **MahApps UI** фреймуъркът).

По подразбиране като основен таб е избран – *‘Моят профил’*. Менюто е изградено на базата на един основен навигационен прозорец, вляво на който е разположен навигационния интерфейс. Благодарение на него се осъществява прехода между отделните View-та, изградени като елементи от тип *User Control*. Те се разполагат в ContentContainer, намиращ се вдясно на основния прозорец.

Здравейте, Иван Ангелов


ДАННИ ЗА ВАШИЯ ПРОФИЛ

Име: Иван


Презиме: Петров

Фамилия: Ангелов

Служебен номер: 9

Парола: 

☒ Пътен полицай



Редактирай

Фиг. 8 – ContentContainer

Моят профил

МПС

Нова регистрация

Справка по номер

Водач/Собственик

Регистрирай лице

Справка по ЕГН

Служители

Нов служител

Търси служител

Фиг. 7 – Навигационен интерфейс

В горната част на основния прозорец са разположени данни, описващи заглавието на страницата, име и фамилия на вписания служител и възможност за отписването му.

‘Моят профил’ е визуализиран по подразбиране. Чрез този компонент се визуализират данните на вписания служител. Предлага се и възможност за редакция на паролата на служител. Паролата е скрита и визуализирането и става само след задържане на поставения за целта бутон.

КАТ - ГЛАВНО МЕНЮ Иван Ангелов Изход

Моят профил

МПС

Нова регистрация

Справка по номер

Водач/Собственик

Регистрирай лице

Справка по ЕГН

Служители

Нов служител

Търси служител

Нова автомобилна регистрация

ДАННИ ЗА РЕГИСТРАЦИЯ

Регистрационен номер:

ЕГН на собственик:

Производител:

Модел:

Тип:

Кубатура:

Конски сили:

Цвят:

VIN номер:

Първа регистрация:

Настояща регистрация:

ДАННИ ЗА ВИНЕТКА

☐ Винетка

Валидна до:

ДАННИ ЗА ЗАСТРАХОВКА ГРАЖДАНСКА ОТГОВОРНОСТ

☐ Гражданска отговорност

Застраховател:

Сключена на:

Валидна до:

ДАННИ ЗА ЗАСТРАХОВКА 'АВТОКАСКО'

☐ Автокаска

Застраховател:

Сключена на:

Валидна до:

Добави Изчисти

Фиг. 9

‘Нова регистрация’

Табът **‘Нова регистрация’** предоставя на потребителя функционалност за добавяне на нова автомобилна регистрация. Попълват се съответните бизнес данни, като за окончателния им запис в базата е необходимо да бъде натиснат бутона ‘Запамети’.

Възможно е и изчистване на полетата при грешка или при подготовка за нов запис. Това става чрез предвидения за целта бутон – ‘Изчисти’.

Друг основен таб е **‘Справка по номер’**, който също е част от категорията ‘МПС’.

КАТ - ГЛАВНО МЕНЮ Иван Ангелов Изход

Моят профил

МПС

Нова регистрация

Справка по номер

Водач/Собственик

Регистрирай лице

Справка по ЕГН

Служители

Нов служител

Търси служител

Справка за регистрация

ТЪРСИ РЕГИСТРАЦИЯ

Номер на превозното средство:

Търси

Всяка заявка ще бъде показвана в отделен прозорец, за да може обработката на заявките да се осъществява паралелно.

Фиг. 10 - ‘Справка по номер’

Компонентът е сравнително семпъл. Като входни данни се очаква да бъде въведен регистрационен номер на превозно средство. След успешна справка за наличие на регистрация, се визуализира дъщерен прозорец, а при несъответствие на данните се визуализира контролна информация за валидност (Грешки под формата на съобщения – MessageBox)

При успешна справка по номер и открита регистрация в база данни, се зарежда следния прозорец:

КН6666АН - СПРАВКА ЗА РЕГИСТРАЦИЯ

Справка за автомобилна регистрация

ДАННИ ЗА РЕГИСТРАЦИЯ

Регистрационен номер: КН6666АН

ЕГН на собственик: 9403122826

Производител: Opel

Модел: Astra

Тип: Лек автомобил

Кубатура: 1400

Конски сили: 62

Цвят: зелен

VIN номер: SJFSH46DSF45DS

Първа регистрация: 04-May-16

Настояща регистрация: 02-May-16

ДАННИ ЗА ВИНЕТКА

☒ Винетка

Валидна до: 22-Apr-16

ДАННИ ЗА ЗАСТРАХОВКА ГРАЖДАНСКА ОТГОВОРНОСТ

☒ Гражданска отговорност

Застраховател: Alianz

Сключена на: 15-Apr-16

Валидна до: 03-Mar-17

ДАННИ ЗА ЗАСТРАХОВКА 'АВТОКАСКО'

☐ Автокасco

Застраховател: Застрахователно дружество

Сключена на: Введете дата

Валидна до: Введете дата

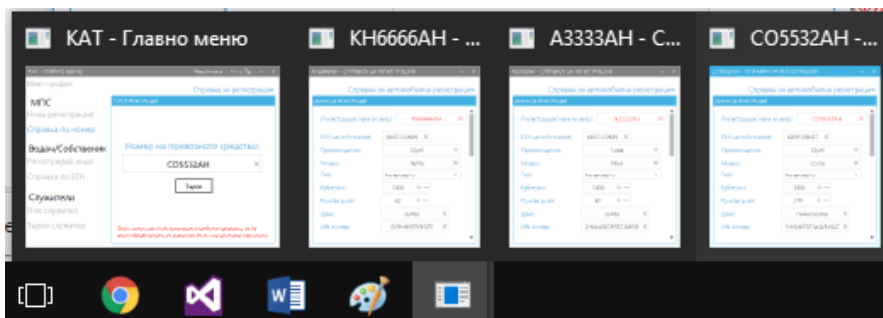
Фиг. 11 – 'Справка по номер' - детайли

Той представлява placeholder, в който се разполага детайлна информация, свързана с данните, описващи автомобилна 'Регистрация'.

Дизайнът е опростен, а навигацията се осъществява с интуитивно скролиране, вградено в 'Данни за регистрация' контролата. (GroupBox).

Дъщерни контроли (вградени в основната – 'Данни за регистрация' контрола) са тези, свързани с документно-застрахователни дейности (Винетка, застраховка 'Гражданска отговорност', 'Автокасco').

Заглавието на прозореца започва със регистрационния номер, с цел да се улесни паралелното осъществяване на справки.



Фиг. 12 – Паралелна обработка на множество справки без необходимост от фокусирането им на екрана

Друга основна функционалност на системата е регистрацията на водач, въплътена като подкатегория ‘Регистрирай лице’ (част от категорията ‘Водач/Собственик’).

КАТ - ГЛАВНО МЕНЮ Иван Ангелов Изход

Моят профил

Добавяне на водач

МПС

Нова регистрация

Справка по номер

Водач/Собственик

Регистрирай лице

Справка по ЕГН

Служители

Нов служител

Търси служител

ДАННИ НА ВОДАЧ/СОБСТВЕНИК

ЕГН на водач: 0

Име: Въведете име Мъж Жена

Презиме: Въведете презиме Гражданство: България

Фамилия: Въведете фамилия

Дата на раждане: 01-Jan-00

Месторождение: Въведете месторождение

Постоянен адрес: Въведете постоянен адрес

Въведете телефонен номер

Въведете мейл

Контролни точки: 39

Документът е издаден от: Въведете МВР клон

Издаден на: 13-May-16

Валиден до: 13-May-26

КАТЕГОРИИ

Категория	Придобита на	Валидна до	Ограничения
A1	Въведете дата	Въведете дата	
A	Въведете дата	Въведете дата	
B1	Въведете дата	Въведете дата	
B	Въведете дата	Въведете дата	
C1	Въведете дата	Въведете дата	

Страницата съдържа всички полета, необходими (задължителни и незадължителни) за добавяне на нов запис на лице в системата.

Веднага след персоналната информация, следва въвеждането на данни относно шофьорския статус на лицето.

Важна подкатегория е ‘КАТЕГОРИИ’. В нея се съдържа информация за настоящата правоспособност за управление на различни видове МПС.

Страницата е изградена съобразно със съществуващите норми (дизайн) при физическите документи (контролен талон на водача и шофьорска книжка).

Фиг. 13 – Страница ‘Регистрация на водач’

Освен вече представената – справка за регистрацията по номер, системата разполага и с механизъм за осъществяване на справки за водач по ЕГН на водача. Тази функционалност е достъпна от менюто – ‘Справка по ЕГН’ (част от ‘Водач /Собственик’).

Фиг. 14 – Справка за водач/собственик по ЕГН

При опит за търсене има два основни изхода – водачът е наличен в системата – следва зареждане на данните базата и тяхното последващо визуализиране в нов прозорец и другият изход – не съществуват данни за водач със съответното ЕГН. Частни случаи са съобщенията за грешки (валидации на ниво вход/изход и валидации свързани с отговор от базата).

Фиг. 15 – Резултатен прозорец при успешна справка (лице с въведеното ЕГН е налично в базата)

При съвпадение на въведеното ЕГН и запис от базата се зарежда прозорецът с информация за водача/собственика.

Справката е разделена в три отделни секции с цел по – лесен и бърз достъп до необходимата информация.

При зареждане, по подразбиране се визуализира съдържанието на секцията ‘Лични данни’. Заглавието е достатъчно описателно само по себе си. Тук се представят в подходящ вид персоналните данни на лицето.

ГЕОРГИ ЧЕЛЕНКОВ - СПРАВКА ЗА ВОДАЧ

Лични данни Шофьорска книжка Нарушения

Шофьорска книжка на Георги Челенков

ДАННИ ЗА ШОФЬОРСКА КНИЖКА

Документът е издаден от: MBP Кюстендил

Издаден на: 02-Jul-12 Валиден до: 02-Jul-22

Контролни точки: 39

КАТЕГОРИИ

Категория	Придобита на	Валидна до	Ограничения
A1	06-Apr-16	07-Apr-16	agafas
A	Select a date	Select a date	
B1	Select a date	Select a date	

Фиг. 16 – Справка за водач – категория ‘Шофьорска книжка’

В следващия таб – ‘Шофьорска книжка’ е позиционирана информация за шофьорския статус на лицето.

Различаващ се от останалите, чисто презентационни категории, е ‘Нарушения’.

Категорията представя цялостен списък от нарушение на даденото лице, като ги подрежда в таблица.

Съществува възможност за премахване на записа на нарушение чрез селектирането му и натискане на предназначения за целта бутон (изобразяващ кофа за боклук).

При необходимост от напомняне или свързване с лицето относно детайли по дадено нарушение, съществува възможност за директно изпращане на съобщение на електонната поща. За достъп до този feature е необходимо да се избере бутонът в горния десен ъгъл (пощенски плик). След избирането му се отваря прозорец характерен за всяка една имейл поща.

ГЕОРГИ ЧЕЛЕНКОВ - СПРАВКА ЗА ВОДАЧ

Лични данни Шофьорска книжка Нарушения

Нарушения на Георги Челенков

ДАННИ ЗА ИЗВЪРШЕНИ НАРУШЕНИЯ

№	ИЗВЪРШЕНО НА	НАРУШЕНИЕ
14	4/14/2016 9:23:39 AM	Преминаване на Червен светофар

За да видите повече детайли, кликнете двукратно с мишката върху съответното нарушение

Фиг. 17 – Справка за водач – категория ‘Нарушения’

ГЕОРГИ ЧЕЛЕНКОВ - КОНТАКТ

Съобщение до Георги Челенков

Подател: <resiloveca94@gmail.com> Иван Ангелов

Получател: <resiloveca94@gmail.com> Георги Челенков

Тема: Въведете тема (до 80 символа)

Съобщение: Въведете съобщение (до 500 символа)

Изпрати

Фиг. 18 – Връзка с нарушителя чрез електронно съобщение

За повече подробности около дадено нарушение е възможно отварянето на детайлна страница чрез двукратно кликане върху избраното нарушение. Вследствие на това, се отваря компактна страница, пригодена да представи основните детайли на нарушението.

НАРУШЕНИЕ № 14 - ГЕОРГИ ЧЕЛЕНКОВ

Нарушение на Георги Челенков

ДАННИ ЗА НАРУШЕНИЕ

Контролен номер: 14

Дата на извършване: 14-Apr-16

Дата на регистрация: 14-Apr-16

Съставител служебен №: 9

Местопроизшествие: София, бул. Андрей Сахаров

Описание: Преминаване на Червен светофар

Възражение: Нарушението е извършено при намалена видимост и с неразрешена скорост


Map showing location: Slatina, The Mall, Boulevard Professor, Boulevard Sveti Kiment O, Malinov, Ulitsa Nikola Gabrovski, Ulitsa 8-mi dekemvri, Boulevard Andrei Lyapchev.

Фиг. 19 – Детайли за нарушение

На тази страница ясно се открояват данните, съпътстващи нарушението, както и карта, визуализираща точното местопроизшествие. За реализирането на функционалността с картата, е използван интерфейсът, предоставен от **Bing Maps**, зареден с генерирания **CredentialToken** (ключ, позволяващ пълноценното използване на Bing Maps компонента).

Освен представената информация, тази страница е изходна точка за две основни функции – генериране на документ – възражение на водач (чрез кръглия бутон с pdf страница) и директната справка за служителя – съставител на документа (кръглия бутон с бюст до ‘Съставител служебен №’).

Шаблонът на документа, който се генерира, представлява официална форма и използвана в тези случаи от институцията. В зависимост от конкретните данни, той се попълва автоматично и бива готов за принтиране и съхранение под формата на широкоразпространения - **.PDF** формат.



До началника на ОПП-СДВР

ВЪЗРАЖЕНИЕ

от **<Име>**
ЕГН **<ЕГН>** , живущ на
адрес: **<Адрес>**

по акт за установяване на
административно нарушение бланков
№ **<№ акт>**
от дата **<Акт дата>**

Уважаеми господин Началник,

Представям на вниманието Ви в срок възраженията си по съставения ми акт за установяване на административно нарушение бланков № **<№ акт>** от дата **<Акт дата>**

Не съм извършил така вмененото ми нарушение и правя следните обяснения:

<Възражение>

Не считам, че констатацията по акта е коректна и във връзка с това моля да приложите чл.54 ЗАНН и да не ми налагате наказание за нещо, което не съм извършил. Моля да ми бъдат предоставени писмени мотиви по вапето становище във връзка с моите възражения.

Към възражението си прилагам препис от акта.

ПРИЛОЖЕНИЕ:

Копие от акт № **<акт №>** от **<акт дата>**

Дата: **<дата на справка>**

С уважение: **<имена>**

Фиг. 20 – Шаблон на документа за възражение

Последната категория в системата е ‘*Служители*’. В нея се съдържат две подкатегории – ‘*Нов служител*’ и ‘*Търси служител*’. Имената им показват еднозначно каква функционалност изпълняват.

Кат - ГЛАВНО МЕНЮ

Иван Ангелов | Изход

Моят профил

Добавяне на нов служител

ДАННИ НА СЛУЖИТЕЛЯ

Име: Въведете име X

Презиме: Въведете презиме X

Фамилия: Въведете фамилия X

Парола: Въведете парола X

☐ Пътен полицай

На всеки новорегистриран служител ще бъде генериран автоматично служебен номер

Добави Изчисти

Фиг. 21 – Форма за добавяне на нов служител

Бутонът ‘Добави’ служи за добавяне на данните за новия служител в основната база.

Подкатегорията ‘Търси служител’ предоставя възможност на потребителя, въвеждайки служебен номер на служител, да прави справка за даден служител.

Кат - ГЛАВНО МЕНЮ

Иван Ангелов | Изход

Моят профил

Справка за служител

ТЪРСИ СЛУЖИТЕЛ

Служебен номер на служителя:

Въведете Служебен номер X

Търси

Всяка заявки ще бъде показвана в отделен прозорец, за да може обработката на заявките да се осъществява паралелно.

Фиг. 22 – Категория ‘Търси служител’

Аналогично на предходните форми (‘*Нова регистрация*’ и ‘*Регистрирай лице*’) ‘*Нов служител*’ представлява форма, която помага на администраторите в полицейското управление при наемане на нови служители на длъжност. Формата е семпла и самоописателна.

За изчистване на въведените данни, е предвиден бутон ‘Изчисти’.

ИВАН АНГЕЛОВ - СПРАВКА ЗА СЛУЖИТЕЛ

Справка за служител

Име: Иван

Презиме: Петров

Фамилия: Ангелов

Служебен номер: 9

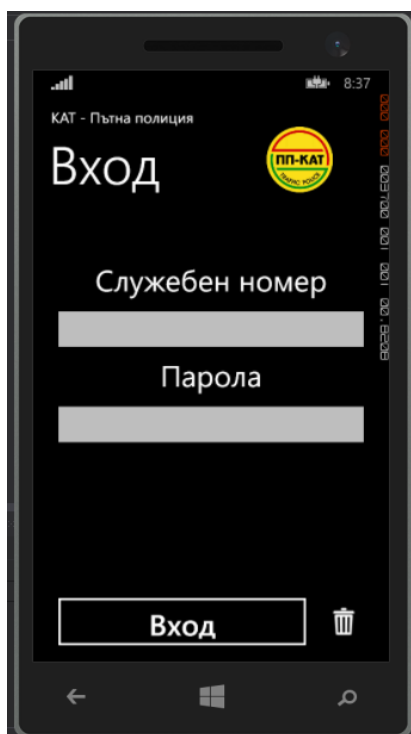
☒ Пътен полицай

Фиг. 23 – Резултат при успешна справка

▪ Концептуален дизайн – мобилна версия

На този етап, кроссплатформената архитектура (изградена с MVVM Cross) е въплътена в интерфейс на Windows Phone 8.1. Предвижда се изграждането и на iOS и Android интерфейс, като процесът и преходът би бил доста плавен. Необходимо е единствено да се нареди native интерфейсът на съответната система и да се осъществи връзката към вече изградените мултиплатформени ViewModels.

Отправната точка на мобилното приложение е неговата т. нар. ‘Login’ форма. Чрез нея е предвидено полевите полицаи да се впишат в началото на всеки работен ден, тъй като приложението играе роля на тяхна ‘мобилна’ работна станция. Чрез него, те имат достъп до минималната задължителна функционалност за работа при полеви условия. Идеята и функционирането на формата е аналогични на login формата при десктоп версията на системата, с тази разлика, че дизайнът е изцяло съобразен с официалните насоки за дизайн на мобилно приложение (Windows Phone Application Design Guidelines).



Фиг. 24 – ‘Login’ форма

Формата разполага с необходимите контроли, чрез които потребителят да въведе своите данни и да получи достъп до системата.

При некоректно въведени данни, съществува възможност (чрез бутона с форма на кош за отпадъци) бързо и лесно да се нулират полетата с цел – ново въвеждане.

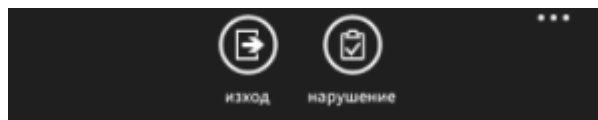
При успешен вход в системата се зарежда страницата – ‘Главно меню’, която представлява контейнер (меню базирано на типичната – ‘*Panorama*’ контрола), в който са разположени две от най – често използваните функционалности при работа на полевите служители – Справка за регистрацията и справка за водач.



Фиг. 25 – Справка за водач и справка за регистрация

Над заглавието на категорията се представя поздравително съобщение, насочено към служителя, успешно влязал в системата.

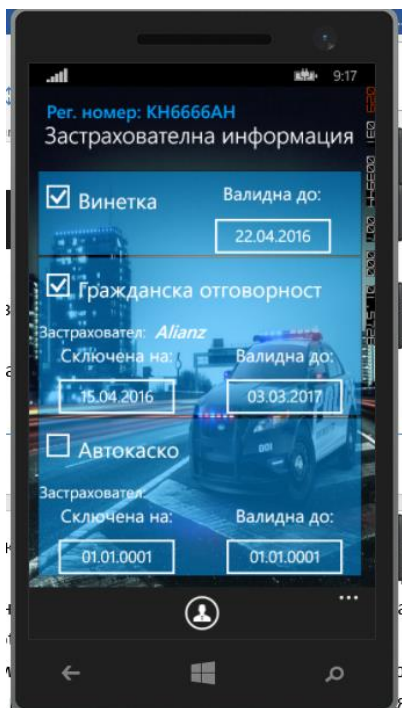
Имплементирана е функционалност за изход от системата (т. нар. Logout), която може да бъде достъпена чрез бутон, намиращ се в областта на типичното за системата AppBar Menu.



Фиг. 26 – AppBar бутони

От главното меню, чрез бутона ‘*нарушение*’ може директно да бъде достъпена страницата за съставяне на наказателно постановление. Дизайнът на тази страница ще бъде представен по-нататък в настоящото изложение.

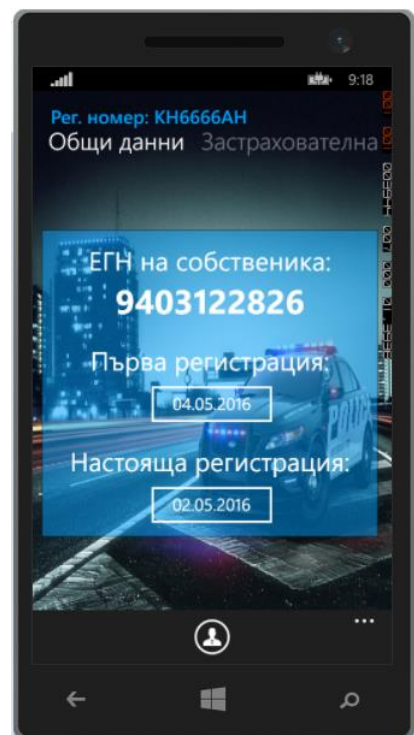
При успешна справка по регистрацията се зарежда ново меню (чрез контролата ‘Pivot’) което предоставя необходимата информация на служителя по един лесен за възприемане и достъпен начин – посредством предоставяне на информацията на отделни ‘порции’. Pivot менюто съдържа 3 подкатегории – ‘Общи данни’, ‘Застрахователна информация’ и ‘Технически характеристики’, чрез които във всеки един момент може да се направи нужната справка. Менюто е изцяло с информативен характер.



Фиг. 27 – Подменю
‘Застрахователна
информация’



Фиг. 28 – Подменю
‘Технически
характеристики’



Фиг. 29 – Подменю ‘Общи
данни’

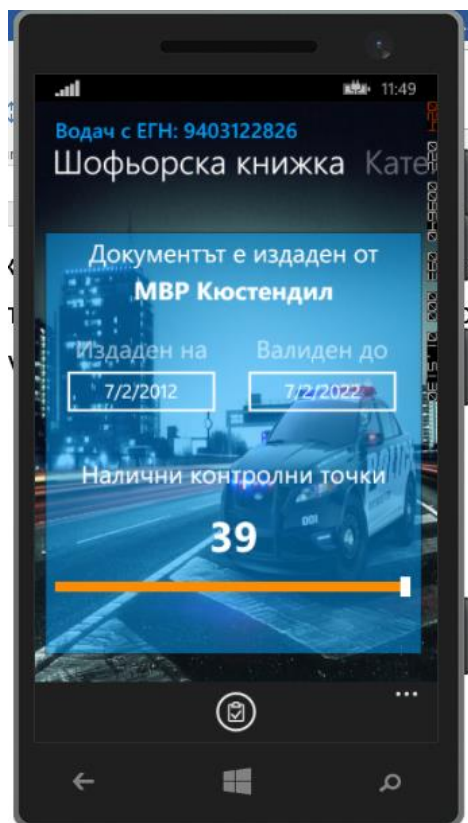
И трите подменюта покриват стандартите за съвременен и интуитивен, унифициран дизайн, като през цялото време над заглавието на подменютата се визуализира информация за регистрационния номер на текущата справка.

Предвидена е и възможността за директна справка за водача-собственик на регистрацията, чрез ApplicationBar бутона.

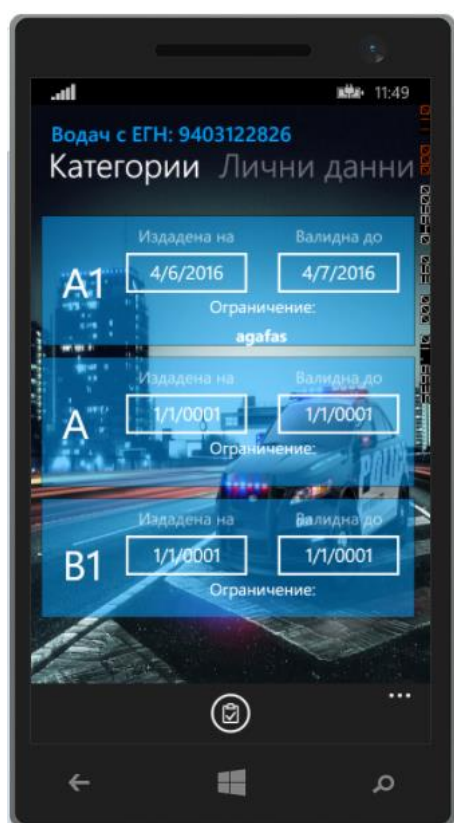


Фиг. 30 – AppBar бутон за директна справка за водача-собственик на регистрацията

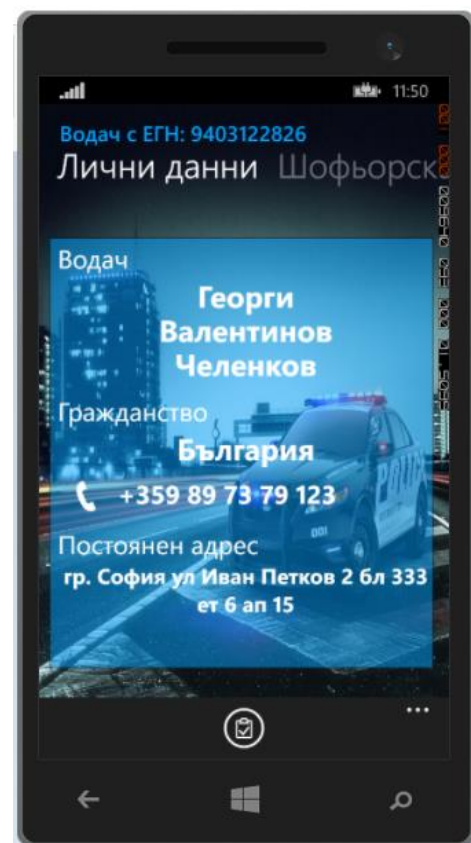
Като резултат от справка за водач по ЕГН, потребителят получава достъп до информация, форматирана по аналогия на достъпа до регистрация – 3 основни подменюта, разположени в Pivot контрола.



Фиг. 31 – Подменю
'Шофьорска книжка'



Фиг. 32 – Подменю
'Категории'



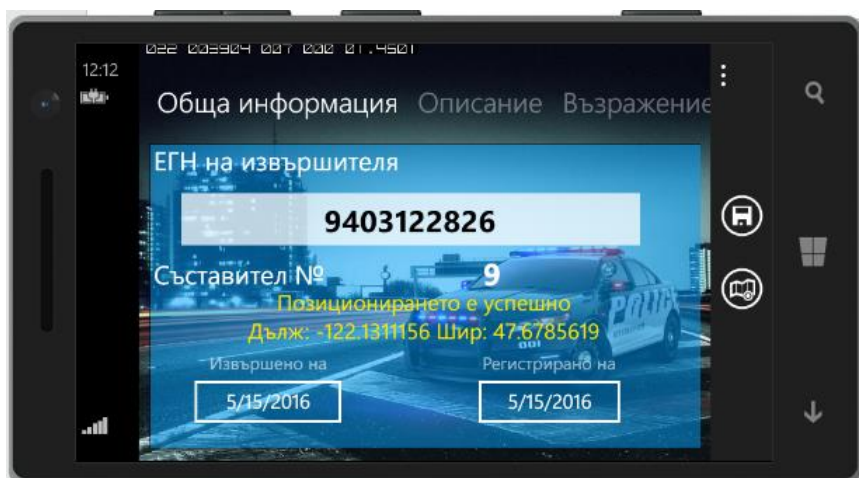
Фиг. 33 – Подменю 'Лични
данни'

В подменюто 'Шофьорска книжка' служителят може да види основната интересуваша го информация при проверка на шофьор. Поради този факт, това подменю е селектирано по подразбиране.

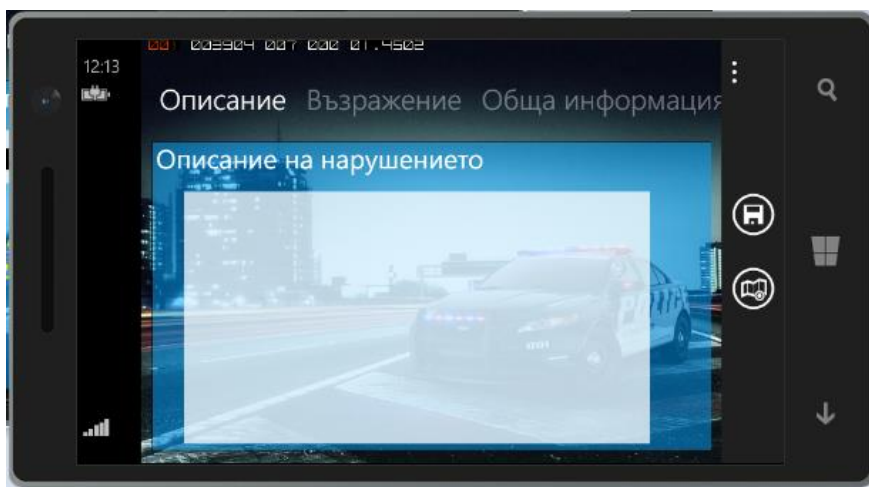
'Категории' е скролиращ се панел, в който, под формата на обособени 'плочки' се изобразява информация, копие на съществуващия формат на шофьорска книжка.

AppBar бутонът е със същата функционалност, като бутона за съставяне на нарушение в основния навигационен прозорец, с тази разлика, че на служителите няма да му се налага да вписва ЕГН на водача, а то автоматично ще бъде заредено във формата за регистрация на нарушение. И трите подменюта имат презентационен характер.

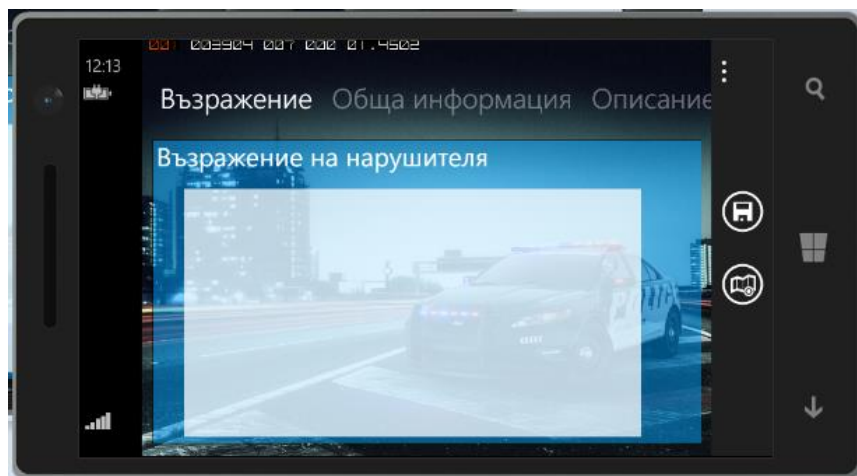
Друга основна способност на мобилното приложение е възможността за съставяне на нарушения и записът им в базата данни.



Фиг. 34 – Подменю 'Обща информация'



Фиг. 35 – Подменю 'Описание'



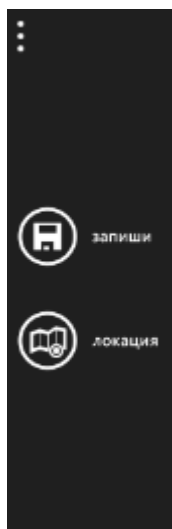
Фиг. 36 – Подменю 'Възражение'

‘Обща информация’ представя главните полета, необходими за съставяне на валидно нарушение. При избиране на опцията за съставяне на нарушение от главното меню, потребителят ще трябва ръчно да въведе желаното ЕГН на нарушителя. Ако отправна точка на менюто за нарушение е справката за водач, то ЕГН ще бъде автоматично попълнено.

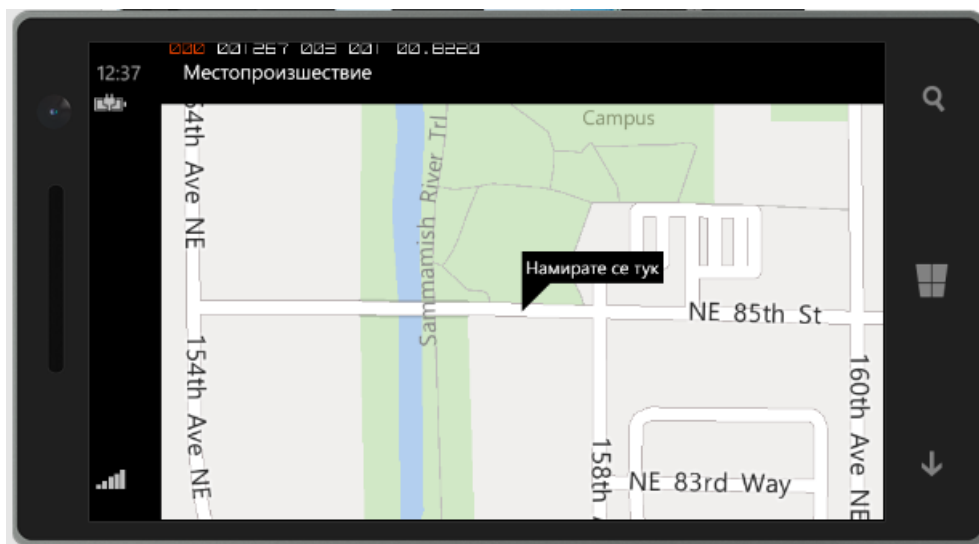
Важно допълнение е осъществяващото се в този момент позициониране, благодарение на GPS системата на устройството. Чрез системата за позициониране, автоматично ще бъде събрана информация за местоположението на устройството, респективно за местопроизшествието (географска ширина и дължина), която ще бъде използвана в последствие, както за запис в базата, така и за визуализиране на позицията на картата.

Детайли за нарушението и възражение от страна на нарушителя се въвеждат в съответните подменюта.

В AppBar менюто се открояват два бутона, благодарение, на които се реализира окончателния запис на приложението в базата, както и визуализация на настоящото местоположение на картата.



Фиг. 37 – AppBar бутони
– ‘Запиши’ и ‘Локация’



Фиг. 38 – Визуализация на
местопроизшествието

Реализация

- База данни

Базата данни, с която работи приложението, е проектирана да покрива цялостните му нужди. Тя представлява релационна база данни, изградена с помощта на **MySQL**. И със средствата на графичната среда MySQL Workbench.

Разположена е на хостинг (различен от localhost), като данните за достъп до нея са следните:

Host: sql7.freemysqlhosting.net
Database name: sql7112557
Database user: sql7112557
Database password: Fwg8uEkpT1
Port number: 3306

За управлението на базата се предоставя и графична среда, базирана на **PHPMyAdmin**. (достъп до информацията се получава след вход в системата от <http://www.phpmyadmin.co>)

Предвидени са следните таблици:

driver_owners
driver_owner_id BIGINT(10)
first_name VARCHAR(20)
second_name VARCHAR(20)
last_name VARCHAR(20)
sex ENUM('M', 'Ж')
nationality VARCHAR(20)
birth_date DATE
birth_place VARCHAR(30)
residence VARCHAR(50)
tel_num VARCHAR(20)
email VARCHAR(30)
remaining_pts TINYINT(2)
licence_issue_date DATE
licence_expiry_date DATE
licence_issued_by VARCHAR(30)
a1_acquiry_date DATE
a1_expiry_date DATE
a1_restrictions VARCHAR(30)
a_acquiry_date DATE
a_expiry_date DATE
a_restrictions VARCHAR(30)
b1_acquiry_date DATE
b1_expiry_date DATE
b1_restrictions VARCHAR(30)
b_acquiry_date DATE
b_expiry_date DATE
VARCHAR(30)
c1_acquiry_date DATE
c1_expiry_date DATE
c1_restrictions VARCHAR(30)
36 more...
Indexes

Фиг. 39 – Данново представяне на водач/собственик

registrations
reg_num VARCHAR(13)
driver_owner_id BIGINT(10)
car_manufacturer VARCHAR(15)
car_model VARCHAR(15)
car_color VARCHAR(15)
car_type ENUM
car_cubage INT(6)
car_hp INT(5)
car_vin VARCHAR(17)
first_reg_date DATE
civil_insurance BIT(1)
civil_insurer VARCHAR(20)
civil_insurance_start_date DATE
civil_insurance_end_date DATE
has_vignette BIT(1)
vignette_valid_until DATE
damage_insurance BIT(1)
damage_insurer VARCHAR(20)
damage_insurance_start_date DATE
damage_insurance_end_date DATE
recent_certificate_reg_date DATE
Indexes

Фиг. 40 – Данново представяне на регистрация

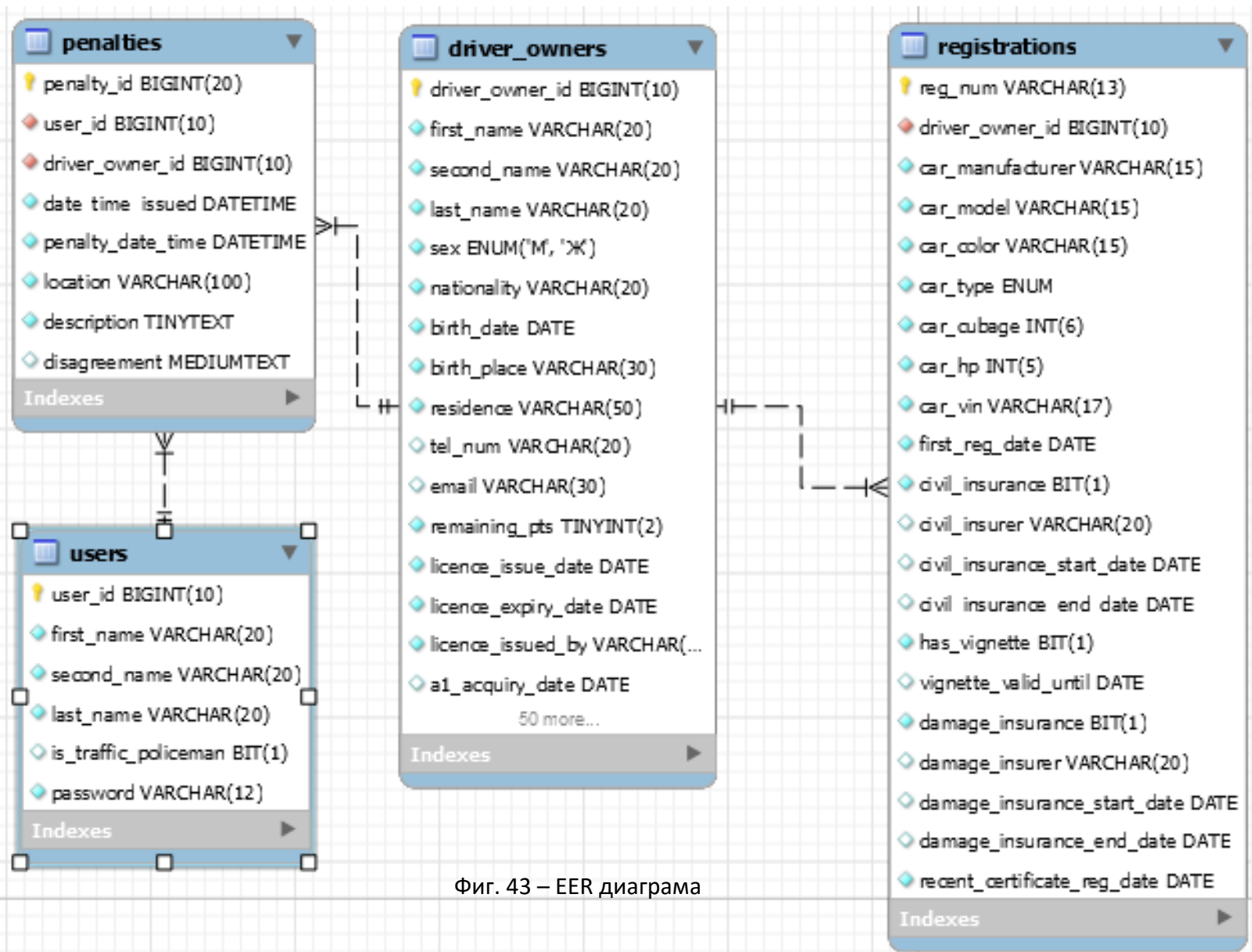
users
user_id BIGINT(10)
first_name VARCHAR(20)
second_name VARCHAR(20)
last_name VARCHAR(20)
is_traffic_policeman BIT(1)
password VARCHAR(12)
Indexes

Фиг. 41 – Данново представяне на потребител

penalties
penalty_id BIGINT(20)
user_id BIGINT(10)
driver_owner_id BIGINT(10)
date_time_issued DATETIME
penalty_date_time DATETIME
location VARCHAR(100)
description TINYTEXT
disagreement MEDIUMTEXT
Indexes

Фиг. 42 – Данново представяне на нарушение

Релацията между отделните обекти (Entities) е представена на следната подробна EER диаграма (EER – Enhanced Entity-Relationship):



Фиг. 43 – EER диаграма

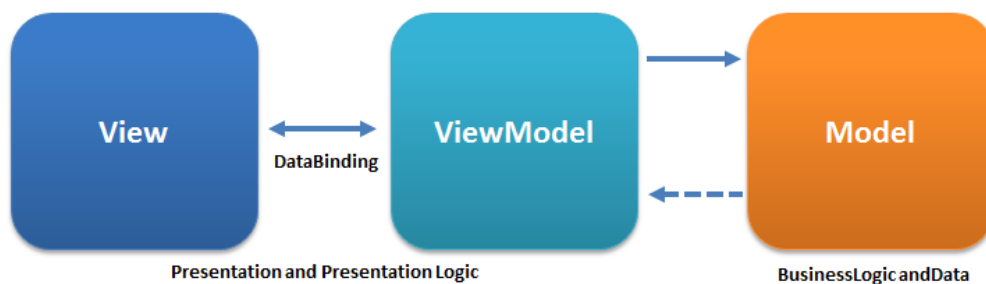
С цел – повишаване на гъвкавостта и сигурността на базата, заявките, които се извършват са пакетирани в процедури (Stored Procedures). Съвкупността от процедури формира функционалния интерфейс на системата.

Routines	
Name	
add_driverowner	
add_penalty_to_driverowner	
add_registration	
add_user_and_get_id	
get_available_car_types	
get_driverowner_data_by_id	
get_driverowner_penalties_info	
get_reg_by_regnum	
get_user_by_id_and_pass	
remove_penalty_by_id	
update_user	

Фиг. 44 – Функционален интерфейс на системата

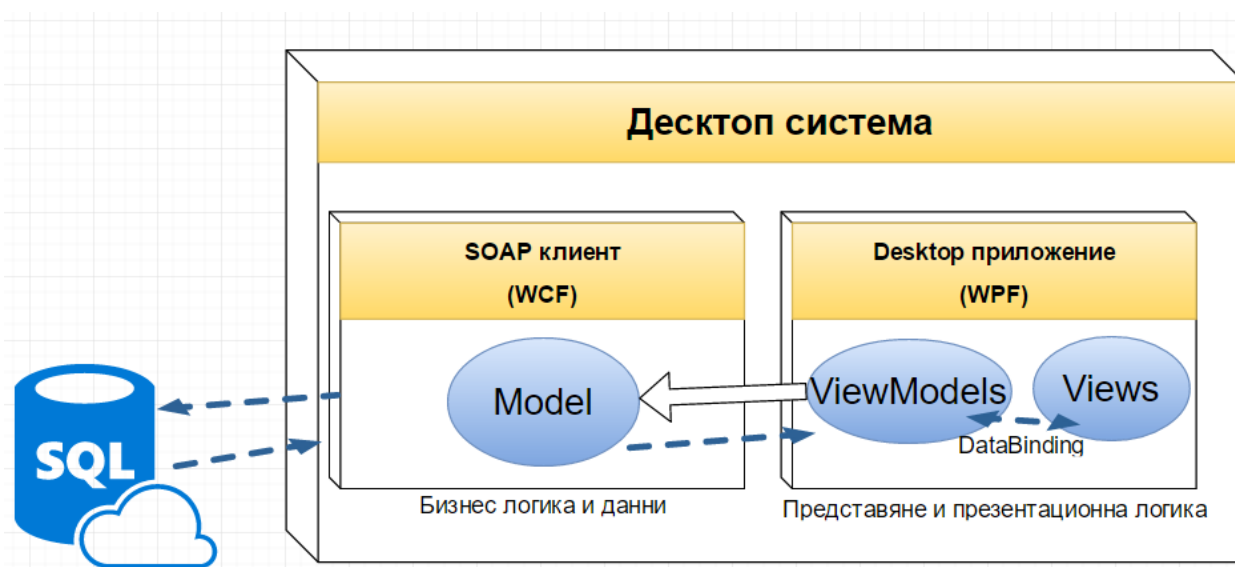
- Desktop архитектура

Десктоп системата е изградена съобразно MVVM (Model-View-ViewModel) архитектурата.



Фиг. 45 – MVVM архитектура

Разделена е на 2 основни приложения – *WCF SOAP Service* и *TrafficPoliceDesktop*



Фиг. 46 – MVVM архитектурен модел на приложението

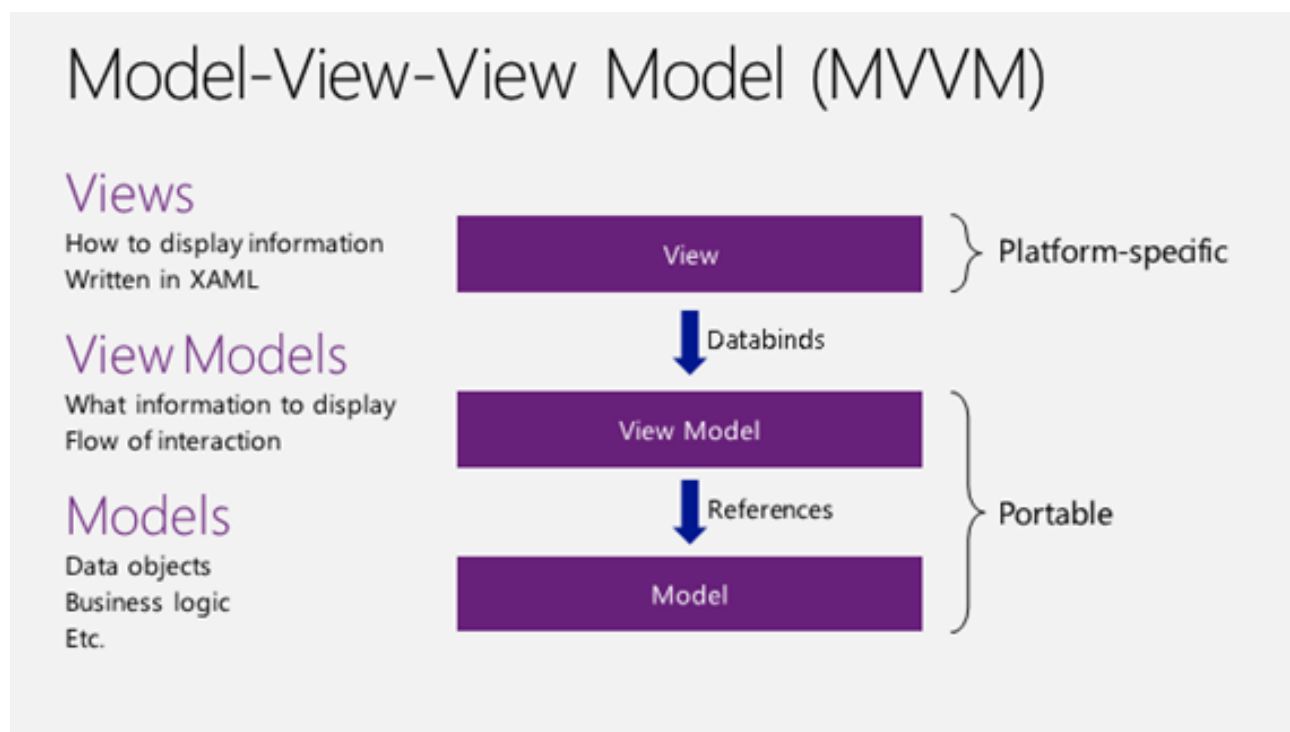
- WCF SOAP Service
 - Приложението е базирано на WCF (Windows Communication Foundation). То предоставя обектния модел, който ще бъде използван от TrafficPoliceDesktop. SOAP сървизът осъществява връзката с базата и предоставя на основното приложение интерфейс с възможната за достъпване функционалност.

- **TrafficPoliceDesktop**
 - Приложението е базирано на WPF (Windows Presentation Foundation). То предоставя достъп до ViewModel и View слоевете на десктоп системата. View слойът е изграден, спазвайки принципите и препоръките на Microsoft – минимален или нулев Code-behind, с идеята да се елиминира възможността за спагети код, който в последствие би бил труден за обработка при извършването на автоматизирано Unit тестване.

- **Мобилна архитектура**

Архитектурата на мобилното приложение е изградена на базата на MVVM Cross Framework. Мобилната система е изградена от 3 взаимно свързани подпроекта – *TrafficPolice.Core*, *WCFMobileSOAP*, *TrafficPolice.WP*. В *TrafficPolice.Core* проектът са съсредоточени **ViewModels**, *WCFMobileSOAP* предоставя функционалността за връзка с базата, функционалния интерфейс и **Model**, а *TrafficPolice.WP* проектът описва **Views** за Windows Phone Silverlight 8.1. Предвидено е и изграждането на Views за iOS – *TrafficPolice.IOS*, и версия за андроид – *TrafficPolice.Android*.

Разделението на проектите, описващи ViewModels и Views е ограничение, наложено от MVVM Cross мултиплатформения фреймуърк. *WCFMobileSOAP* и *TrafficPolice.Core* приложенията (т.е. ViewModels и Views) са общи и за трите платформи – WP, iOS и Android. Функционалността на ViewModels е изградена посредством множеството плъгини, разработени за нуждите на MVVM Cross кросплатформената разработка. Единствено проектът, описващ Views е индивидуален и native за всяка система.



Фиг. 47 – MVVM кросплатформена архитектура

Мултиплатформеното решение е изградено с помощта на Xamarin, в комбинация с MVVM Cross.

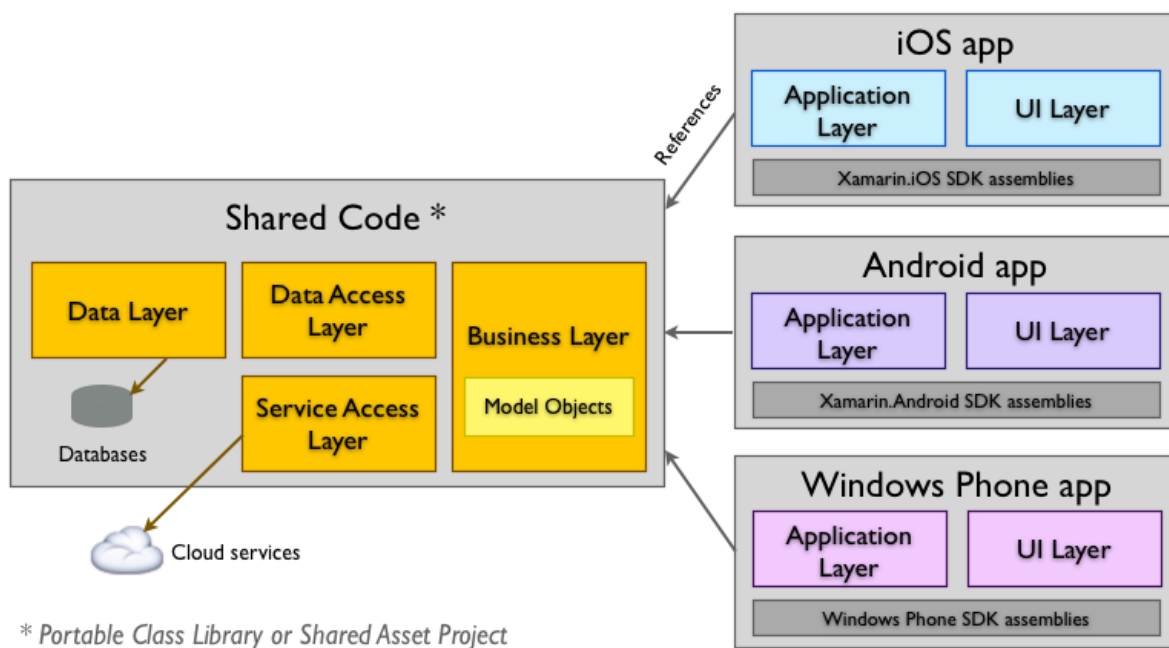
Xamarin + MvvmCross = cross-platform coolness

Фиг. 48 – Xamarin + MvvmCross = Кроссплатформено решение

Xamarin - <https://www.xamarin.com/>

MVVMCross Framework - <https://github.com/MvvmCross>

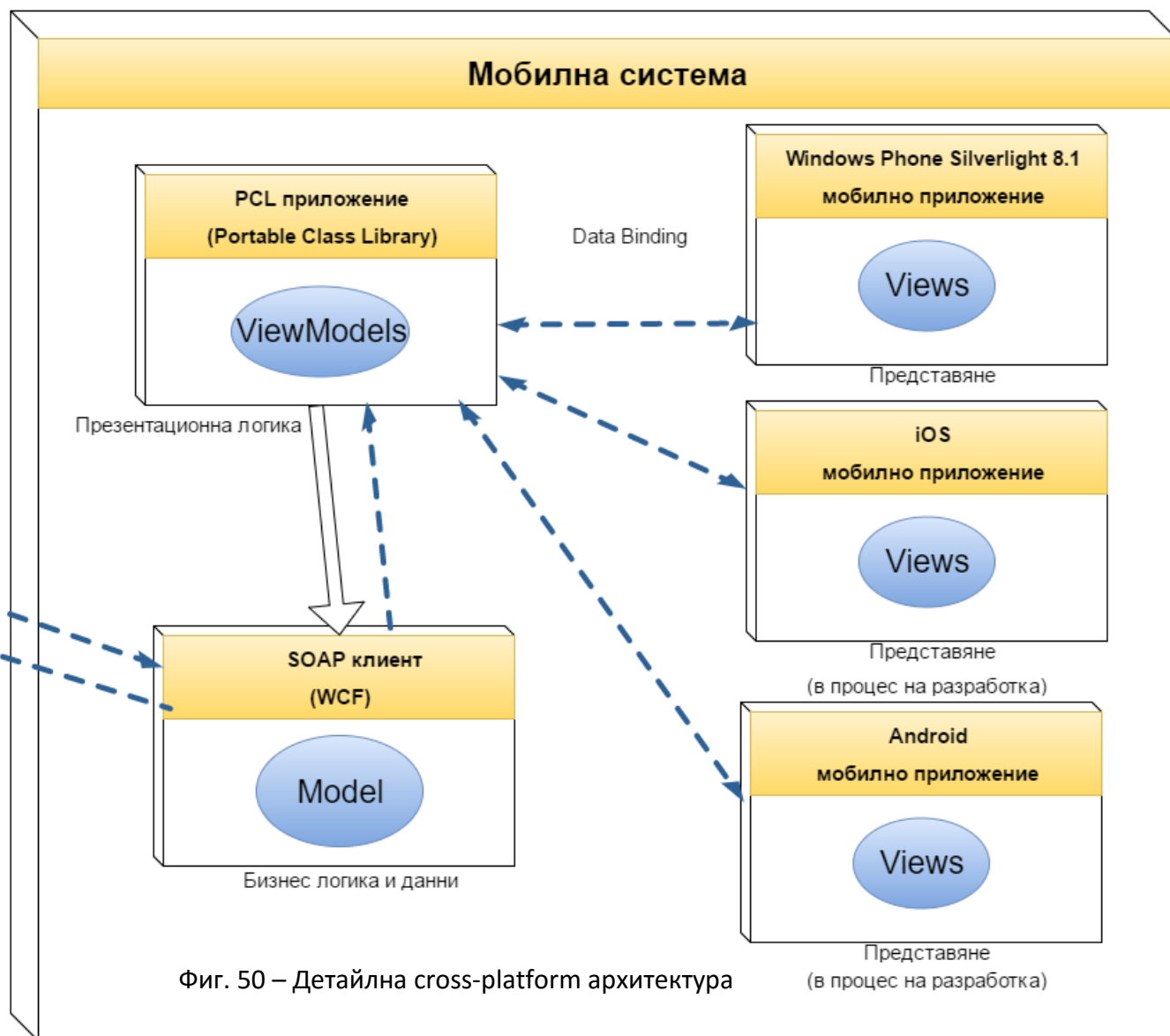
TrafficPolice.Core приложението представлява приложение от тип **Xamarin Portable Class Library** (Windows Phone, Android, iOS).



Фиг. 49 – Детайлна cross-platform архитектура

Специфичният тип на приложението налага ограничения откъм използване на типични похвати при отделните системи. Огромна част от библиотеките, които приложенията, работещи под различни операционни системи използват, са проектирани конкретно за системата. В тази ситуация на помощ на разработчика идват множеството плъгини за MVVM Cross, които са 'опаковани' ('wrapper') да използват типичната функционалност на системата.

Мултиплатформеното решение е изградено с помощта на Xamarin, в комбинация с MVVM Cross.



Системата е снабдена с функционалност, осъществяваща валидация, както на ниво интерфейс, така и валидация на отговора от извиканите процедури. Реализиран е клас – *InputValidator*, който описва правилата за валидация на отделните компоненти.

Потребителският интерфейс, както на десктоп приложението, така и на мобилното приложение, е изработен с помощта на стандартните контроли, предоставени от средата за разработка.

Използвана контрола, която не е част от стандартния пакет е т. нар. *Bing Map*. При него е необходимо да се генерира специален код, чрез който програмистът може да достъпи цялостната функционалност на системата (т. нар. *CredentialToken*).

Функционален на интерфейс на **десктоп** версията:

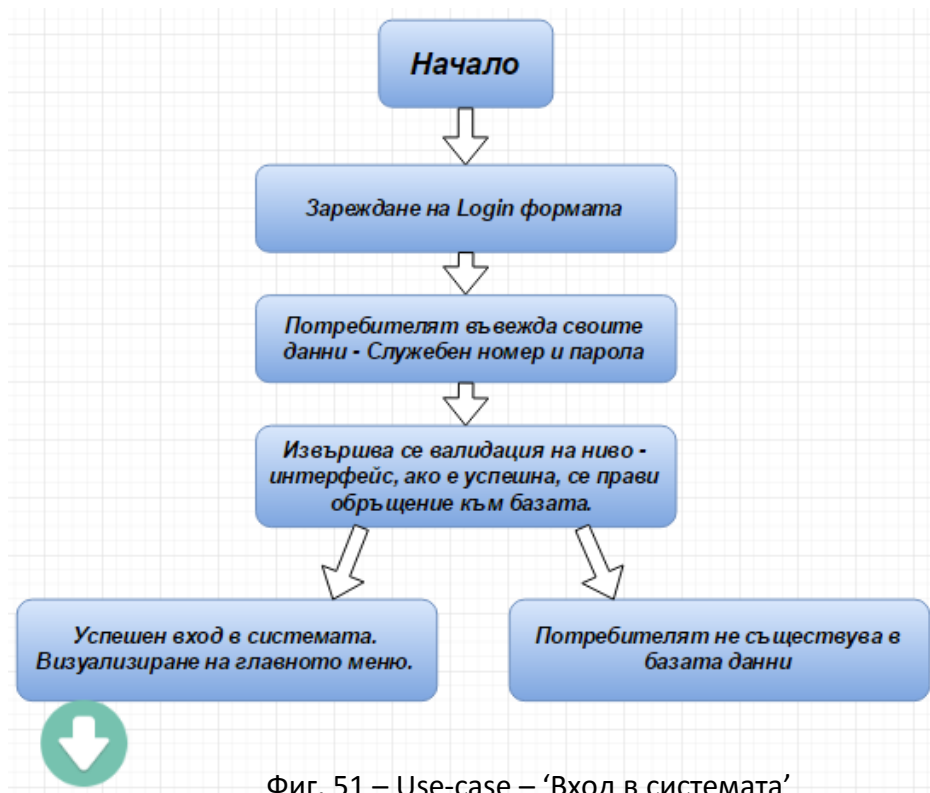
```
int UpdateUser(User user);
User GetByIdAndPass(string id, string password);
User GetReadOnlyUserById(string id);
string InsertUserAndGetGeneratedId(User usr);
int InsertCat(Categories categ);
int InsertRegistration(Registration reg);
int RegisterDriverOwner(DriverOwner drOwner);
DriverOwner GetDriverOwnerById(string id);
int removePenalty(Penalty pen);
List<string> getAvailableCarTypes();
Registration getRegByRegNum(string regNum);
```

Функционален на интерфейс на **мобилната** версията:

```
User GetByIdAndPass(string id, string password);
DriverOwner GetDriverOwnerById(string id);
Registration getRegByRegNum(string regNum);
string addPenaltyToDriverOwner(Penalty pen);
```

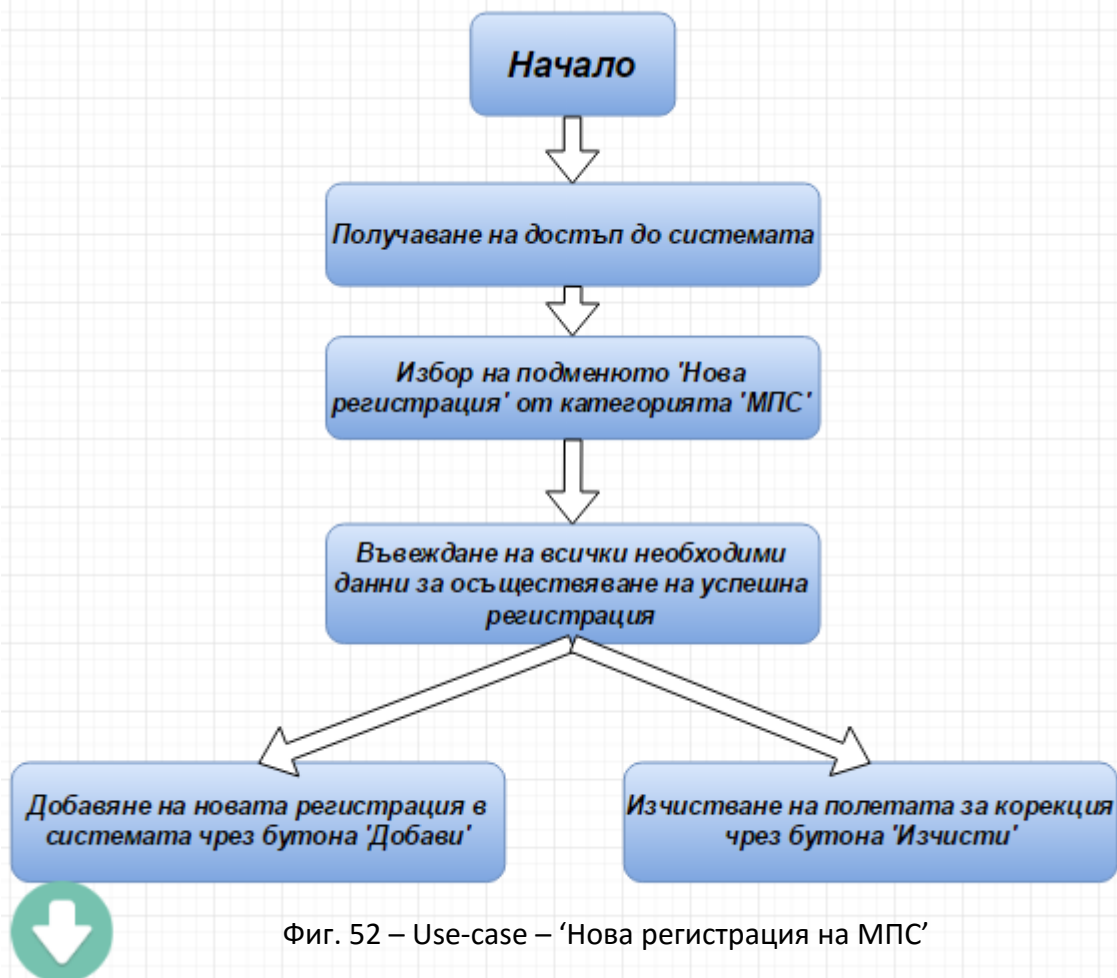
Потребителско ръководство (Резултати)

- Достъпване на функционалност при **десктоп** версията (Desktopapp use cases)
 - Вход на служител в системата;

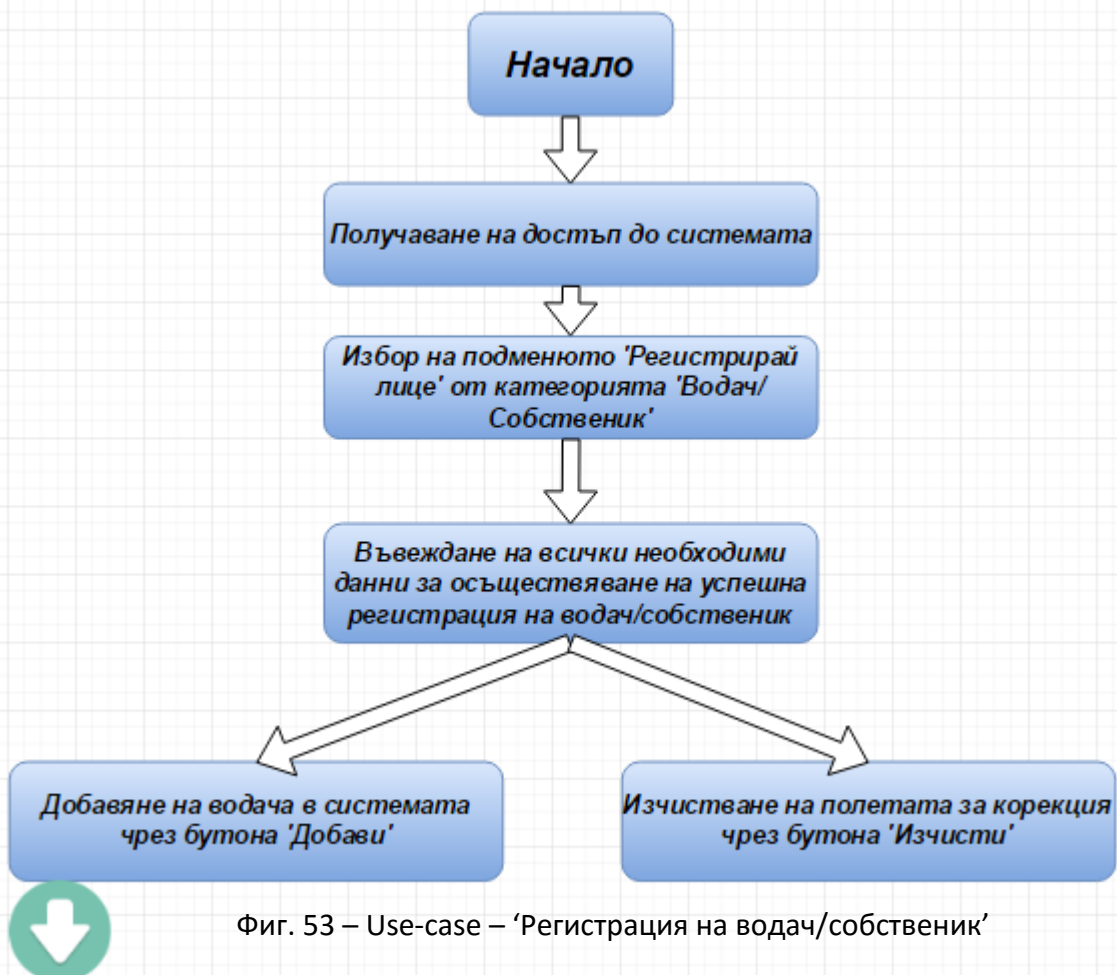


Фиг. 51 – Use-case – ‘Вход в системата’

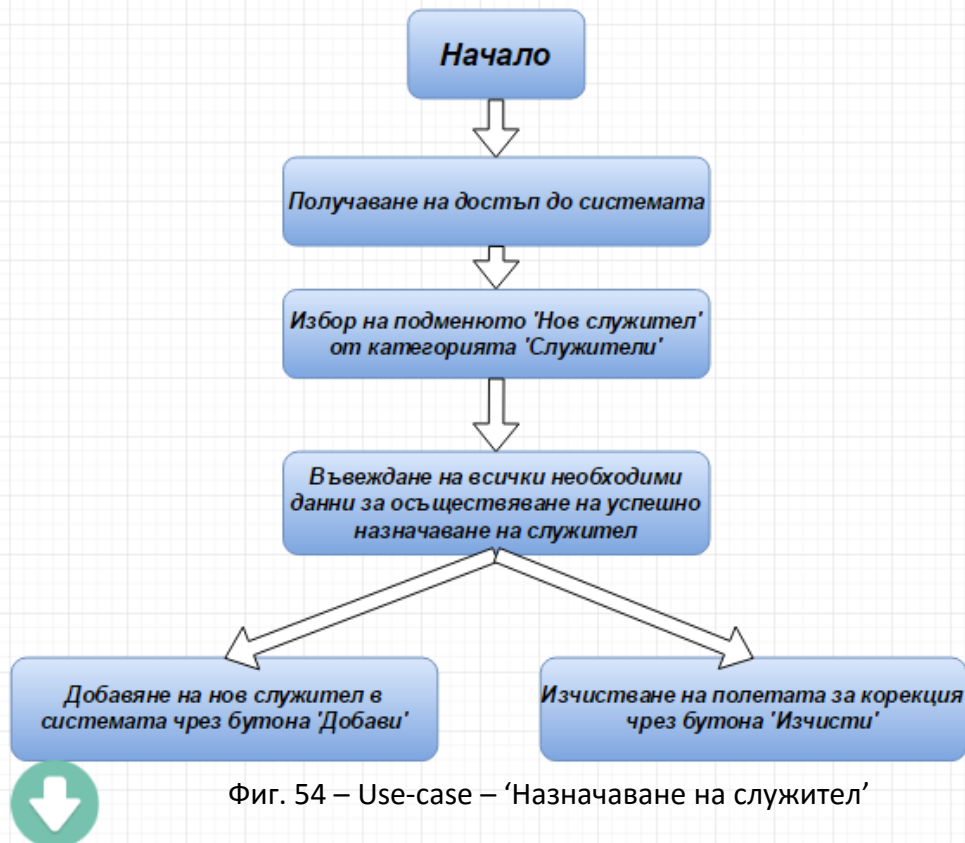
- Нова регистрация на МПС;



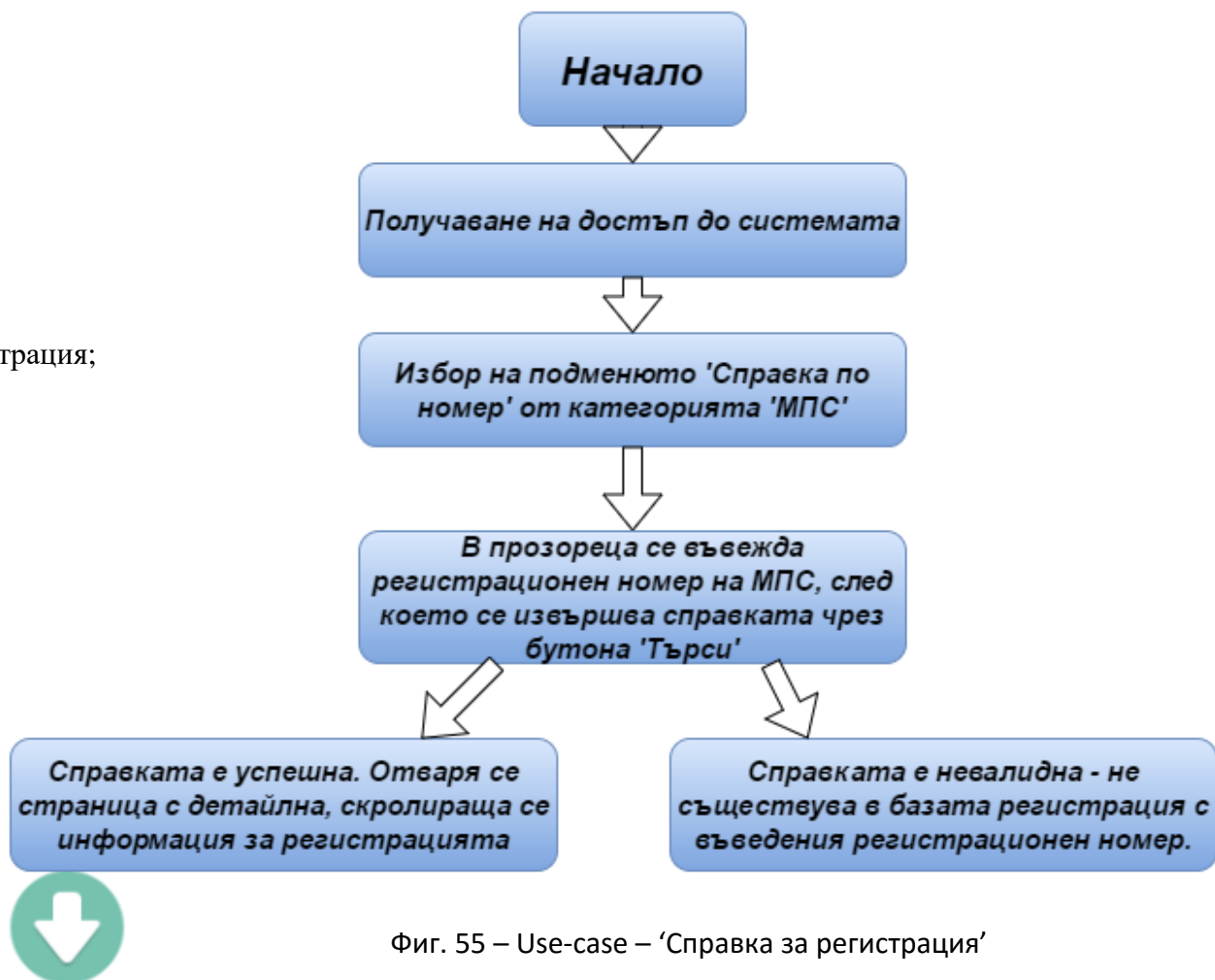
- Регистрация на водач/собственик;



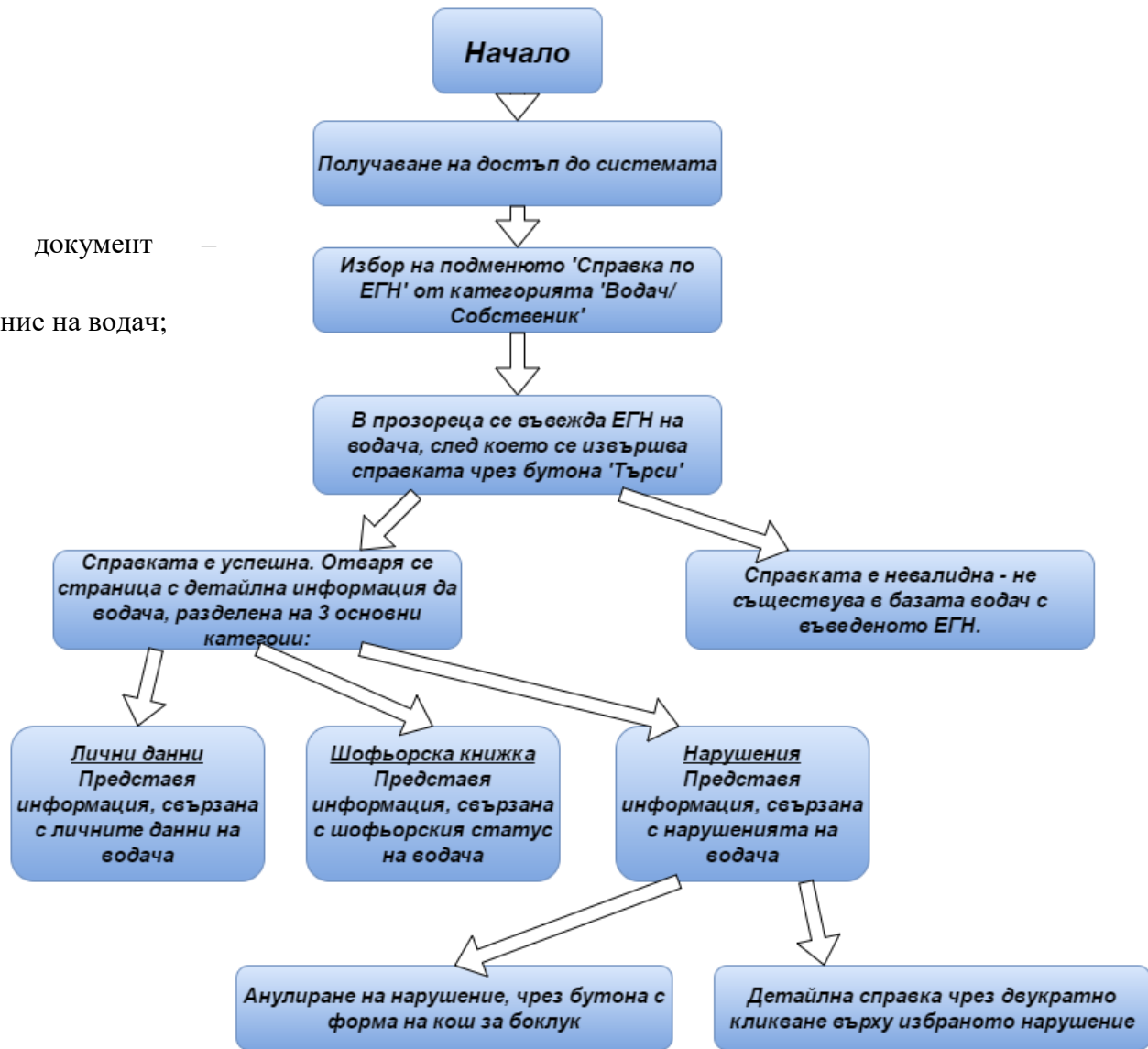
- Назначаване на служител;



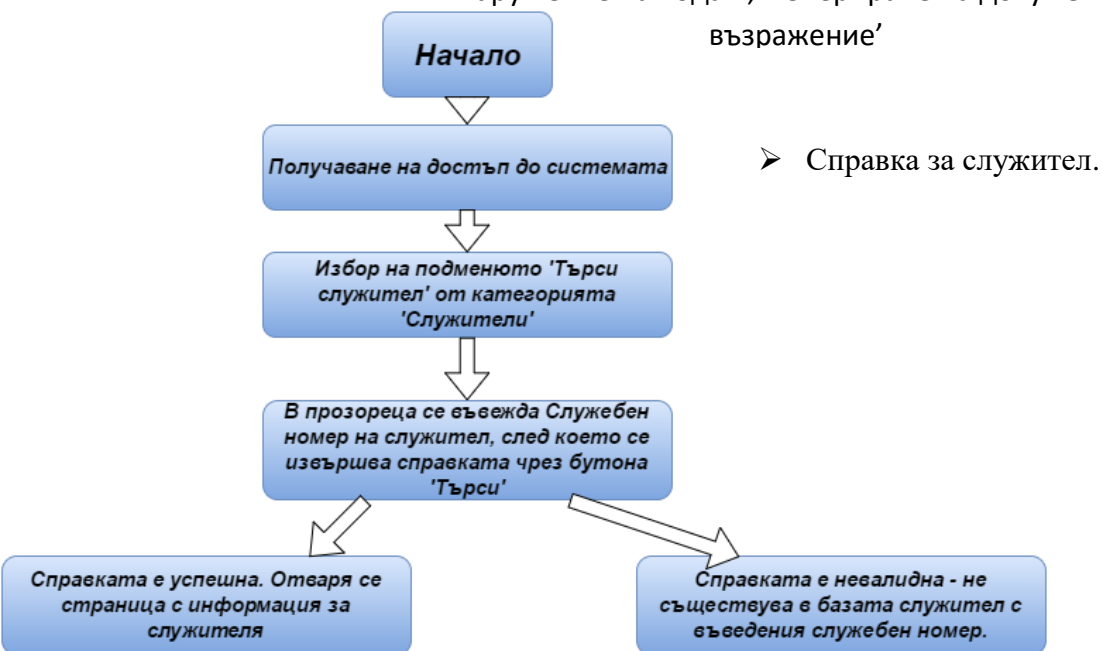
- Справка за регистрация;



- Справка за водач;
- Генериране на документ – възражение;
- Справка за нарушение на водач;



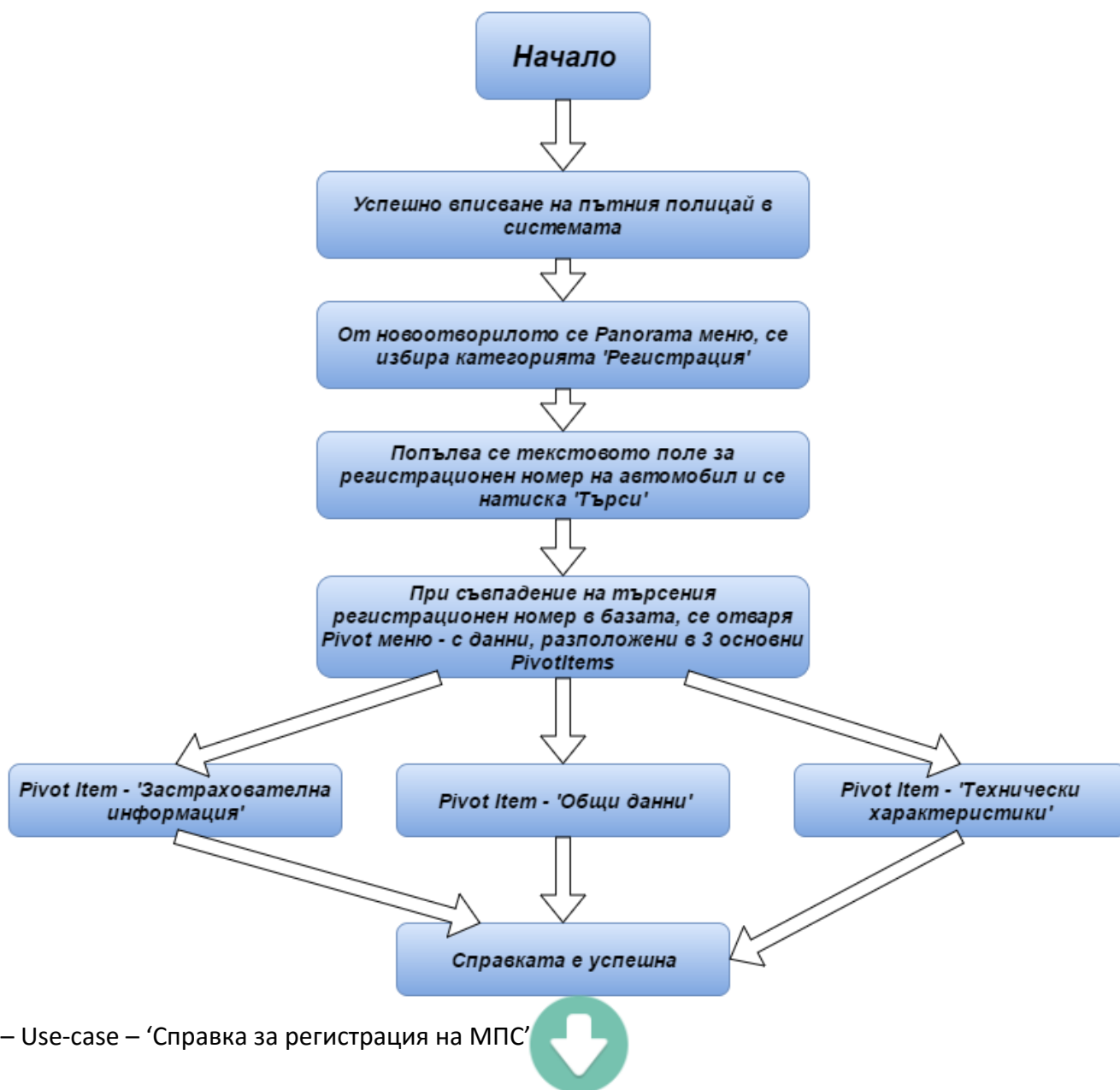
Фиг. 56 – Use-cases – ‘Справка за водач’, ‘Справка за нарушение на водач’, ‘Генериране на документ – възражение’



Фиг. 57 – Use-case – ‘Справка за служител’

- Справка за служител.

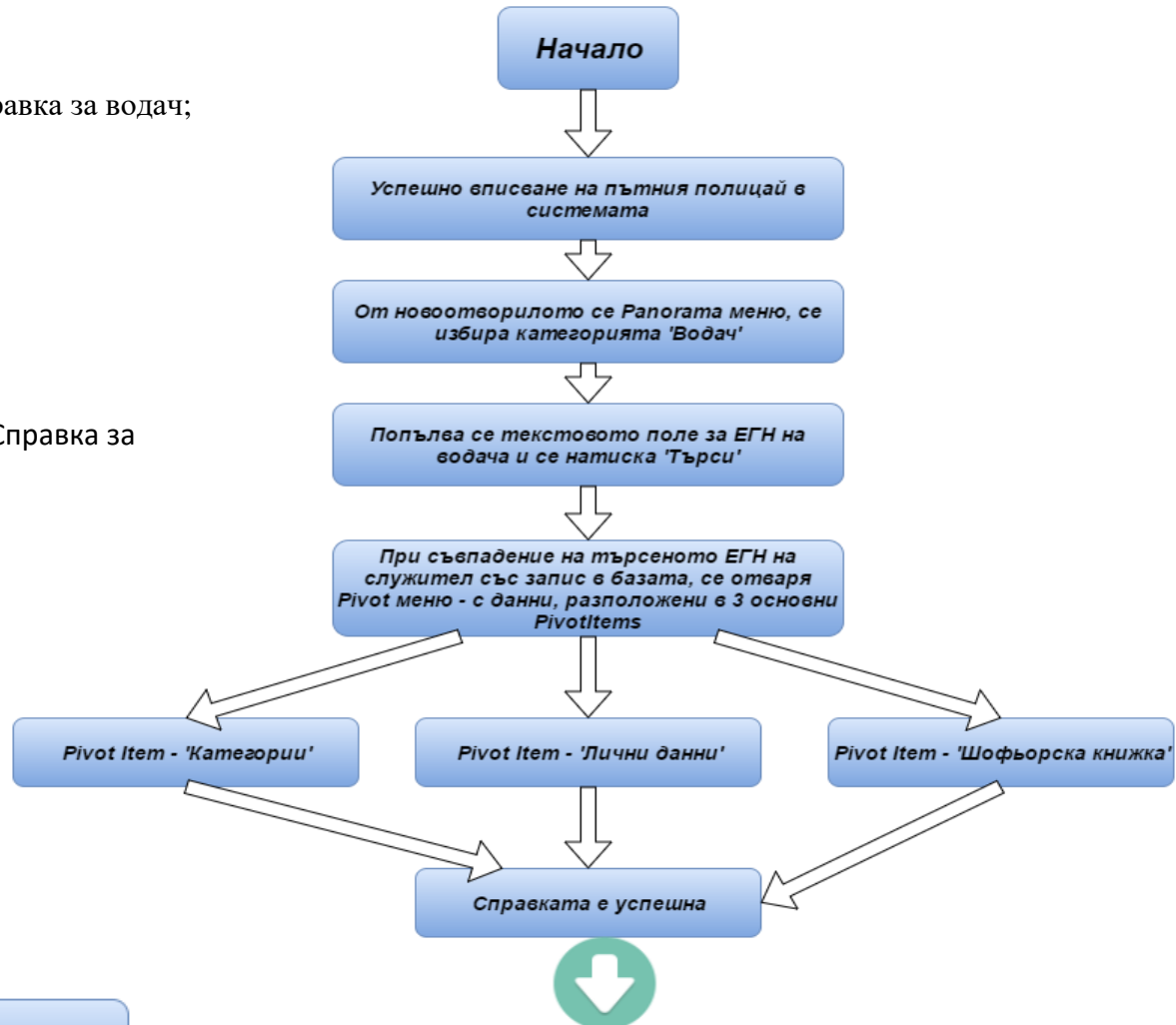
- Достъпване на функционалност при *мобилната* версия (Mobile app use cases)
 - Вход на пътен полицаи в системата – аналогично на десктоп действието;
 - Справка за регистрация на МПС;



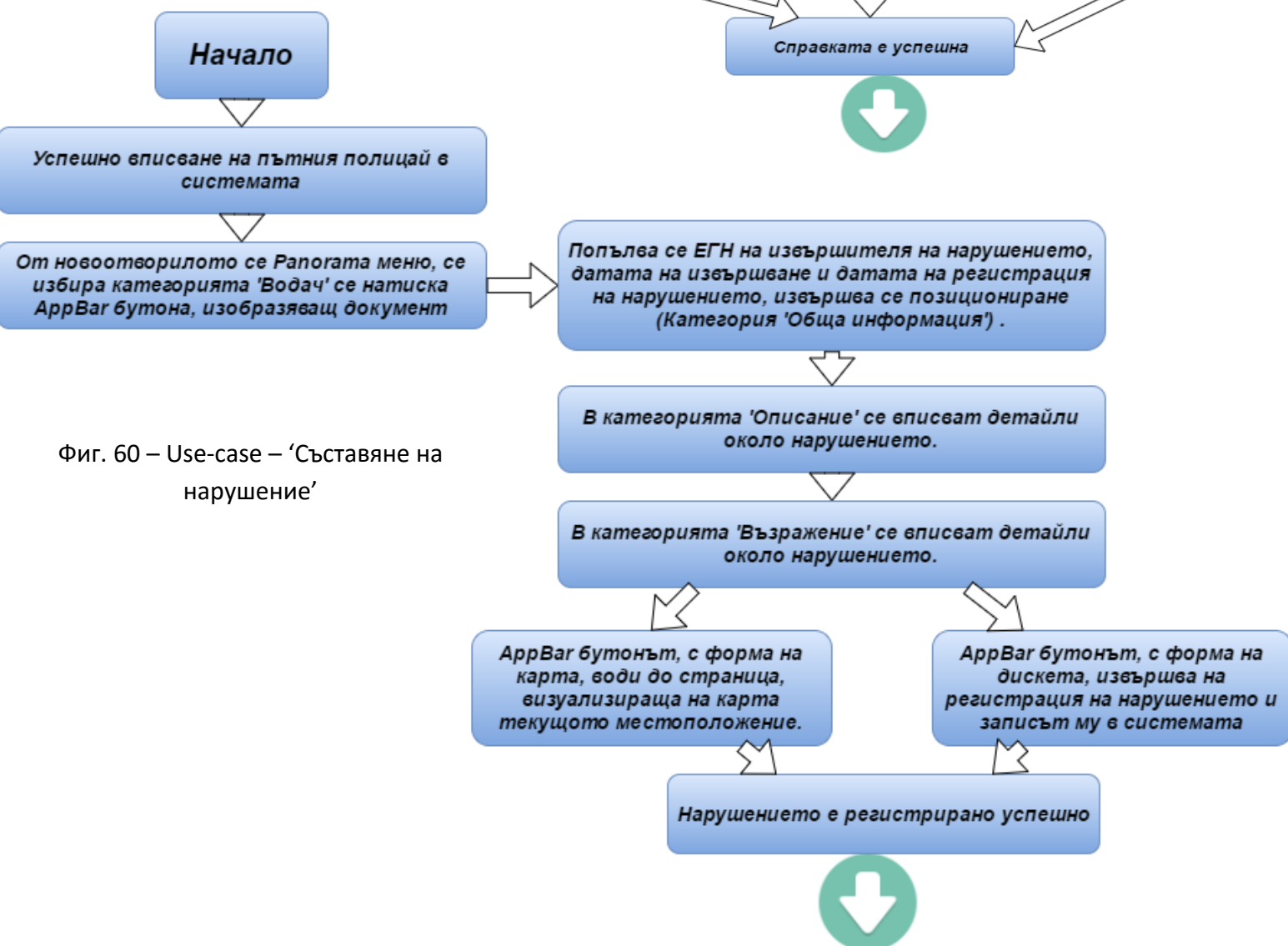
Фиг. 58 – Use-case – 'Справка за регистрация на МПС'

- Справка за водач;

Фиг. 59 – Use-case – ‘Справка за водач’



Фиг. 60 – Use-case – ‘Съставяне на нарушение’



Заклучение

Приложението премина успешно предвидените тестове на разработчика преди официалното си представяне. Основните функционалности са имплементирани и системата може да влезе в експлоатация, като съществува минимална възможност за възникването на непредвидени ситуации (бъгове), като в този случай, ви молим да докладвате за възникналия проблем, посредством предоставената форма.

Предстои в края на бета версията, да бъде разширен настоящия набор от функционалности, като се прибавят:

- Автоматична справка за нарушения при логване на администратор/служител в системата;
- Съхранение на снимкови материали за водач, служител и превозно средство;
- Автоматично уведомяване на нарушител по имейл за неплатени глоби;
- Предоставяне на актуална информация в десктоп приложението за местоположението на всеки един патрул върху детайлна карта. (При справка за служител).
- и други.

Опростеният интерфейс и минималният, но достатъчен набор операции прави настоящата система конкурентна на пазара. Разходите за обучение на персонала (служители – администратори и пътни полицаи) биха били минимални. Усвояването на умения за работа с приложението би бил един лесен и краткосрочен процес.

Използвани източници

- [1] **Microsoft Developer Network Library** - <https://msdn.microsoft.com>
- [2] Model-View-ViewModel Explained – <https://codeproject.com>
- [3] MVVM Cross – <https://github.com/MvvmCross>
- [4] Xamarin: Mobile App Development & App Creation Software– <https://www.xamarin.com/>
- [5] N+1 days of MvvmCross видео серии– <https://www.youtube.com/user/MrHollywoof/playlists>
- [6] MVVM Cross документация – <https://mvvmcross.com/docs>
- [7] Officer™ Traffic – <https://www.gtechna.com/product/officer-traffic-suite/>
- [8] CrimeStar™ – <http://www.crimestar.com>
- [9] eForce™ – <http://eforcesoftware.com/>
- [10] .NET MySQL Connector– <https://dev.mysql.com/downloads>
- [11] MySQL DB and Workbench – <https://www.mysql.com/>
- [12] Достъп до графичната среда за управление на базата данни– <http://www.phpmyadmin.co/>
- [13] Хостинг сървър, на който е разположена базата– <https://www.freemysqlhosting.net/>
- [14] Gmail SMTP настройка на сървър– <http://www.serversmtp.com/en/smtp-gmail-configuration>
- [15] Карти Bing Maps – www.bing.com
- [16] What is WPF? - The complete WPF tutorial – <http://www.wpf-tutorial.com/>
- [17] Помощни ресурси за .NET разработка– <http://www.tutorialspoint.com/>
- [18] .NET практически примери – <http://www.codeproject.com/>
- [19] How to Consume WCF Service in Xamarin.forms PCL ? – <https://forums.xamarin.com/discussion/19303/how-to-consume-wcf-service-in-xamarin-forms-pcl>
- [20] Expose WCF 4.0 Service as SOAP– <http://www.dotnetcurry.com/wcf/728/expose-wcf-service-soap-rest>


```
[ServiceContract]
public interface IService1
{
    // TODO: Add your service operations here
    [OperationContract]
    int UpdateUser(User user);

    [OperationContract]
    User GetUserByIdAndPass(string id, string password);

    [OperationContract]
    //Getting user data with nullified pass
    User GetReadOnlyUserById(string id);

    [OperationContract]
    string InsertUserAndGetGeneratedId(User usr);

    [OperationContract]
    int InsertCat(Categories categ);

    [OperationContract]
    int InsertRegistration(Registration reg);

    [OperationContract]
    int RegisterDriverOwner(DriverOwner drOwner);
    [OperationContract]
    DriverOwner GetDriverOwnerById(string id);

    [OperationContract]
    int removePenalty(Penalty pen);

    [OperationContract]
    List<string> getAvailableCarTypes();

    [OperationContract]
    Registration getRegByRegNum(string regNum);
}
```

Десктоп приложение – SOAP API Interface

```

[ServiceContract]
public interface IService1
{

    [OperationContract]
    User GetUserByIdAndPass(string id, string password);

    [OperationContract]
    DriverOwner GetDriverOwnerById(string id);

    [OperationContract]
    Registration getRegByRegNum(string regNum);

    [OperationContract]
    string addPenaltyToDriverOwner(Penalty pen);
}

```

Мобилно приложение – SOAP API Interface

```

void ConnectToDb()
{
    server = "sql7.freemysqlhosting.net";
    database = "sql7112557";
    uid = "sql7112557";
    password = "Fwg8uEkpT1";
    charset = "utf8";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";" + "CHARSET=" + charset + ";";

    connection = new MySqlConnection(connectionString);
}

//Open connection to database
private bool OpenConnection()
{
    try
    {
        connection.Open();
        return true;
    }
    catch (MySqlException)
    {
        return false;
    }
}

//Close connection
private bool CloseConnection()
{
    try
    {
        connection.Close();
        return true;
    }
    catch (MySqlException)
    {
        return false;
    }
}

```

Функционалност за връзка с базата данни

```
BEGIN

INSERT INTO penalties (user_id,driver_owner_id,date_time_issued,penalty_date_time,
                        location,latitude,longitude,description,disagreement)

VALUES(
    (SELECT users.user_id FROM users WHERE users.user_id=user_id),
    (SELECT driver_owners.driver_owner_id FROM driver_owners WHERE
driver_owners.driver_owner_id=driver_owner_id),
    date_time_issued,
    penalty_date_time,
    location,
    latitude,
    longitude,
    description,
    disagreement
);

END
```

MySQL процедура за съставяне на
нарушение на водач

```

public string addPenaltyToDriverOwner(Penalty pen)
{
    //Test
    string insertQuery = "CALL
add_penalty_to_driverowner(@user_id,@driver_owner_id,@date_time_issued,@penalty_date_time," +
"@location,@latitude,@longtitude,@description,@disagreement)";

    //DB - Connected
    if (this.OpenConnection() == true)
    {
        try
        {
            MySqlCommand cmd = new MySqlCommand(insertQuery, connection);

            //Test
            //Declaring query params
            cmd.Parameters.Add("@user_id", MySqlDbType.UInt64, 10);
            cmd.Parameters.Add("@driver_owner_id", MySqlDbType.UInt64, 10);
            cmd.Parameters.Add("@date_time_issued", MySqlDbType.DateTime);
            cmd.Parameters.Add("@penalty_date_time", MySqlDbType.DateTime);
            cmd.Parameters.Add("@location", MySqlDbType.VarChar, 100);
            cmd.Parameters.Add("@latitude", MySqlDbType.Double);
            cmd.Parameters.Add("@longtitude", MySqlDbType.Double);
            cmd.Parameters.Add("@description", MySqlDbType.TinyText);
            cmd.Parameters.Add("@disagreement", MySqlDbType.MediumText);

            //Setting params
            cmd.Parameters["@user_id"].Value = pen.IssuerId;
            cmd.Parameters["@driver_owner_id"].Value = pen.DriverOwnerId;
            cmd.Parameters["@date_time_issued"].Value = pen.IssuedDateTime;
            cmd.Parameters["@penalty_date_time"].Value = pen.HappenedDateTime;
            cmd.Parameters["@location"].Value = pen.Location;
            cmd.Parameters["@latitude"].Value = pen.Latitude;
            cmd.Parameters["@longtitude"].Value = pen.Longtitude;
            cmd.Parameters["@description"].Value = pen.Description;
            cmd.Parameters["@disagreement"].Value = pen.Disagreement;

            cmd.ExecuteNonQuery();
            return "SUCCESS";
        }
        catch
        {
            return "QUERY_ERROR";
        }
        finally
        {
            this.CloseConnection();
        }
    }
    else
    {
        //DB - Not connected
        return "DB_ERROR";
    }
}

```

Метод за съставяне на нарушение на водач,
извикващ съответната процедура