DronesPark

Курсова работа по Софтуерни архитектури и разработка на софтуер

Изготвили:
Янислав Янков, 62357
Павлина Тодорова, 62324

Съдържание

1. Въведение

a) Обща информация за текущия документ

i. Предназначение на документа

Документа представя софтуерната архитектура на система за следене и управление на свободните паркови места в големите градове.

ii. Описание на използваните структури на архитектурата.

За всяка структура се пояснява какъв точно аспект на системата показва, кои компоненти включва и какви са връзките между тях. Тук се описва и мотивацията за избор на съответната структура.

- **Декомпозиция на модулите** – показва обособяването на отделни модули (логически обособени единици) в рамките на системата. Най-общо системата е разделена на 4 модула: UI, Server, Drone, Data Base Configuration. UI визуализира системата чрез мобилно приложение и уеб приложение. Server осигурява бизнес логиката, обработката на всички заявки и връзката между останалите модули. Drone е основният движещ елемент, който заснема паркоместата. Data Base Configuration съхранява всички данни в системата и предоставя достъп до тях.

- **Допълнителни структури**
  - **Структура на процесите** - Изобразяват се нагледно взаимодействието между компонентите на изчислителните процеси в системата.
  - **Структура на потока на данни** - Представя през какви процеси преминават данните в системата и каква обработка получават.
  - **Структура на внедряването**- Показва как софтуера се разполага върху хардуерните елементи.

iii. Структура на документа

Документът е съставен от 5 секции:

- Въведение
- Представяне на структурата Декомпозиция на модулите
- Представяне на Допълнителни структури, които са Структура на процесите, Структура на потока от данни, както и Структура на внедряване
- Архитектурна обосновка, описваща архитектурните стилове и тактики, използвани в системата
- Допълнителна информация свързана с бъдещето на системата

b) Общи сведения за системата

В последните години намирането на места за паркиране в големите градове се превръща в голямо предизвикателство. Системата предоставя по-лесен, по-удобен и по-бърз начин

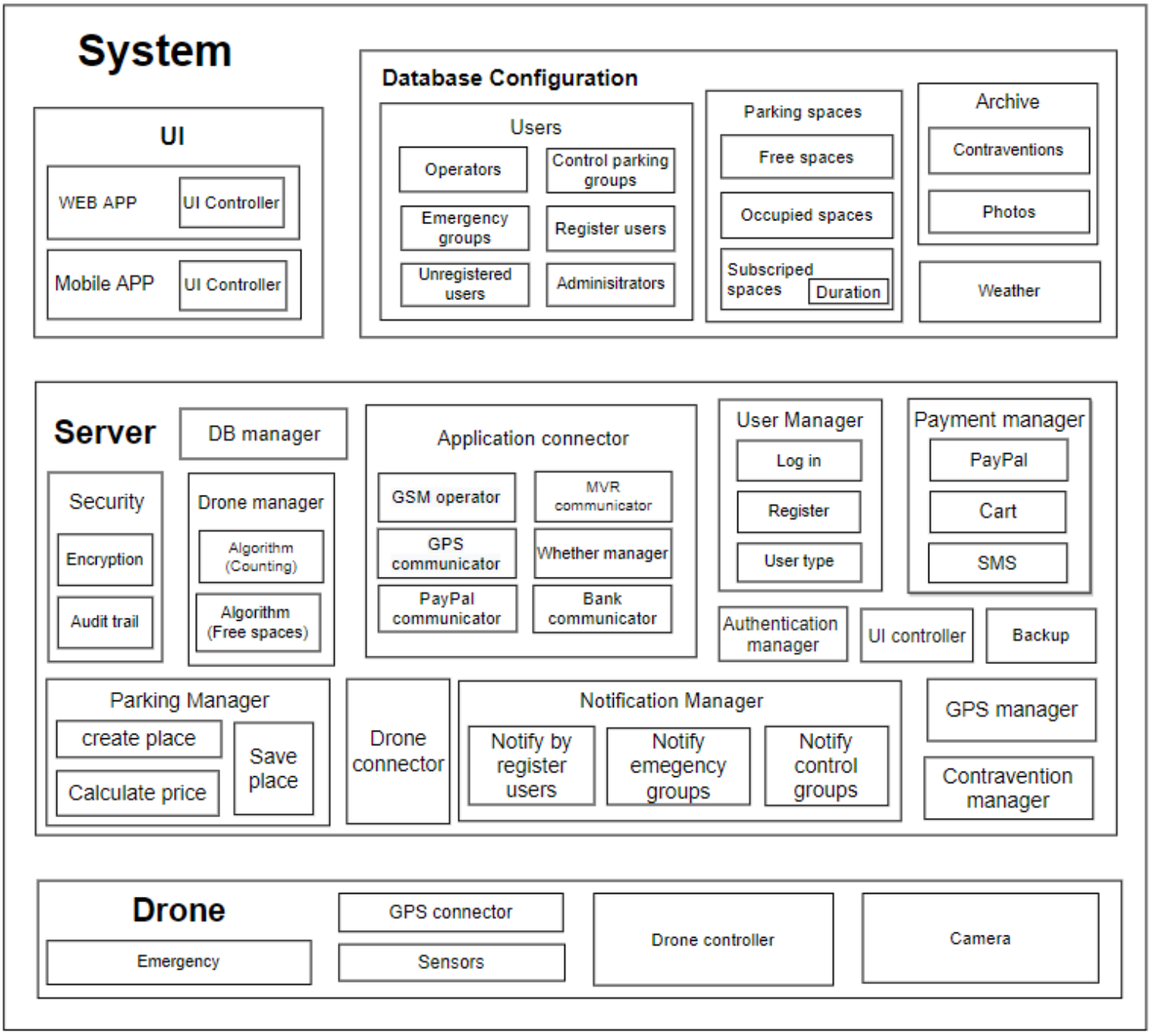
за намиране на свободно паркомясто, като за идентифициране на паркоместата се използва система от дронове. Основна цел е да се улесни живота на потребителите и да се намалят нарушенията по пътя. Системата поддържа 6 вида потребители- Администратор, Оператор, Аварийни групи, Групи по контрол на паркирането (т.нар. „паяци“), Регистрирани потребители и Обикновени потребители.

с) Терминологичен речник

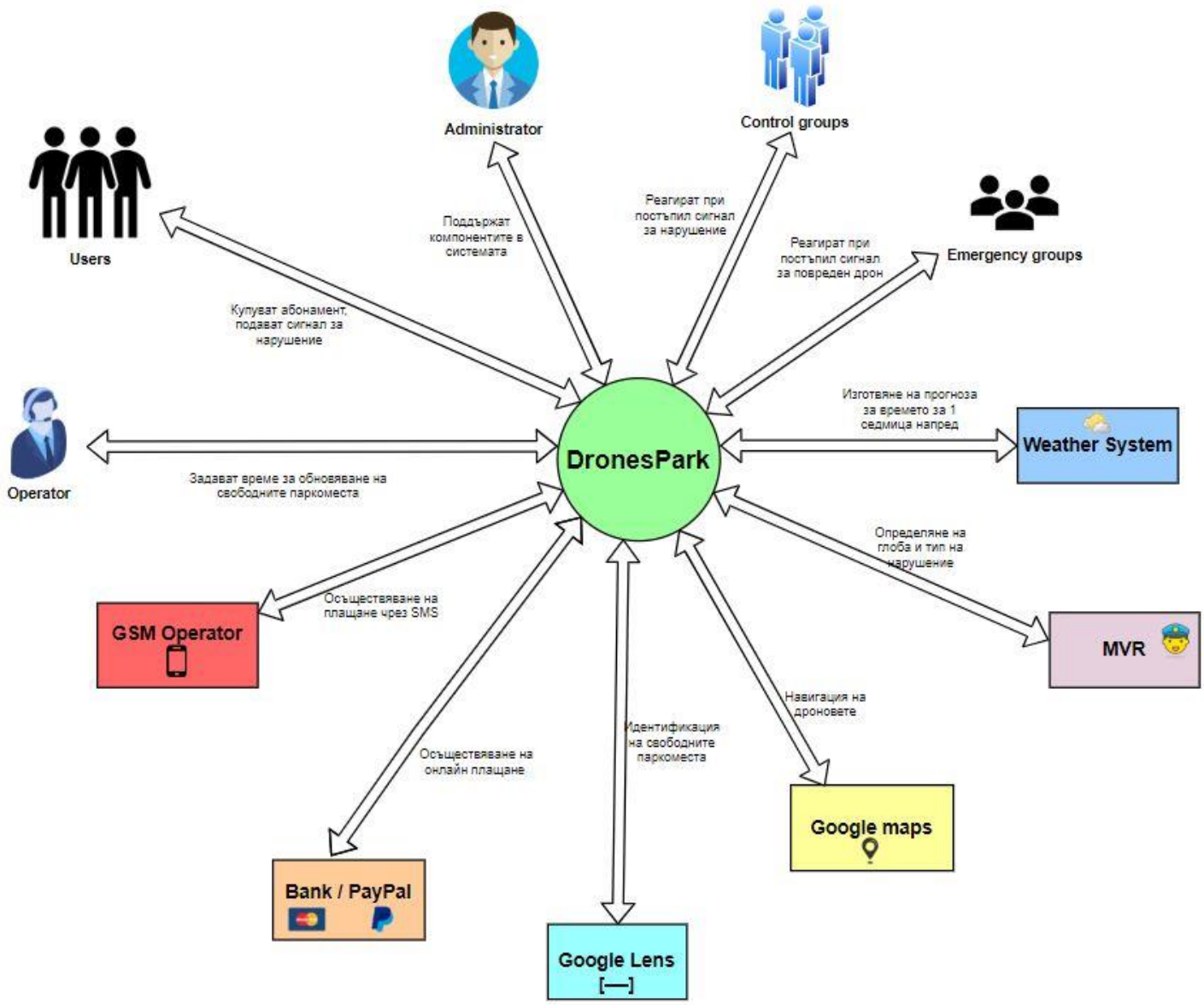
- Софтуер – набор от машинно-четими инструкции, които управляват действията на компютърния процесор
- Софтуерна система – комуникиращи помежду си компоненти, които са част от една компютърна система
- Външна система – отдалечена софтуерна система, която е източник на данни или функционалности, които се използват от настоящата система
- База от данни – колекция от информация, която е така организирана, че да може лесно да се достъпва, управлява и актуализира
- Кеш-памет - спомагателна памет за ускоряване обмена на данни между различните нива в йерархията на паметта.
- Криптиране - кодиране на значението на информация
- Декриптиране – разкодиране на значението на информация
- Потребител – човек, който използва компютърна или мрежова услуга
- Сървър – стартирана инстанция на софтуерна система, която може да приема заявки от клиент и да връща подходящи отговори
- Клиент – част от компютърна или софтуерна система, която достъпва услуга, предоставена от сървър
- Приложение (application) - софтуер, предназначен да помогне на потребителя да извърши определена задача
- WEB приложение (WEB application) – приложение, направено за компютър или лаптоп, достъпвано чрез браузър
- Мобилно приложение (mobile application) - приложение, направено за мобилни устройства
- Интерфейс (interface) – споделена граница, между която два отделни компонента на компютърна система си обменят информация
- Модул – логически обособена софтуерна единица
- Процес - съвкупност от стъпки, която изгражда логическо действие и стига определенасцел
- Декомпозиция - софтуерна структура, показваща как системата се разделя на отделни модули. Типовете елементи изграждащи тази структура са модули, а връзките между тях са от типа “X е подмодул на У”.
- Структура на внедряване – архитектурна структура, която показва връзката между софтуера и хардуера
- Android - Android е операционна система на Google Inc. за мобилни устройства, базирана на ядрото на Linux.
- iOS - iOS е мобилна операционна система на компанията Apple Inc.

2. Декомпозиция на модулите

а) Общ вид на декомпозицията на модули за системата



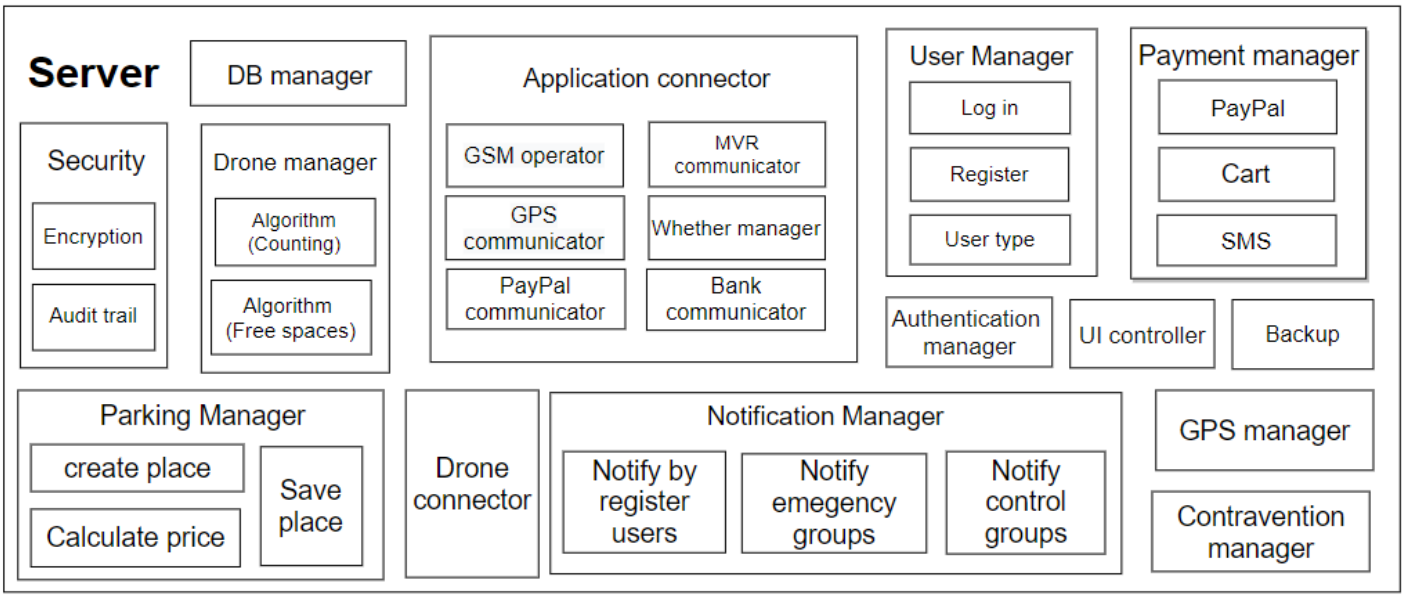
b) Контекстна диаграма



Системата ни взаимодейства с външни системи благодарение на модула Application connector в сървъра. Чрез него се осъществяват плащанията посредством SMS, като се свързва с мобилен оператор, посредством PayPal, като се свързва със системата на PayPal, както и чрез банкова карта, като си взаимодейства с Банка. Освен плащания, системата си комуникира и с външна система, прогнозираща метеорологичните условия за определен период от време. Навигирането на дроновете също се осъществява благодарение на външната система Google maps. Разпознаването на паркоместата е възможно, благодарение на Алгоритъма за идентифициране в Сървъра, който използва външната система Google Lens, за разпознаване на обекти. Фишовете и глобите при възникнало нарушение се изготвят от МВР.

c) Подробно описание на всеки модул

➢ Server



i. Предназначение на модула

Модулът представя бизнес логиката на системата. Играе роля на връзката между всички други модули. Основните отговорности на модула са да съдържа всички функционалности на система, както и да изчислява алгоритмите, които се съдържат в него. Също така трябва да приема и изпраща заявки към всички модули и да осигурява сигурност и отказоустойчивост на системата.

ii. Основни отговорности на модула в системата

- **Security** - модулът се грижи за сигурността на системата посредством криптиране на данните. Audit trail е компонент, който успешно се справя с възстановяването на информацията след атака.

- **Drone manager** - грижи се за това да поддържа тактиката Heartbeat. Също така в този модул се съдържат двата най-важни алгоритъма на системата. Първият взима под внимание интензивността на движението и метеорологичните условия, за да изчисли колко на брой дрона да се пуснат в движение. Вторият алгоритъм разпознава от какъв тип е паркомястото на всяка изпратена снимка (свободно, заето, с абонамент). След което изпраща снимката в базата данни.

- **Application connector** - модулът поддържа връзката с външните системи, с които системата комуникира. Weather manager изпълнява две задачи.

Първата и най- важната: прогнозата за времето, получена от външна система, се съхранява в базата данни, за да може Counting алгоритъма в Drone manager да функционира нормално. Освен информация за времето, въпросният модул сигнализира на Операторите при трайно намалена видимост (мъгла). Благодарение на MBP комуникатора е възможна синхронизация на цените на всеки фиш.

- **User manager** - един от най-важните модули. Той ще се грижи за потребителите в системата. Посредством UI се въвеждат данните от потребител, UI controller ги препраща към Register, който от своя страна добавя регистрираният вече потребител в базата данни. При влизане в системата, UI въвежда данните в системата, те минават през UI Controller-а, който ги изпраща към модула Log in. User type се грижи за определяне на достъпа на всеки потребител. След което Authentication manager-а прави проверка за съществуването на определения тип потребител в базата данни. Отделен е от User Manager с цел по-добра защита при опит за атака.

- **Backup** - всяка система среща трудности при съхранение и запазване на информация в хранилището. Често ставаме свидетели на загуба на данни, сривове или някакъв вид неизправности при обработката на данните. Нашата системата се придържа към Shared data style, тоест имаме добра централизираност, което ни дава добри условия да добавим модулът Backup. Той ще ни осигури копие на цялата информация постъпваща към базата данни. По този начин постигаме 100% наличност на системата в светлата част на деня.

- **Payment manager -** модулът дава възможност за извършвания на плащане от регистрираните потребители. UI приема заявката, след което UI Controller-ът насочва към конкретния начин на плащане, избран от потребителя. След което Payment manager взаимодейства със съответния подмодул на Application connector (например PayPal communicator), за да се осъществи транзакцията към дадената система, например PayPal.

iii. Описание на интерфейсите на модула.

- **Register** – Всеки нерегистриран потребител може да си направи регистрация в системата. Предимствата на регистрираните потребители, пред тези без регистрация са, че първите могат да купуват абонамент за дадено паркомясто за избран от тях интервал от време.

```
public bool initiateRegistration(string username, string email, string password);
private bool checkUsername(string username);
private bool checkEmail(string email);
private bool checkPassword(string password);
```

  - Вход: потребителско име, мейл, парола
  - Изход: Съобщение дали операцията е успешна

- **Login –** Всеки потребител с направена регистрация, може да влезе в профила си чрез въвеждане на име и парола.

  public bool login(string username, string password);

  - Вход: потребителско име, парола
  - Изход: Съобщение дали валидацията е успешна

- **Encryption –** За да постигнем по-добра сигурност криптираме данните за потребителя, постъпващи към базата данни

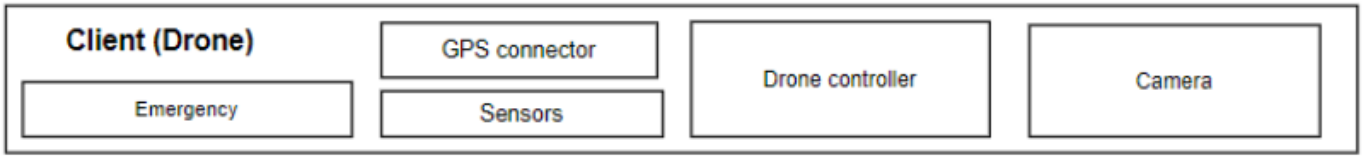
  public string encrypt(string password);

  - Вход: парола
  - Изход: Съобщение за успешно криптиране на парола

  public string decrypt(string encryptedPassword);

  - Вход: криптирана парола
  - Изход: Съобщение за успешно декриптиране на парола

➢ Drone



i. Предназначение на модула

Модулът представя основните компоненти във всеки дрон. Основните отговорности са свързани с правене на снимки на паркоместата, посредством камерата. Снимките се изпращат до сървъра чрез тактиката Heartbeat.

ii. Основни отговорности на модула в системата

- **Camera** - Най-важният модул във всеки дрон е камерата. Тя прави снимки на парковите места и моментално ги изпраща на Drone manager-a, който от своя страна разпознава какъв е типът на парковото място.
- **Sensors** - Модулът Sensors се грижи за ориентацията на дрона в околната среда. Например ако дрон се е насочил към клон на дърво, то сензорите ще го засекат и ще го заобиколят.
- **GPS connector** – свързан е с GPS manager-a в сървъра, като се грижи за направлението на дрона в небето.
- **Emergency** - играе роля на паник бутон, който се включва при повреда с дрона. При някакъв тип техническа неизправност.

iii. Описание на интерфейсите на модула.

➢ **Camera –** Камерата прави изображения на паркови места, след което ги изпраща на сървъра за обработка.

```
private void takePicture();
private void sendPicture(Picture picture);
```

➢ **GPS connector –** този модул се грижи дронът да поддържа зададения курс и съответно при неточно спазване на направление да подаде сигнал до Drone managere-a.

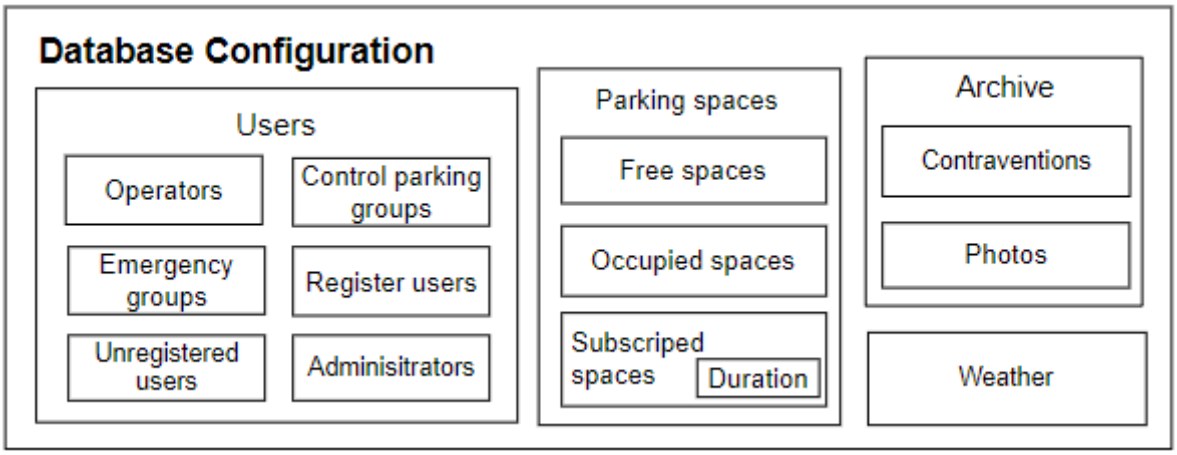
```
public void locateDrone();
private void sendSignalForProblem();
```

➢ Drone controller – изпраща всички заявки от дрона към сървъра

```
private void sendSignal();
private void sendPicture();
```

- Вход: снимка, сигнал
- Изход: информация за изпратената заявка (Успешно/Неуспешно изпратена)

➢ Database Configuration



i. Предназначение на модула

Модулът организира информацията постъпваща в базата данни. Целта на модула е да организира информацията, получена от сървъра в определени подмодули. Подпомага бързодействието и производителността, които са от първостепенно значение за нашата система.

ii. Основни отговорности на модула в системата.

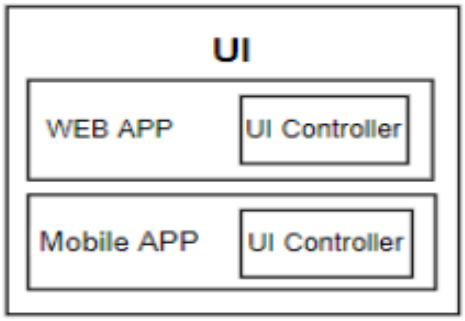
- **Users** - Модулът, в който се пази цялата информация за всички потребители се нарича Users. Подмодулите са Operator, Control parking, Emergency group, Unregistered user (Обикновени потребители), Administrator, Register users. Във всеки подмодул се пази информация за всеки тип потребител. В Unregistered user се пазят данни за подадени сигнали от потребители без регистрация.

- **Parking space** - Всички снимки на редовно спрелите автомобили се съхраняват в модула. Там имаме подмодулно разделение на Свободните места, Заетите места и местата с абонамент. В местата с абонамент освен снимка се пази й за какъв интервал от време е абонаментът.

- **Archive** - Системата съхранява освен всичко друго и нарушения, направени от шофьорите на автомобили за период от 25 години. Фишовете от направеното нарушение се съхранява в подмодула на Archive - Contraventions, като се има предвид, че глобата на фиша е синхронизирана със системата на МВР. В подмодулът Photos пазим снимките на изображенията от нарушенията.

- **Weather** - В модулът пазим информация за времето в период от седмица напред. Тази информация ни е нужна с цел безпроблемна работа на алгоритъма, изчисляващ какъв брой дронове са необходими в небето.

➢ UI



i. Предназначение на модула

Модулът визуализира всички функционалности на системата. Модулът отговаря за осъществяване на връзката между системата и потребителя. Системата използва мобилно приложение и уеб приложение, с което осъществява взаимодействието с шестте вида потребители.

ii. Основни отговорности на модула в системата.

Обикновените потребители, регистрираните потребители, аварийните групи и групите по контрол на паркирането могат да използват системата през мобилното приложение. Останалите потребители Оператори и Админи ще имат достъп до системата през WEB и Mobile APP.

- **Mobile App** - През мобилното приложение ще бъдат достъпни всички функционалности на системата, включително и преглед на снимки и фишове на всички нарушителите.

- **Web App** - WEB приложението се използва единствено за преглед на снимки и фишове на всички нарушители (не платили или заели място, за което нямат абонамент, неправилно паркиране и др.).

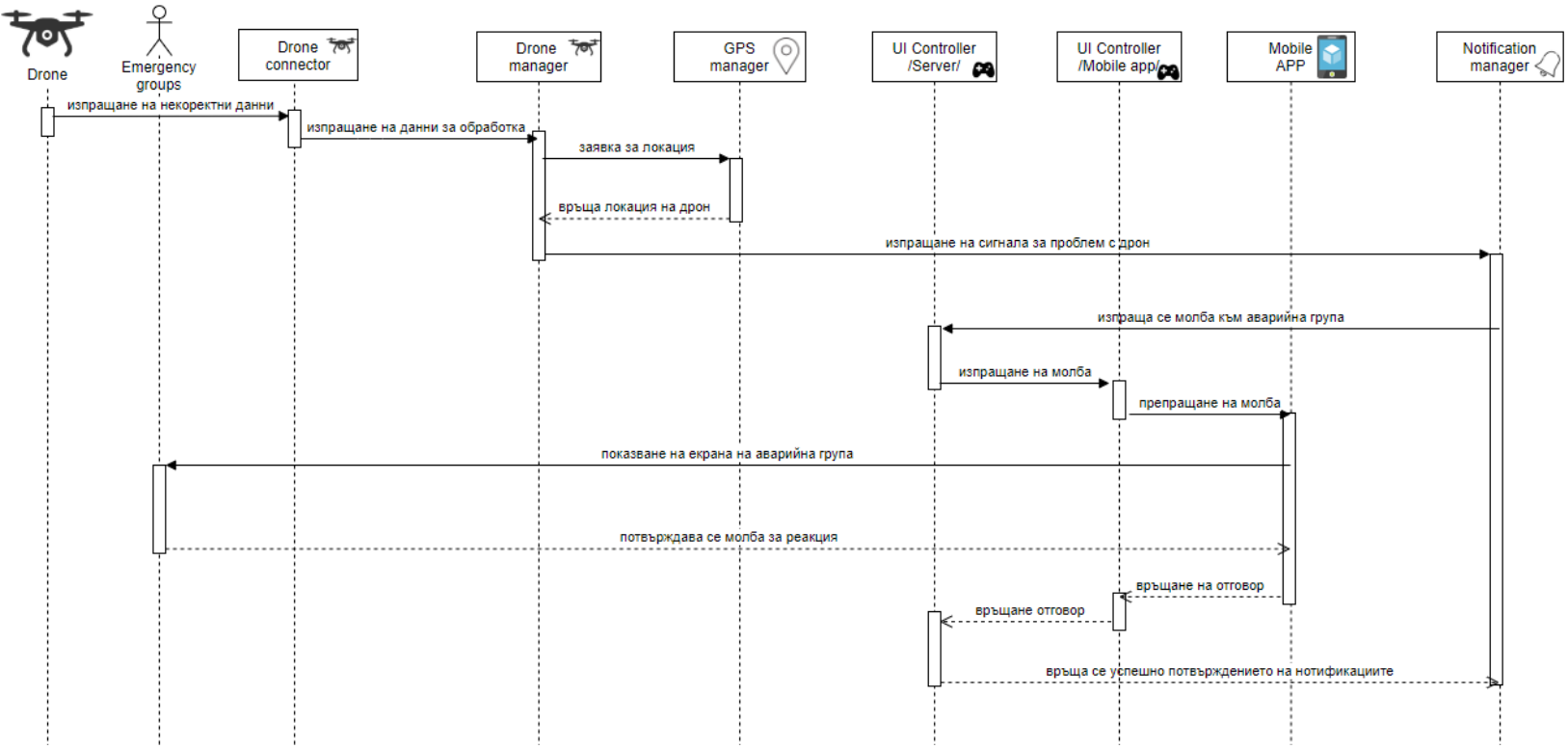
- Всеки модул има подмодул UI Controller, който се грижи за това всяка команда от UI да бъде пратена към сървъра. Модулът в сървъра, който се грижи за приемането се нарича съответно UI Controller, само че от сървъра.

iii. Описание на интерфейсите на модула. Описанието на всеки интерфейс съдържа:

Чрез модула се визуализират всички функционалности на системата. Като това се случва посредством телефон или компютър.

3. Структура на процесите (Sequence диаграма 1)

a) Първично представяне Sequence диаграма 1



Диаграмата представя процеса, в който се открива неизправност в работата на дрон. След което системата сигнализира на Аварийните групи за проблема.

b) Описание на елементите и връзките

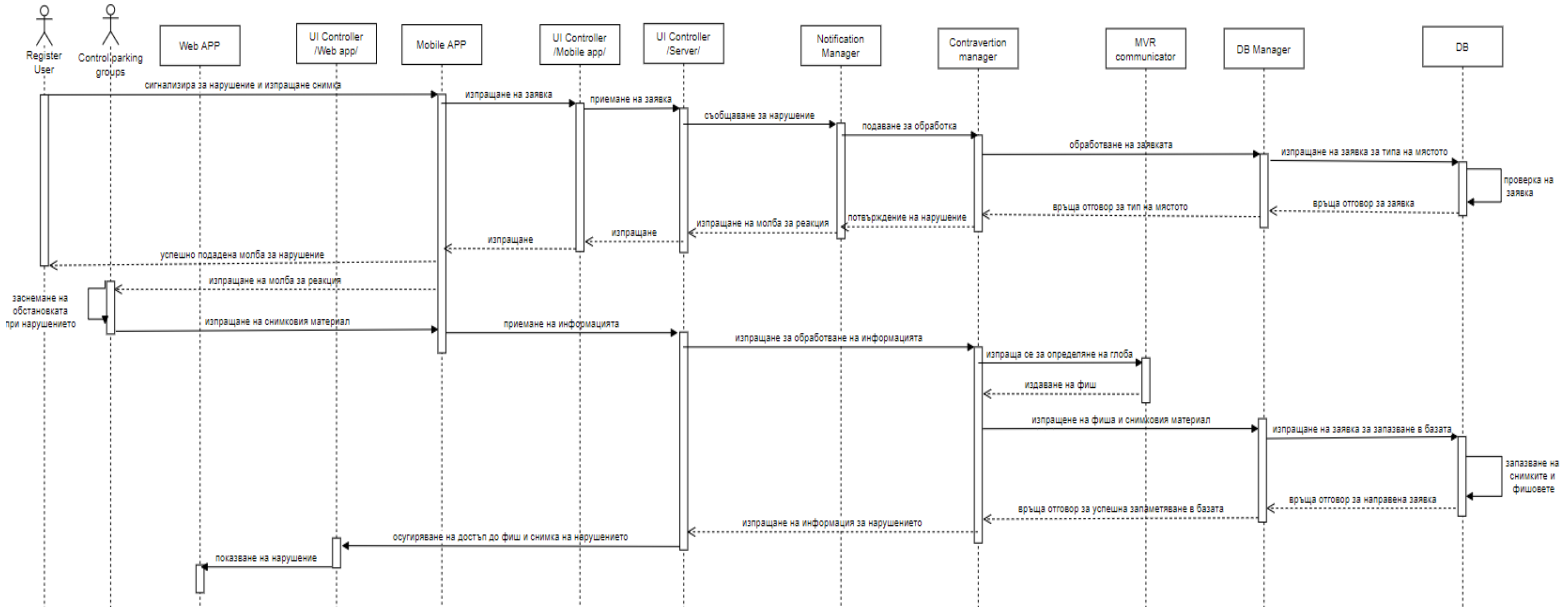
Всеки дрон се следи от системата посредством Heartbeat. Ако Drone manager-ът счита, че дрон не може да изпълнява посочените му от системата указания, то веднага праща заявка за локацията на дрона от GPS manager-а. След като получи местоположението, изпраща сигнал (включващ локацията на дрон) до Notification manager-а за проблем. Този модул се грижи да изпрати възможно най-бързо Emergency група, която да прибере дрона. Как става това? Notification manager-ът изпраща известие към най-близкия екип. Екипът (в перфектния случай) потвърждава заявката моментално и тръгва към въпросната локация. Потвърждението на Emergency групата се връща до Notification manager-а, чийто жизнен път приключва тогава. Ако не получи потвърждение в рамките на 1мин или Аварийната група откаже заявка, то Notification manager-ът подава нова молба за реакция към друг екип.

c) Описание на обкръжението

Системата си взаимодейства с GPS manager, който осъществява работа чрез връзка с външна GPS система.

4. Структура на процесите (Sequence диаграма 2)

a) Първично представяне Sequence диаграма 2



Диаграмата представя процеса, в който регистриран потребител подава сигнал за нарушение. Системата съответно изпраща група по Контрол на паркиране, за да установят нарушението и да се погрижат за нарушителя.

b) Описание на елементите и връзките

Всеки регистриран потребител с абонаментно място може да подаде сигнал при установяване на нарушение. (Например чужда кола е спряла на място, за което потребителят има абонамент). Заедно с подадения сигнал се изпраща снимка на мястото и нарушителя (регистрационния номер на автомобила). Системата проверява дали въпросното парково място е с абонамент и дали колата на него има право да спира там. При установяване на нарушението, системата изпраща молба за реакция до групите по контрол (тип паяк), както и съобщение за успешно подаден сигнал за нарушение до въпросния регистриран потребител. Първият екип, потвърдил молбата, се изпраща на мястото на нарушението. Екипът прави снимки на нарушението и ги изпраща към

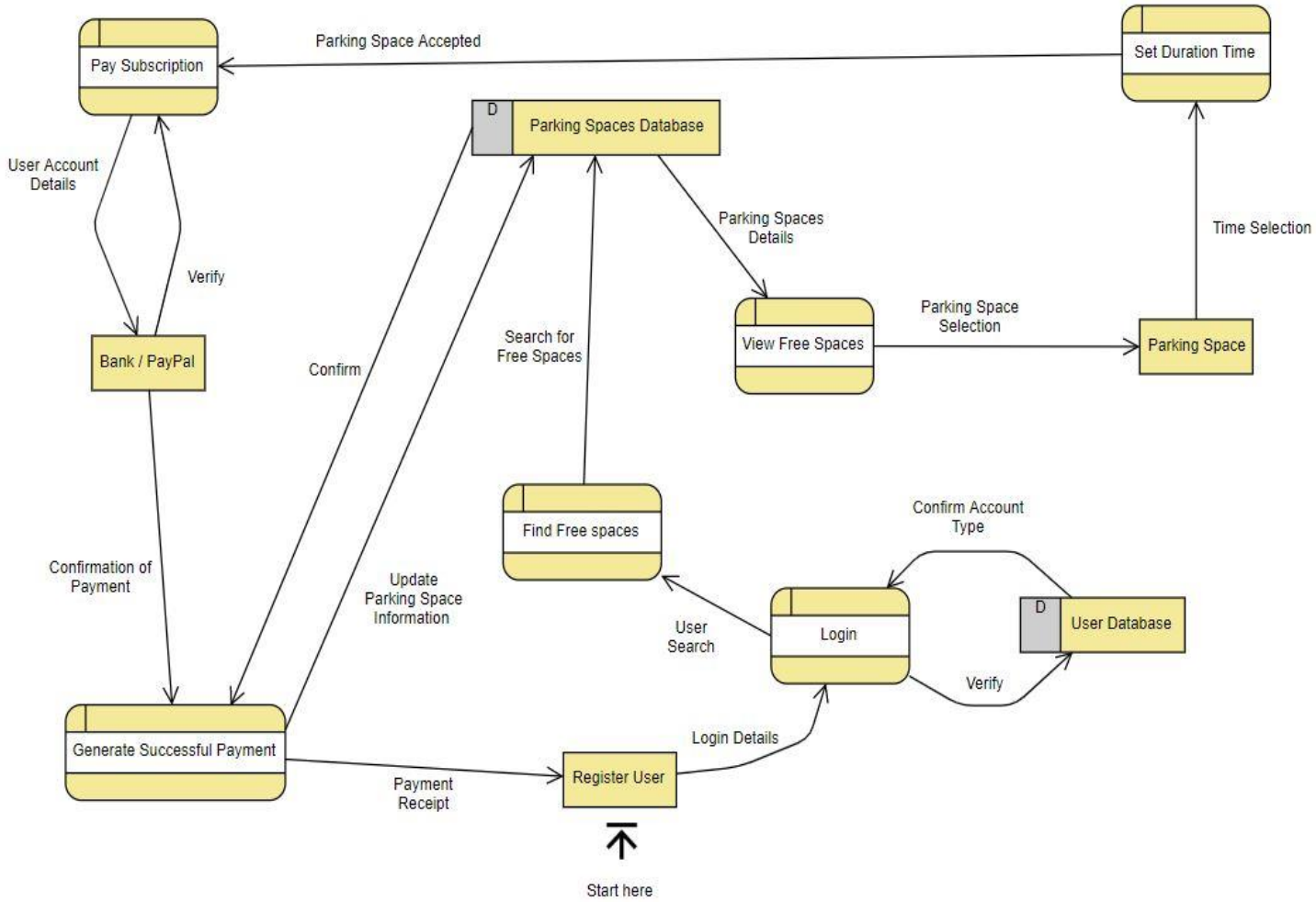
системата. Външна услуга MBP определя глобата, която да се начисли на нарушителя. След което се запазват фиша и снимката в Архива на базата данни, публикува се нарушението и в Web APP.

c) Описание на обкръжението

Системата си взаимодейства с външната система MBP, чрез която се определя какъв тип е нарушението и каква глоба трябва да се начисли на извършителя.

5. Структура на потока от данни

a) Първично представяне Data flow диаграма



Диаграмата представя как се обработват данните, когато регистриран потребител реши да си направи абонамент за дадено паркомясто.

b) Описание на елементите и връзките

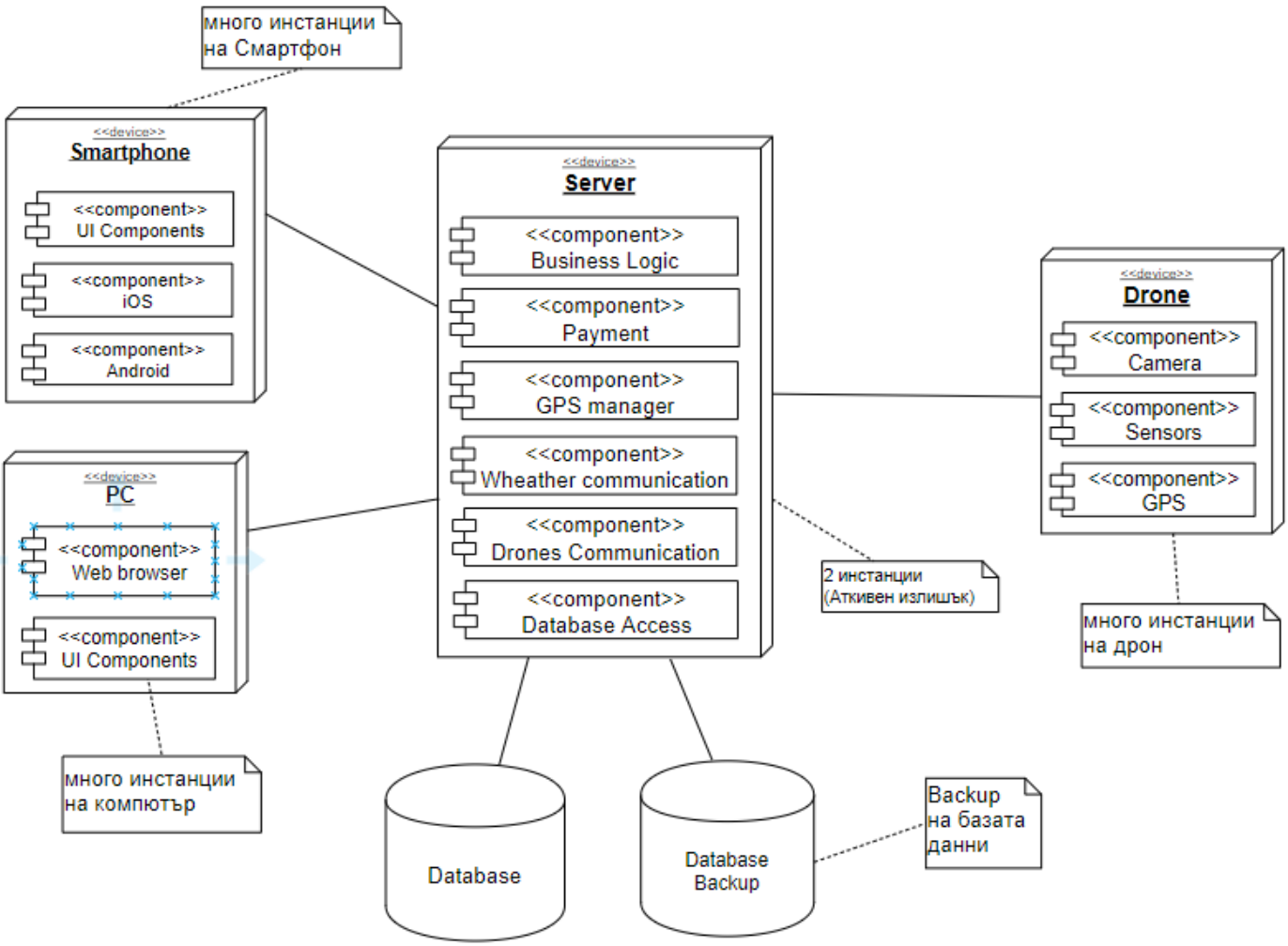
Всеки регистриран потребител може да се абонира за някое свободно паркомясто по свой избор. За да се осъществи това, регистрираният потребител първо трябва да се логне в системата. Извършва се проверка в базата данни за валидността на данните му (име и парола на потребител). След успешно влизане в системата, потребителят има възможност да потърси свободно паркомясто, за което да си направи абонамент. Системата проверява в базата данни всички свободни места, за които няма направен абонамент, след което ги генерира на картата. Клиентът избира най-удобното за него място. След това системата предоставя възможност за избор на времето, за което да бъде активен този абонамент. Потребителят прави своя избор и преминава към плащане. То може да се осъществи по 2 начина: чрез кредитна/дебитна карта или PayPal. (Със СМС могат да се заплатят само временни паркоместа без абонаменти.) Потребителят въвежда информация, необходима за осъществяване на плащането, която се верифицира от съответния орган (Bank/PayPal). След което заплаща фиксирана сума спрямо абонамента. Съответното Entity връща потвърждение за направено плащане. При успешна транзакция, системата запазва данните (паркомясто и duration) за абонамента в базата данни. След потвърждаване, че информацията е обновена, системата изпраща на регистрирания потребител съобщение за успешно заплащане и активиран абонамент.

c) Описание на обкръжението

Системата си взаимодейства с различни външни системи - банки, PayPal и GSM оператори, чрез които се осъществява успешното заплащане на паркоместата.

6. Структура на внедряване

a) Първично представяне Deployment диаграма



Диаграмата представя как се разполагат софтуерните компоненти върху хардуерните устройства.

b) Описание на елементите и връзките

Системата е достъпна за работа от смартфони, използващи Android и iOS, както и за компютри, разполагащи с Web browser. В сървъра се съдържа бизнес логиката на системата, както и различни комуникатори с външни системи. Дроновете са устройства,

които използват 3 компонента. Първият отговаря за направата на снимки, вторият за ориентацията в околната среда, а третият за спазването на точно направление. Базата данни ще е на физически носител на езика MSSQL. Сървърът се свързва с дроновете, смартфоните и компютрите чрез HTTP протоколи. Взаимодействието между сървъра и базата данни ще се осъществява чрез OLE-DB access.

c) Описание на обкръжението

Системата си взаимодейства с външна услуга за прогноза за времето, външна GPS система, МВР, както и с различни системи за плащане – банки, PayPal и GSM оператори.

7. Архитектурна обосновка

а) Архитектурни изисквания:

Пояснение: Всички архитектурни драйвери измежду изискванията са изписани с удебелен шрифт.

1. Свободните паркоместа се идентифицират от система от дронове, които обикалят града и заснемат зоните за паркиране (отгоре).

Системата, която разработваме има за цел да улесни паркирането в големите населени градове. Основната и най-важна нейна характеристика е да идентифицира свободните паркоместа чрез използване на дронове, обикалящи определени зони в града. Към всеки дрон ще бъде прикрепена камера "Leica" с качество на видеото UHD 4K за видеонаблюдение в реално време.

2. Броят на летящите в момента дронове и маршрутът на всеки от тях се определя динамично, на базата на предвиждане, за честотата на заемане/освобождаване на места в съответните зони. Това предвиждане зависи от натрупаните данни за динамиката на паркиране в съответния ден и час от седмицата и метеорологичните условия.

До тук имаме основната идея на проекта, която ще се реализира чрез дронове. Второто основно изискване е свързано с това как ще се осъществи работата им. Динамичното определяне на техния маршрут ще става на базата на „предвиждане“. Ще бъде използвана външна услуга- Google maps, чрез която ще се следи местоположението на дрона и ще се навигира посоката му на движение. Определянето на броя на активните устройства в небесното пространство за даден район ще се извършва благодарение на Counting algorithm, базиран на 3 три основни фактора - локация, трафик и метеорологични условия.

3. За определяне на метеорологичните условия да се ползва външна услуга за прогноза за времето.

Системата ще разполага с модул Application connector, чрез който ще си взаимодейства с външните системи. Благодарение на него ще се свързваме със система, определяща прогнозата за времето за седмица напред. Чрез данните за времето ще работят алгоритмите в DronesPark.

4. Системата използва специфичен алгоритъм за разпознаване на свободните места, на база на заснетите изображения.

За да гарантира висока производителност, системата ще включва алгоритъм идентифициращ свободните места, който се уповава на снимков материал. Алгоритъмът ще работи посредством Google Lens (външна система разпознаваща над 1 милион обекта).

5. Системата поддържа следните групи потребители:

a. Администратор

b. Оператор

c. Аварийни групи

d. Групи по контрол на паркирането (т.нар. „паяци“)

e. Регистрирани потребители

f. Обикновени потребители

Това е може би най-важното изискване за целия проект. В приложението ще имаме шест групи потребители. Системата трябва да поддържа различни нива на достъп спрямо своя ползвател. Информацията за потребители ще бъде пазена в базата данни. Потребителският контрол ще се осъществява чрез Администратор на системата, който ще има пълен достъп до правата на всеки регистриран потребител.

6. Ако някой дрон излезе от строя, незабавно трябва да се уведомят аварийните групи, които да получат информация за предполагаемия район, в който се намира дрона и да отстранят повредата.

Връзката между всеки дрон и сървъра ще се осъществява посредством Heartbeat. Ако има проблем с пращането на информация или с нейното състояние, Drone manager-а в сървъра ще се свързва чрез компонента Notification manager с Аварийна група, която ще прибира повредения дрон. (Целия процес е представен чрез Структура на процесите в точка 3.)

7. Информацията за свободните места се обновява на определен интервал от време, който се задава от оператора на системата и може да е най-малко 1 минута.

Операторът ще разполага с функционалност, предоставена от User type, даваща му права да определя интервала за обновяване на свободните места. Колкото по-близък е до 1мин, толкова по-добре за потребителите на системата.

8. При трайно намалена видимост (напр. мъгла), която води до невъзможност да се заснемат паркоместата, да се вземат мерки за известяване на оператора на системата.

При трайно намалена видимост, системата ще осведомява оператора. След което ще се прави оценка на метеорологичната обстановка и при нужда ще се прибират частично или изцяло дроновете от районите.

9. Регистрираните потребители могат да заплащат абонамент за определено парко-място, което се маркира като заето в рамките на периода на абонамента, независимо дали заснетите от дроновете изображения, показват наличието на автомобил на него или не.

В зоните, където ще се интегрира проекта, ще има много живущи и работещи хора с нужда от постоянно място за паркиране. Това ще е възможно чрез специална опция наречена платен абонамент. Системата трябва да съдържа база данни, която да съхранява потребителите с абонамент, както и информация за местоположението на парковото място и времето, за което абонаментът ще бъде активен. За да се избегне заплащане на абонамент, за дадено място, което в момента е динамично заето, активирането на абонамент ще бъде възможно най-рано на следващият ден.

10. Ако няма абонамент, свободните места за парикране може да са безплатни или да се таксуват динамично, като цената се определя според предвиждане за честотата на заемане/освобождаване в съответния ден/час, както и от прогнозата за времето.

Това е важен архитектурен драйвер, тъй като определя видовете такси за всяко паркомясто. Всеки потребител, който няма абонамент, ще трябва да заплати съответна такса в приложението. Методологията, по която се определя тази такса ще е обособена като отделен модул, изчисляващ динамично честотата на паркиране. Потребители спрели на места, обозначени като „безплатни“, няма да се таксуват от системата.

11. Плащането може да се извършва чрез дебитна/кредитна карта, PayPal или СМС, като в бъдеще може да се добавят и други начини на разплащане.

Използването на системата от потребителите е пряко свързано с начина на плащане. Всеки потребител е длъжен да заплати определената от приложението цена. Това може да се осъществи по достъпен и лесен за потребителите начин чрез банкова карта и PayPal при заплащане на абонамент.

Освен това потребителят трябва да има възможност да плаща своята такса със СМС през мобилен оператор при динамично таксуване.

12. Обикновените потребители, регистрираните потребители, аварийните и групите по контрол на паркирането използват системата през мобилно приложение, като може да заемат само свободните места, за които няма абонамент.

Всеки от горе изброените потребители ще има достъп до нашата система през мобилно приложение, стига да имат операционна система Андроид или iOS. Също така функционалностите, които могат да използват ще се управляват от модула User type в Сървъра.

13. Останалите потребители трябва да имат 100% защитен от външна намеса достъп до системата.

Системата трябва да разполага с различни нива за сигурност, които гарантират 100-процентова защита от външна намеса. Криптиране на данните, компонентът audit trail, backup на базата данни, както и втора инстанция на Сървъра, благодарение на тактиката Активен излишък.

14. Групите по контрол на паркирането следят дали няма нарушители (не платили или заели място за което нямат абонамент). При засичане на нарушител, освен принудителното преместване на автомобила, заснемат настъпилото събитие, като снимката се съхранява директно в системата и след това се издава електронен фиш за глоба. Снимката и фишът трябва да са достъпни и през публичен сайт, който се зарежда чрез уеб-браузър.

Една от най-важните характеристики за нашата система е да има контрол. Избраното изискване е драйвер, тъй като разработчиците трябва да предвидят място за съхранение, в което да се пазят снимките на нарушителите. Нужна е връзка със системата на МВР, с която да се синхронизират цените на фишовете за всяко нарушение. Освен всичко друго разработчиците имат и задачата да създадат публичен сайт, зареждащ се от уеб браузър, в който да се публикуват снимките и фишовете.

15. Регистрираните потребители може да подават сигнал до групите по контрол на паркирането за неправомерно заето от друг място, за което са абонирани.

Всеки регистриран потребител може да подава сигнал за нарушение чрез своя телефон. Прави снимка на нарушението, системата проверява дали мястото е с абонамент и ако нарушението се потвърди, то следва реакция от групите по контрол на паркирането. (Целият процес е представен в точка 5 на документа чрез Структура на процесите.)

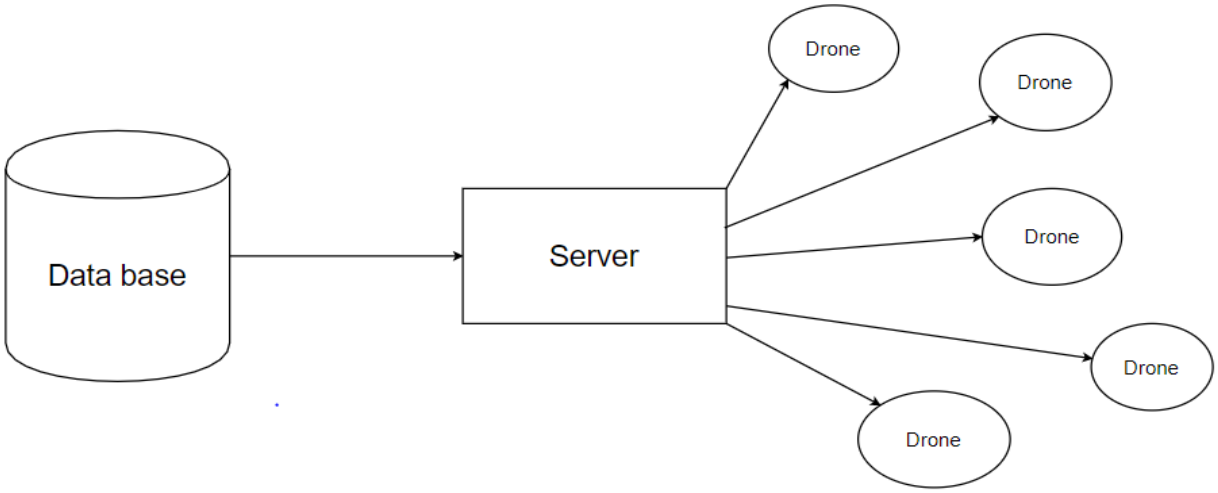
16. Системата да работи 100% без отказ в рамките на светлата част на работния ден (9:00 до 17:00 зимно време и 8:00 – 19:00 лятно време).

Изискването е от изключително значение за нашата система. Важно е системата да работи без абсолютно никакъв отказ, тъй като, ако настъпи срив, дроновете ще изгубят посоката на движение и загубите ще са огромни. Също така, всеки отказ губи доверието в потребителите и ще е добре този вид неизправност да се сведат до нула.

17. Системата да поддържа архив на данните за динамиката на паркирането и всички издадени фишове за глоби за 25 години назад във времето, както и архив на заснетите изображения за 3 години назад.

Изискването е драйвер, защото трябва да се предвиди място за съхранение, което да пази архив съдържащ информация за 25 години, както и архив за изображения за 3 години назад. Идеалното решение за съхранението на тези данни е в облачно хранилище (Dropbox, Metopal и др.). Това ще осигури бърз достъп, висока информационна безопасност и намален риск от загуба на данни.

b) Архитектурни стилове и тактики



Система DronesPark е съставена от два архитектурни стила: Shared data Repository и Three tier client-server model. Причината е, че имаме голям брой дронове, които се явяват клиенти на система, взаимодействащи със сървъра. От друга страна базата с данни е отделен модул от сървъра за по-добра производителност и сигурност. Благодарение на Shared data style може да добавяме нови дронове към системата, които започват да работят паралелно с останалите без да влошават производителността. Имаме централизирано управление на данни, което дава предпоставка за backup и по-добра сигурност.

Всеки дрон (клиент) ще споделя данни към сървъра на системата. Използваме Repository, тъй като комуникацията между всеки дрон е излишна. Те ще бъдат направлявани от сървъра, а информацията, която споделят ще бъде обработвана отново от сървъра. Този метод ще осигури перфектна синхронизация на всички компоненти.

Дроновете са изключително податливи на външни влияния. Поради тази причина има голяма вероятност да претърпят инциденти. Тактиката за изправност Heartbeat ни позволява да поддържаме връзка с всеки дрон. Това се случва посредством снимковия материал. Всеки дрон трябва да изпраща поне една снимка на кратък интервал от време, с което дава ясна представа за неговото състояние. Ако не намира свободно място за дълъг интервал от време, то преди да се изпрати Аварийна група, се преглежда местоположението му на GPS системата.

Системата ни до момента има централизираност, добра производителност, ефективност, както и добра синхронизация. Стабилността на системата и по-конкретно сървърът е под въпрос. Причината е, че разполагаме с един сървър, който играе главна роля в нашата система и при срив, системата ще спре работа . Това налага използването на тактиката Активен излишък, която ще ни осигури желаната отказоустойчивост. Системата ще разполага с допълнителен сървър, който е синхронизиран с основния. При отказ в системата, дублираният компонент ще е в готовност за експлоатация до няколко секунди. Това ще осигури 100% отказоустойчивост в рамките на светлата част на работния ден.

Системата ParkDrones ще разполага със снимков материал на хиляди автомобили. Цялата конфиденциална информация ще бъде защитена от хакерски атаки посредством криптиране на комуникационните канали и на постоянната памет. Ако нападател все пак успее да проникне в системата, то компонентът Audit trail ще се грижи за възстановяване на информацията след атаката.

8. Допълнителна информация

а) Бъдещи подобрения

Помислили сме и за бъдещи подобрения, които биха повишили бързодействието и производителността на системата.

1. Тактика Cache – Кеш е високоскоростен слой за съхранение на данни. Съхранява данни, които се достъпват по-често от потребителите, така че бъдещите заявки мога да се обслужват по-бързо. Това е така, защото няма нужда да се търси често използваната информация в базата данни, а се взима директно от кеша. Тази тактика подобрява производителността и бързодействието, което ще доведе до по-добра работа на системата.

2. Снимки от дроновете – В бъдеще ще се добави правене на снимки не само на едно място, а и на много места. Основното качествена характеристика, която ще се покачи от това подобрение, ще бъде бързодействието на системата

3. Използване на системата – За момента системата се планува да се използва за един град. В бъдеще, ако се хареса на потребителите, системата ще бъде достъпна и за много градове. Всяка област от градове ще се управлява от един сървър, който ще се свързва с всеки сървър на всеки град в областта. По този начин ще запазим централизираността и високата производителност ще остане непокътната, въпреки разширението на системата.