



Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

# Проект

по

Софтуерни архитектури и разработка на софтуер

на тема

**DronesPark – система за следене и управление на  
свободни паркоместа**



**Изготвили:**

Симона Любенова (62258)

Ати Мускова (62308)

## Съдържание

<b>1. Въведение</b>	<b>4</b>
1.1 Обща информация за текущия документ	4
1.1.1 Предназначение на документа	4
1.1.2 Описание на използваните структури на архитектурата	4
1.1.2.1 Декомпозиция на модулите	4
1.1.2.2 Структура на внедряването	4
1.1.2.3 Структура на процесите	4
1.1.3 Структура на документа	4
1.2 Общи сведения за системата	5
1.3 Терминологичен речник	5
<b>2. Декомпозиция на модулите</b>	<b>6</b>
2.1 Общ вид на декомпозицията на модули за системата	6
2.2 Контекстна диаграма	7
2.3 Потребителски интерфейс	7
2.3.1 Предназначение на модула	8
2.3.2 Основни отговорности на модула в системата	8
2.3.3 Описание на интерфейсите на модула	9
2.3.3.1 Потребителски функционалности	9
2.3.3.1.1 Избор на паркомясто	9
2.3.3.1.2 Плащане на паркомясто	9
2.3.3.1.3 Преглед на извършени плащания	10
2.3.3.1.4 Информация за паркоместа	10
2.3.3.1.4.1 Преглед на свободните паркоместа	10
2.3.3.1.4.2 Преглед на заетите паркоместа	11
2.3.3.1.4.3 Информация за избрано паркомясто	11
2.3.3.1.4.3.1 Преглед на оставащо време	11
2.3.3.2 Мениджър заявки	11
2.3.3.2.1 Одобрени заявки	11
2.3.3.2.2 Отхвърлени заявки	12
2.3.3.2.3 Заявка за паркомясто	12
2.3.3.2.3.1 Създаване на заявка	12
2.3.3.2.3.2 Отказване на заявка	12
2.3.3.2.3.3 Коментари към заявка	12
2.3.3.2.3.4 Потвърждаване на заявка	12
2.4 Сървър	13
2.4.1 Предназначение на модула	14
2.4.2 Основни отговорности на модула в системата	14
2.4.3 Описание на интерфейсите на модула	14
2.4.3.1 Конфигурация база данни	14

2.4.3.1.1	Защита.....	14
2.4.3.1.2	Отразени заплащания.....	15
2.4.3.1.3	Данни за паркоместа.....	15
2.4.3.1.4	Потребители.....	15
2.4.3.1.4.1	Данни за потребители.....	15
2.4.3.1.5	Архив.....	16
2.4.3.1.5.1	Данни за паркиране.....	16
2.4.3.1.5.2	Фишове за глоби.....	16
2.4.3.1.5.3	Снимки на нарушителите.....	16
2.4.3.1.5.4	Заснети изображения.....	17
2.4.3.2	Мениджър потребители.....	17
2.4.3.2.1	Влизане.....	17
2.4.3.2.2	Регистрация.....	17
2.4.3.2.3	Удостоверяване.....	18
2.4.3.3	Мениджър на данни.....	18
2.4.3.3.1	Информация от дрона.....	18
2.4.3.3.2	Информация от потребители.....	18
2.4.3.3.3	Достъпна информация за потребители.....	19
2.4.3.3.4	Информация, предназначена за дрона.....	19
2.4.3.4	Заемане на паркомясто.....	19
2.4.3.4.1	Генериране на цена спрямо зоната.....	19
2.4.3.5	Мениджър на изправност.....	19
2.4.3.5.1	Уведомяване при нарушение в маршрута.....	19
2.4.3.5.2	Уведомяване при метеорологични затруднения.....	20
2.4.3.6	Извършване на плащане.....	20
2.4.3.6.1	СМС.....	20
2.4.3.6.2	Дебитна карта.....	20
2.4.3.6.3	Кредитна карта.....	21
2.4.3.6.4	PayPal.....	21
2.4.3.7	Мениджър за разпознаване на свободни паркоместа.....	21
2.4.3.8	Мениджър заявки.....	21
2.5	Дронове.....	21
2.5.1	Предназначение на модула.....	22
2.5.2	Основни отговорности на модула в системата.....	22
2.5.3	Описание на интерфейсите на модула.....	22
2.5.3.1	Заснемане на изображения.....	22
2.5.3.2	Мениджър заявки.....	22
2.5.3.3	Регулатор.....	22
2.5.3.3.1	GPS устройство.....	23
2.5.3.3.2	Индикатор за установяване на неизправност.....	23

<b>3. Описание на допълнителни архитектури.....</b>	<b>23</b>
<b>3.1 Структура на внедряването.....</b>	<b>23</b>
3.1.1 Първично представяне.....	23
3.1.2 Описание на елементите и връзките.....	25
3.1.2.1 Сървър.....	25
3.1.2.2 База данни.....	25
3.1.2.3 Дронове.....	25
3.1.2.4 Мобилни устройства.....	25
3.1.2.5 Компютри.....	25
3.1.2.6 Уеб сървър.....	26
3.1.3 Описание на обкръжението.....	26
3.1.3.1 Външна система за прогноза за времето.....	26
3.1.3.2 Външна система за извършване на плащане.....	26
3.1.4 Описание на възможни вариации.....	26
<b>3.2 Структура на процесите.....</b>	<b>26</b>
3.2.1 Извършване на регистрация.....	26
3.2.1.1 Първично представяне.....	26
3.2.1.2 Описание на елементите и връзките.....	28
3.2.2 Известяване на групи при поява на неизправност.....	29
3.2.2.1 Първично представяне.....	29
3.2.2.2 Описание на елементите и връзките.....	29
3.2.3 Заемане на паркомясто.....	30
3.2.3.1 Първично представяне.....	30
3.2.3.2 Описание на елементите и връзките.....	31
3.2.3.3 Описание на обкръжението.....	33
3.2.4 Описание на възможни вариации.....	33
<b>4. Архитектурна обосновка.....</b>	<b>33</b>
4.1 Архитектурни изисквания и обосновка.....	34
4.2 Архитектурни драйвери.....	45

# 1. Въведение

## 1.1 Обща информация за текущия документ

### 1.1.1 Предназначение на документа

Основното предназначение на документа е да представи софтуерната архитектура на система за следене и управление на свободни паркоместа.

### 1.1.2 Описание на използваните структури на архитектурата

#### 1.1.2.1 Декомпозиция на модулите

Показва отделни модули на системата, които са:

- ❖ Потребителски интерфейс
- ❖ Сървър
- ❖ Дронове

**Потребителският интерфейс** предлага различен изглед и различни функционалности в зависимост от това какъв е потребителят. **Сървърът** осигурява комуникацията между системата и нейните обекти. **Дроновете** са устройства, които идентифицират свободните паркоместа и заснемат отгоре зоните за паркиране.

#### 1.1.2.2 Структура на внедряването

Показва разположението на елементите върху хардуера и комуникационното оборудване.

#### 1.1.2.3 Структура на процесите

Изобразява нагледно някои от важните процеси в системата.

### 1.1.3 Структура на документа

Документът се състои от следните главни компоненти:

1. Въведение към документа
2. Архитектурна структура „Декомпозиция на модулите“
3. Допълнителни архитектурни структури – „Структура на внедряването“ и „Структура на процесите“
4. Архитектурна обосновка, в която се изброяват аргументи за удовлетворяването на изискванията на системата и се отбелязват избраните архитектурни драйвери

## 1.2 Общи сведения за системата

**DronesPark** е система за следене и управление на свободни паркоместа, чиято цел е да улесни подхода при намиране на място за паркиране, давайки актуална и достъпна информация на интересуващите се хора. Системата поддържа следните 6 групи потребители – обикновени, регистрирани, администратори, оператори, аварийни групи и групи по контрол. Тя е достъпна през мобилно приложение, което дава възможност на своите потребители да изберат свободно паркоместото и да платят за него, а регистрираните потребители имат предимството да закупят абонамент за определен период от време. Системата поддържа архив на всички издадени фишове за глоба и заснети изображения, които са следствие от установяването на нарушители. Те са достъпни и през публичен сайт, който се зарежда чрез уеб-браузър.

## 1.3 Терминологичен речник

**Софтуер** – съвкупност от компютърни програми, които при изпълнение осигуряват желаните характеристики, функциониране и производителност

**Периферни устройства** – всички спомагателни устройства, които увеличават функционалността на компютрите, като улесняват и разширяват връзката им с околния свят

**Компютърна система** – съвкупност от компютри, софтуер и периферни устройства, които са необходими за функционирането на дадена система

**Софтуерна система** – комуникиращи си компоненти, които са част от една компютърна система

**Операционна система** – основна част от системния софтуер, която управлява и координира ресурсите на хардуера и софтуера и обслужва компютърните програми

**Браузър** – приложен софтуер, предназначен за разглеждане на информационни ресурси в Уеб

**Android, iOS** – видове мобилни операционни системи

**База данни** – колекция от логически свързани данни в конкретна предметна област, които са структурирани по определен начин

**Структура** – съвкупност от софтуерни елементи, видимите им свойства и връзките между тях

**Софтуерна архитектура** – съвкупност от структури

**Функционални изисквания** – определят какво трябва да прави една софтуерна система

**Качествени изисквания** – определят как една софтуерна система да работи

**Архитектурни драйвери** – най-важните изисквания, които са в основата за проектирането на софтуерна архитектура

**Клиент** – част от компютърна или софтуерна система, която достъпва определен вид услуга

**Сървър** – част от компютърна или софтуерна система, която предоставя услуги на други програми и може да приема заявки от клиент, след което да върне подходящ отговор

**Мобилно приложение** – компютърна програма, която е предназначена да работи на мобилни устройства

**Интерфейс** – споделена граница между два отделни компютърни компонента, които обменят информация помежду си

**Потребител** – физическо лице, което използва за свои нужди определени услуги

**Акаунт** – потребителски абонамент за компютърна мрежа или компютърна система, включващ персонален идентификатор, парола и права за достъп до определени услуги

**Модул** – логически обособена софтуерна единица

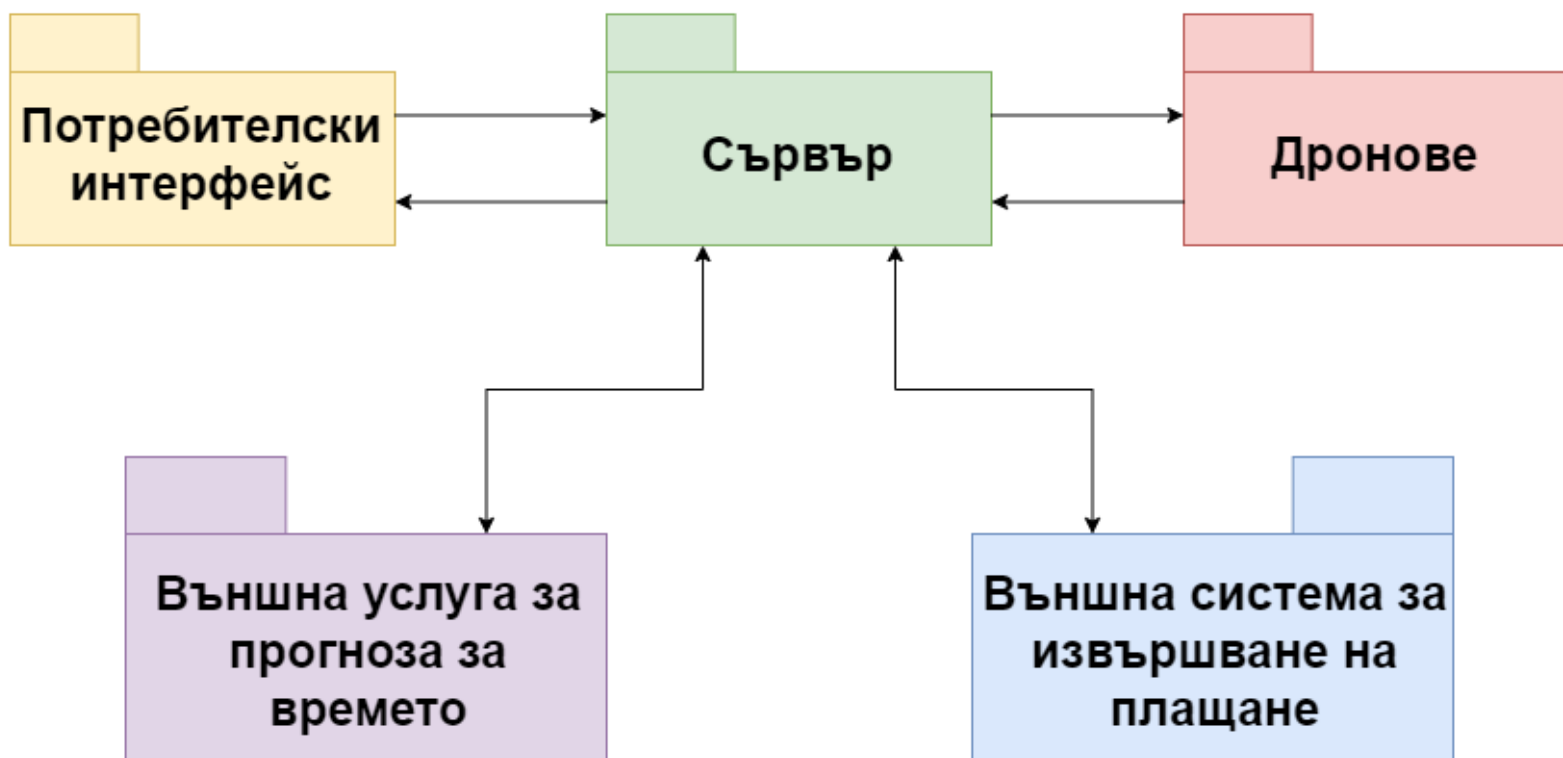
**Декомпозиция на модули** – софтуерна архитектурна структура, която показва как системата се разделя на отделни модули. Връзките между модулите са от вида „X е подмодул на Y“

**Структура на внедряването** – показва как софтуерът се разполага върху хардуера

**Структура на процесите** – показва последователност от стъпки, необходими за изпълнението на определен процес

## 2. Декомпозиция на модулите

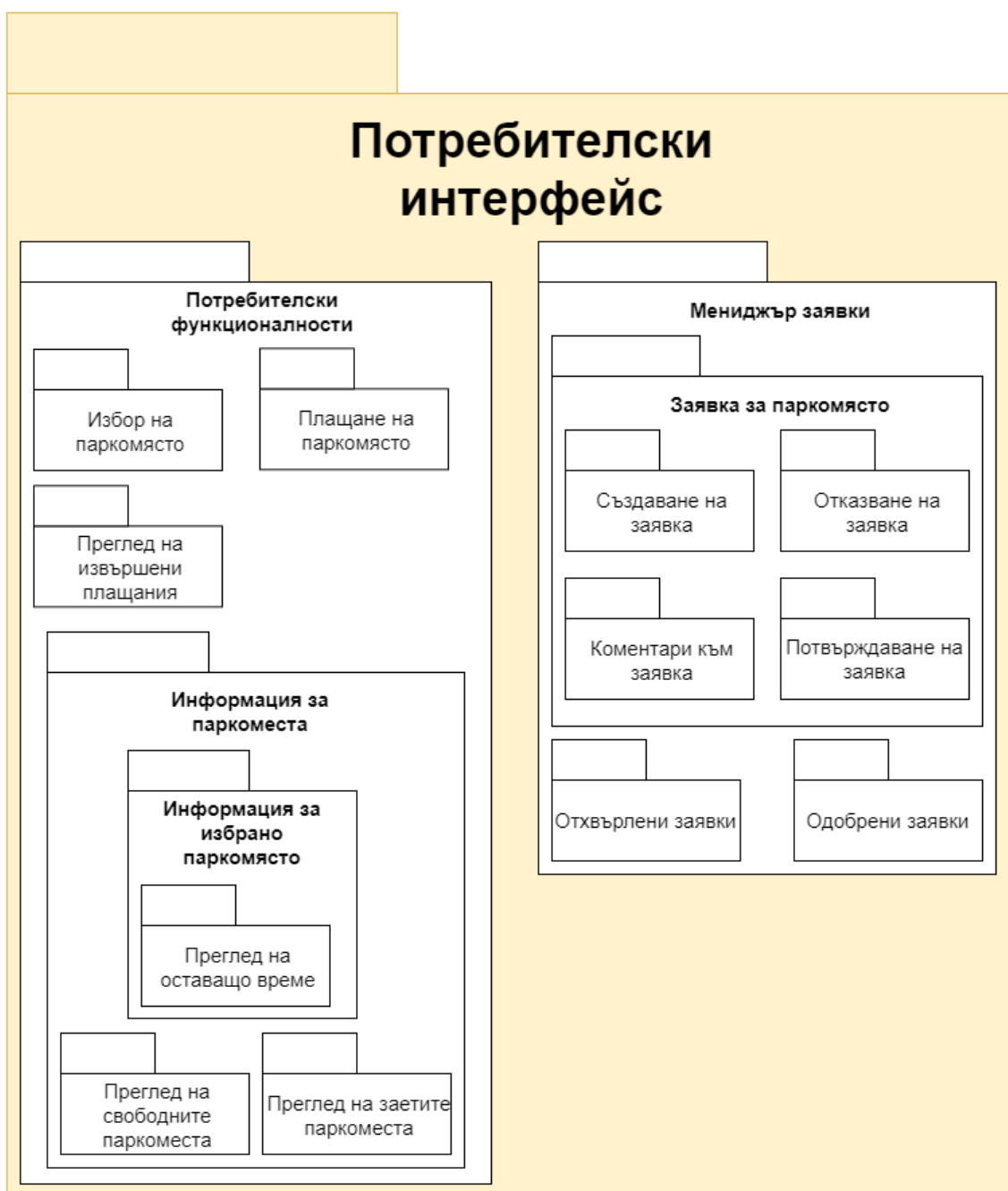
### 2.1 Общ вид на декомпозицията на модули за системата



## 2.2 Контекстна диаграма

Системата DronesPark се състои от главните модули „Потребителски интерфейс“, „Сървър“ и „Дронове“. Модулът „Сървър“ комуникира и с потребителите, и с дроновете, като за някои дейности се нуждае от външни услуги и системи. За определяне на метеорологичните условия, на базата на които се определя маршрута на дроновете, се използва външна услуга за прогноза за времето. За извършване на плащане на определено паркомясто или закупуване на абонамент от страна на регистрираните потребители се използва външна система за извършване на плащане. Ние ще разгледаме по-подробно модулите „Потребителски интерфейс“, „Сървър“ и „Дронове“, защото модулите „Външна услуга за прогноза за времето“ и „Външна система за извършване на плащане“ са част от обкръжението на нашата система и могат да бъдат ползвани от нея.

## 2.3 Потребителски интерфейс





### **2.3.1 Предназначение на модула**

Модулът „Потребителски интерфейс“ е основен модул в нашата система. Неговото предназначение е да предлага различен изглед и функционалности в зависимост от това какъв е потребителят. Той включва в себе си обикновени потребители, регистрирани потребители, администратори, оператори, групи по контрол на движението и аварийни групи. Всеки може да влезе в системата като обикновен потребител, ако няма акаунт, но може и да си направи регистрация, за да създаде такъв. Регистрираните потребители могат да закупят абонамент, да сигнализират за нарушители, а обикновените потребители не могат. Администраторите, операторите, групите по контрол на движение и аварийните групи също притежават акаунти в системата, което означава, че те също са регистрирани, но имат различни права от тези на малката група регистрирани потребители. Нашата представа е, че всички потребители, освен обикновените, са регистрирани. За да се направи разлика между регистрираните потребители и по-голямата група регистрирани потребители, можем да ги разделим по този начин:

- ❖ Обикновени нерегистрирани потребители
- ❖ Голяма група регистрирани потребители
  - Обикновени регистрирани потребители
  - Администратори
  - Оператори
  - Групи по контрол на движението
  - Аварийни групи

Идеята е, когато се направи регистрация в системата, да се определя статус на съответния потребител, за да му се предоставят необходимите и достъпни функционалности. Не би било удобно на никого от администраторите, операторите, групите по контрол на движението и аварийните групи да притежават по два акаунта – един за регистриран потребител и един за съответната група, в която е.

### **2.3.2 Основни отговорности на модула в системата**

Освен че предоставя необходимите и достъпни функционалности, модулът „Потребителски интерфейс“ отговаря и за обработването на заявки от страна на потребителите, които са свързани с оторизация, преглед на определена информация или промяна. След като бъдат обработени, заявките се изпращат в модула „Сървър“, чиято задача е да одобри или да отхвърли получените заявки и съответно да бъдат предприети необходимите действия. Заявките са свързани с идентификация на даден потребител в модула „Сървър“, преди да бъде извършено дадено изискване или предоставяне на информация.

## **2.3.3 Описание на интерфейсите на модула**

### **2.3.3.1 Потребителски функционалности**

#### **2.3.3.1.1 Избор на паркомясто**

Изборът на паркомясто се извършва на базата на предоставената от сървъра информация. Тя е достъпна както за регистрирани, така и за нерегистрирани потребители. Достъп до тази информация имат и останалите четири групи, отговарящи за функционирането на системата в ролята си на регистрирани потребители. След като е разгледана предоставената информация и има желание за заемане на дадено паркомясто, преди то да бъде отбелязано като заето, е необходимо да бъде предварително заплатено, за да се избегнат конфликтни ситуации между различните потребители – едно паркомясто се счита за заето от момента, в който то е заплатено.

#### **2.3.3.1.2 Плащане на паркомясто**

Този модул отговаря за всичко необходимо, за да се заплати избрано паркомясто. Плащането през системата за определено паркомясто може да бъде извършено от всички потребители, а закупуването на абонамент е достъпно за потребители от голяма група регистрирани потребители. Чрез модула „Мениджър заявки“ се извършва комуникацията с главния модул „Сървър“. Преди някой потребител от голямата група регистрирани потребители да заплати избрано паркомясто, или да закупи абонамент, е необходимо първо той да е влязъл в системата, след което да се извърши удостоверяване на неговия акаунт. Тези дейности се извършват в подмодул „Мениджър потребители“ на главния модул „Сървър“. Нерегистрираните потребители и тези от голяма група регистрирани потребители, които са влезли в системата и са идентифицирани, могат да извършат плащане чрез СМС, кредитна карта, дебитна карта или PayPal. Спрямо избрания начин на плащане, потребителите се прехвърлят във външната система за плащане, въвеждат необходимите данни, след което външната система уведомява дали е успешно заплащането, като необходимата информация се записва в подмодула „Конфигурация база данни“ в модула „Сървър“. Системата не поддържа информация за нерегистрираните потребители и поради тази причина част от информацията, която трябва да се попълни във външната система за плащане, може да е свързана с данни за обратна връзка, благодарение на които нерегистрираният потребител ще има възможност да разполага с информация, свързана например с отразени плащания.

### **2.3.3.1.3 Преглед на извършени плащания**

Системата предоставя на потребителите възможност да получат информация за извършените плащания до момента. Нерегистрираните потребители разполагат с такъв тип информация на място, което са посочили по време на плащането – например имейл, СМС на мобилно устройство. За да получат информация за своите отразени плащания до момента, потребителите от голяма група регистрирани потребители е необходимо да влязат в своите акаунти. Модулът „Мениджър потребители“ в модула „Сървър“ предоставя възможност за влизане в системата, както и за удостоверяване на съответния потребител. След като потребителят влезе успешно в системата, той има възможност да получи информация за отразените плащания до момента, които се осигуряват от модула „Сървър“ – извличат се от подмодула „Отразени заплащания“ в модула „Конфигурация база данни“, изпращат се на модул „Мениджър на данни“ и по-конкретно в неговия подмодул „Достъпна информация за потребители“, след което модулът „Мениджър заявки“ от главния модул „Сървър“ изпраща необходимата информация до модула „Мениджър заявки“ в главния модул „Потребителски интерфейс“ и по този начин потребителят получава информацията, която го интересува за извършените плащания до този момент.

### **2.3.3.1.4 Информация за паркоместа**

Този модул предоставя информация на потребителите за паркоместа – кои са свободни, кои са заети, както и ако потребителят е заел определено паркомясто, да може да наблюдава оставащото време. Информация за заетите паркоместа е достъпна само за потребителите от голяма група регистрирани потребители. При закупуване на абонамент избраното паркомясто се счита за заето в рамките на абонаментния период, като се съхранява информация и за номера на автомобила. По този начин потребителите имат възможност по-лесно да забележат нарушител – ако имат съмнение за определено паркомясто, върху което има паркиран автомобил, те могат да проверят кои са заетите абонаментни места, да направят проверка със съответния номер на паркирания автомобил и при наличие на нарушител могат по собствено желание да подадат сигнал.

#### **2.3.3.1.4.1 Преглед на свободните паркоместа**

Всички потребители на системата имат възможност да прегледат конкретна информация, свързана със свободните паркоместа. Информацията за свободните места се обновява на определен интервал от време в модула „Конфигурация база данни“ в главния модул „Сървър“. Необходимата информация от подмодула „Данни за паркоместа“ се извлича и се предоставя на модула „Мениджър на данни“ и по-конкретно – в неговия

подмодул „Достъпна информация за потребители“, след което се изпраща достъпната информация на модула „Мениджър заявки“ в главния модул „Потребителски интерфейс“. По този начин потребителите разполагат с необходимата информация за свободните паркоместа.

#### **2.3.3.1.4.2 Преглед на заетите паркоместа**

Аналогично е на опцията „Преглед на свободните паркоместа“, като разликата е, че нерегистрираните потребители нямат достъп до тази функционалност. Отново се извлича информация от модула „Конфигурация база данни“ от главния модул „Сървър“, но този път тя е свързана не със свободните паркоместа, а със заетите.

#### **2.3.3.1.4.3 Информация за избрано паркомясто**

След като даден потребител е изпълнил всички необходими условия за заемане на паркомясто, той има възможност да получава информация за него.

##### **2.3.3.1.4.3.1 Преглед на оставащо време**

Потребителите имат възможност да прегледат информация за заетото от тях паркомясто – например колко време е изминало от извършеното плащане, колко време остава до изтичане на периода за заетото паркомясто или колко време остава до изтичане на закупен абонамент при потребители от голяма група регистрирани потребители.

### **2.3.3.2 Мениджър заявки**

Този модул има особено важно значение за системата. Той осигурява комуникацията с главния модул „Сървър“. През него преминават абсолютно всички потребителски функционалности, превърнати от съответните модули в заявки, които се изпращат към главния модул „Сървър“. Освен че изпраща, също така той получава заявки от сървъра, които обработва и благодарение на които доставя необходимата информация на потребителите. Представили сме примерен подмодул „Заявка за паркомясто“, която описва събитията, свързани с оформяне на заявка за определено паркомясто. Разбира се, системата е доста голяма и могат да бъдат добавени още повече функционалности, както и заявки за всяка една отделна функционалност.

#### **2.3.3.2.1 Одобрени заявки**

Този модул отговаря за разпознаването на заявки. Когато дадена заявка е успешна/одобрена, то или се показва тази информация на заинтересования подател, или изпраща сигнал към друг модул, че всичко до момента е

успешно и може да се премине нататък към изпълнението на други заявки или друга част на заявката.

#### **2.3.3.2.2 Отхвърлени заявки**

Всички неодобрили/отхвърлени заявки се изпращат в този модул. Той, от своя страна, или прекъсва основната заявка, която е довела до това събитие, и се уведомява нейният подател, или в зависимост от заявката, се прескача, като се запазва за бъдещи проблеми.

#### **2.3.3.2.3 Заявка за паркомясто**

Потребителите могат да прегледат информация за свободните паркоместа и да изберат кое да заемат. След направен избор те трябва да създадат заявка, в която да попълнят необходимите данни, имат възможност да потвърдят или да откажат създадената от тях заявка, както и да добавят коментари към нея.

##### **2.3.3.2.3.1 Създаване на заявка**

Създаването на заявка включва въвеждане на информация, в която потребителите да посочат избрано паркомясто, време, за което ще го заемат, номер на автомобил, начин на плащане и други.

##### **2.3.3.2.3.2 Отказване на заявка**

Възможно е потребителят да е избрал грешно паркомясто, да е въвел грешни данни или просто да се е отказал от заемане на паркомясто. Поради тези причини съществува опция за премахване на сегашната заявка, чиято цел е да изтрие попълнената информация и да се даде възможност за попълване на нова заявка.

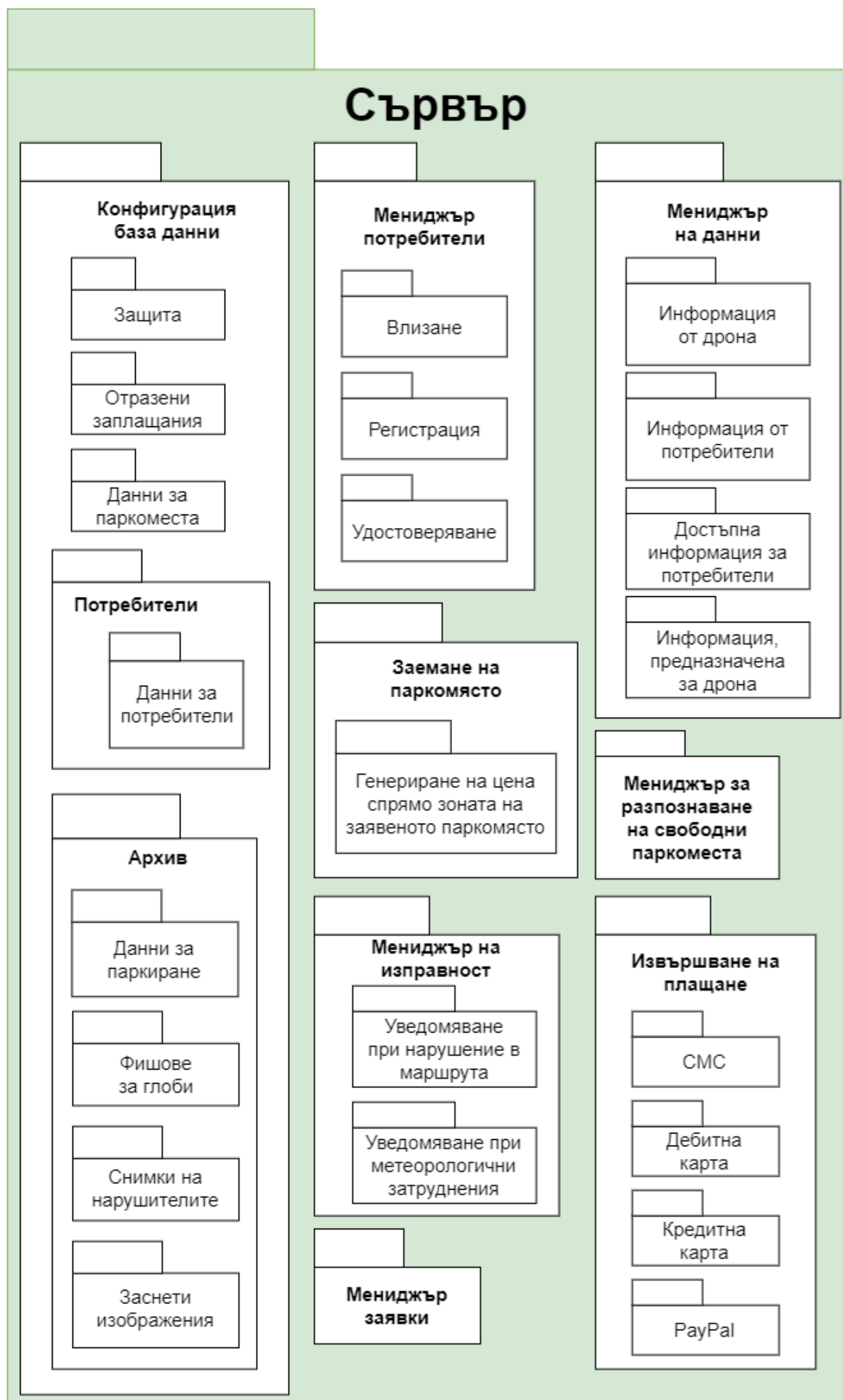
##### **2.3.3.2.3.3 Коментари към заявка**

Когато заявката е готова, но потребителят или сървърът имат някакви допълнителни изисквания към самата заявка, то те се добавят като коментари и се обработват допълнително от съответния модул.

##### **2.3.3.2.3.4 Потвърждаване на заявка**

След като дадена заявка е създадена и попълнена от потребител, преди да се изпрати към съответните модули за изпълнение, е необходимо потвърждение от подателя, че е сигурен и желае дадената заявка да бъде изпратена. Веднъж бъде ли потвърдена дадената заявка, тя се изпраща за изпълнение и е невъзможно да бъде отказана.

## 2.4 Сървър



## **2.4.1 Предназначение на модула**

Сървърът също е основен модул в нашата система. Неговото предназначение е да осигурява комуникацията между всички останали модули в системата. Чрез него се осъществява и връзката с външните системи за плащане и прогноза за времето.

## **2.4.2 Основни отговорности на модула в системата**

Този модул изцяло отговаря за стриктното изпълнение на бизнес логиката на нашата система. В него се осъществява защита, оторизация, обработване на снимки, разпознаване на свободни паркоместа, съхраняване и доставяне на информация.

## **2.4.3 Описание на интерфейсите на модула**

### **2.4.3.1 Конфигурация база данни**

Можем да кажем, че този модул отговаря за съхранението на цялата информация на системата, използвана някога от нея. Понеже се съхранява изключително важна информация, то е необходимо да се осигури максимална защита от външен достъп.

#### **2.4.3.1.1 Защита**

Тъй като системата разполага с уязвима информация, която обработва, не можем да не осигурим защита. Независимо дали е малка и едвам забележима система, или голяма и известна, необходимо е тя да бъде добре защитена от външна намеса. Самото наличие на някаква лична информация е магнит за атаки. Този модул отговаря за предотвратяване на тези атаки.

Предотвратяването се осъществява чрез ограничаване на достъпа до модул „Конфигурация база данни“. Тези ограничения са свързани с достъпа, но точно, че той е ограничен. Когато дадена уязвима информация трябва да се впише, тя може да се криптира (да се замаскира оригиналната информация и да се замени с друга). Ако някоя информация трябва да се прочете, но е криптирана, то тя да може да се декриптира. Това може да се използва и за регистрираните потребители – когато те извършват своята регистрация, те добавят парола, която трябва да се запази в криптиран вид. За да може да получим желана от нас информация, е необходимо да бъдат изпълнени определени изисквания, които зависят от подмодула, от който искаме информация. Ако не е изпълнено дори едно от поставените условия, достъпът до търсената информация ще бъде предотвратен/отказан и потребителят ще получи известие за това чрез модула „Отхвърлени заявки“ от главния модул „Потребителски интерфейс“.

#### **2.4.3.1.2 Отразени заплащания**

След като успешно е извършено плащане посредством външната система за плащане, тя изпраща известие. Когато известието е за неуспешно плащане, се изпраща заявка към „Отхвърлени заявки“, чрез която е необходимо да се уведоми подателят. Ако пък плащането е успешно, то известието, получено от външната система, съдържа важна информация, свързана с потребителя, паркомясто и период на валидност. Всичко това се записва в този модул.

#### **2.4.3.1.3 Данни за паркоместа**

Съхранява информация за абсолютно всички налични паркоместа в града, както и техните райони, цени и номера на автомобили, ако има заплатен абонамент или мястото е запазено за определен период от днешния ден. От модула „Информация от дрона“ се получават снимки, които се обработват от модула „Мениджър за разпознаване на свободни паркоместа“. Ако някое паркомясто се заеме, то тогава се променя неговият статус и обратно – ако някое паркомясто се освободи, отново се променя неговият статус.

Операторите на системата обновяват през определен интервал от време информацията за свободните и заетите места. Отразяваме също кои са и заетите места, за да може групите по контрол на движението да имат възможност да следят кои са заетите места и евентуално да проверяват за нарушители. Информацията се записва и в модул „Архив“, за да може впоследствие да се използва за статистика.

#### **2.4.3.1.4 Потребители**

Този модул отговаря за записването на цялата информация за потребителите на системата от голяма група регистрирани потребители.

##### **2.4.3.1.4.1 Данни за потребители**

Системата ни се нуждае от съхранена и подредена информация за потребителите, която се получава непосредствено след тяхната регистрация – например като име, телефонен номер, имейл, парола, номер на дебитна или кредитна карта в случай, че е извършено плащане чрез нея.

Регистрираните потребители имат право да заплащат абонамент за определено паркомясто, което се маркира като заето в рамките на определения срок на абонамента, независимо дали заснетите от дронове изображения показват наличието на автомобил на съответното паркомясто. Ако даден регистриран потребител е заплатил абонамент, то тогава ще се пази допълнителна информация за съответния регистриран потребител като дата на заплатения абонамент, заявеното място, номер на автомобил. Ако имаме регистрирани потребители, които не са заплатили абонамент, то тогава като информация в графата за абонамент ще има например „false“, а в



графата за номер на автомобил ще пише например „NULL“. В модула „Данни за потребители“ в „Конфигурация база данни“ освен информация за имена, номера на автомобили, имейли, пароли и абонаменти, както и периоди на абонаменти, се съхранява и информация за статуса на съответния регистриран потребител. Този статус ни е необходим, защото спрямо него се определят функционалностите и изгледът, които могат да виждат отделните групи потребители, които не са обикновени. След регистриране в системата всеки потребител получава статус „регистриран потребител“, който се отбелязва в модула „Конфигурация база данни“. Ако някой от регистрираните потребители трябва да спада към групата на администраторите, операторите, аварийните групи или групите по контрол на движение, то тогава техният статус се променя от най-главния човек в системата, който има достъп до абсолютно всичко в системата и поема цялата отговорност за своите действия.

#### **2.4.3.1.5 Архив**

В модула „Конфигурация база данни“ присъства още един важен модул – „Архив“. В него се съхраняват данни за динамиката на паркирането, всички издадени фишове за глоби, както и заснетите изображения.

##### **2.4.3.1.5.1 Данни за паркиране**

В модула се записва обработената информация от получените изображения. Той включва като информация динамиката на паркиране в съответния ден и час от седмицата, което помага да се определи броят на летящите дронове, както и маршрутът на всеки от тях.

##### **2.4.3.1.5.2 Фишове за глоби**

Модулът съхранява информация за издадените до този момент фишове на съответните нарушители. Издаването на фиш се извършва, когато е засечен нарушител. Групите по контрол на движение отиват на съответното място и в зависимост от нарушението се издава фиш на собственика, като е възможно колата да бъде принудително преместена.

##### **2.4.3.1.5.3 Снимки на нарушителите**

Модулът съхранява информация за получените до този момент снимки на нарушители. При засичане на нарушител, освен принудителното преместване на автомобила, групите по контрол на движение трябва да заснемат настъпилото събитие, като снимката се съхранява директно в системата, и след това се издава електронен фиш за глоба. Избрахме този модул да бъде подмодул на „Архив“, защото заедно с издадените фишове за глоба да има възможност за съхранение и на снимки на съответните

нарушители, като по този начин системата ни да разполага с цялостна информация за настъпилото събитие.

#### **2.4.3.1.5.4 Заснети изображения**

Модулът съхранява информация за заснетите до този момент изображения от дроните. Спрямо тях може да се създаде статистика относно качеството на работа на дроните.

### **2.4.3.2 Мениджър потребители**

Този модул отговаря за правилното менажиране на различните потребители чрез изпълняване на определени условия. Той е изграден от подмодулите „Влизане“, „Регистрация“ и „Удостоверяване“.

#### **2.4.3.2.1 Влизане**

Влизането на потребител в системата се случва, когато се проверят въведените от него данни, направи се негова идентификация и след това той се вписва със своя статус. За тази цел се зарежда формата за влизане, където той трябва да въведе своя имейл и парола. Този модул изпраща заявка до модула „Удостоверяване“, а той – до модула „Конфигурация база данни“. Ако премине защитата, се проверява дали има записан такъв имейл. Ако не е намерен такъв имейл, заявката се отбелязва като „Отхвърлена“ и потребителят е уведомен за това. Ако има такъв имейл, се прави проверка дали въведената парола съответства на запазената в базата за съответния имейл. Ако всичко е успешно, заявката се отбелязва като „Одобрена“ и потребителят се вписва в системата със своя статус. Ако е възникнал някакъв друг проблем, заявката се отбелязва като „Отхвърлена“ и процесът по влизане започва отначало.

#### **2.4.3.2.2 Регистрация**

Когато някой обикновен потребител желае да се регистрира в системата, той трябва да попълни заредената от системата форма. След като попълни всички изисквания – например имейл, парола, име, фамилия, телефонен номер (могат да се добавят и други полета), модулът проверява дали данните са валидни. Валидирането е различно в зависимост от полето, което е попълнено. За полето имейл се правят 2 основни проверки. Например дали имейлът съдържа символите „@“ и „.“ веднъж. Ако тази проверка е успешна, се преминава към втората – дали има такъв имейл, записан в базата данни. Ако съществува, то тогава не може да се създаде акаунт със същия имейл. Ако някое от изискванията е нарушено, то заявката се отбелязва като „Отхвърлена“ и се уведомява потребителят за това. Ако всичко е успешно, се преминава към удостоверяване на паролата. Това включва например дали

съдържа определен брой символи, специални символи, малки и главни латински букви, както и цифри. Ако не отговаря на изискванията, заявката се отбелязва като „Отхвърлена“. Ако всичко е наред, се извършва валидация за името и фамилията, както и дали телефонният номер на потребителя съдържа само цифри, дали започва с 0 и дали броят на цифрите е 10. Ако всичко е успешно, регистрацията е завършена. Ако пък не, аналогично при грешка, заявката се отбелязва като „Отхвърлена“ и процесът по регистрация започва отначало.

#### **2.4.3.2.3 Удостоверяване**

Модулът „Удостоверяване“ осъществява пълно идентифициране на потребителя, опитващ се да влезе в системата.

#### **2.4.3.3 Мениджър на данни**

Този модул отговаря за правилното обработване на данни от всички компоненти на системата чрез подходящи и правилни заявки.

##### **2.4.3.3.1 Информация от дрона**

Задачата на дроновете е да заснемат изображения. Направените от тях изображения се изпращат в сървъра на системата в модул „Мениджър на данни“ и по-конкретно – в подмодула „Информация от дрона“. Получената от дроновете информация се изпраща в модула „Конфигурация база данни“, за да може да се отбележи информация за динамиката на паркиране в съответния ден и час от седмицата, което ще помогне да се определи в бъдеще броят на летящите дронове, както и маршрутът на всеки от тях. Изображенията се изпращат и на модула „Мениджър за разпознаване на свободни паркоместа“, за да се идентифицират кои са свободните и заетите места, като отново се предоставят на модула „Конфигурация база данни“, за да може да се обнови информацията до този момент и да се информират по правилен начин потребителите на системата за наличието на свободни и заети паркоместа.

##### **2.4.3.3.2 Информация от потребители**

Възможно е да се получава информация не само от дроновете, а и от потребителите – когато някой регистриран потребител забележи нарушител, той може да сигнализира на групите по контрол на движение. Подмодулът „Информация от потребители“ в модула „Мениджър на данни“ получава изпратената информация, преработва я и я изпраща на групите по контрол на движение, правейки справка в модула „Конфигурация бази данни“ кои са групите по контрол на движение чрез съответния им статус.

#### **2.4.3.3.3 Достъпна информация за потребители**

Този модул осигурява допълнителна защита на данните в системата. Това се осигурява чрез нова проверка дали даден потребител има достъп до определена информация. Ако всичко е наред, информацията се предоставя на потребителите и заявката се отбелязва като „Одобрена“, а в противен случай – като „Отхвърлена“, като за това се уведомява подателят.

#### **2.4.3.3.4 Информация, предназначена за дрона**

Този модул съдържа информация например за промяна на маршрута в даден момент на определен дрон, както и информация за бройката на дроновете в определен момент от деня, седмицата. Така се ограничава възможността за външна намеса върху управлението на дронове и се увеличава защитата на системата.

#### **2.4.3.4 Заемане на паркомясто**

В сървъра имаме и друг модул – „Заемане на паркомясто“. Преди да се извърши заплащане на съответното паркомясто, е необходимо първо да е известно каква сума би трябвало да се плати за него. Определянето на цената за определено паркомясто се извършва в подмодула „Генериране на цена спрямо зоната“.

##### **2.4.3.4.1 Генериране на цена спрямо зоната**

Избраното паркомясто може да е безплатно или да се таксува. Цената за съответното паркомясто зависи от зоната, за която е подадено запитване. Информация за цена на заявеното паркомясто се получава от модула „Конфигурация база данни“ и по-конкретно – от неговия подмодул „Данни за паркоместа“. След като е ясна цената на заявеното паркомясто, то тогава вече може да се пристъпи към заплащане.

#### **2.4.3.5 Мениджър на изправност**

Целта на модула е да следи за неизправности, свързани с дроновете, и при наличието на такива да уведоми съответните групи, отговарящи за конкретния проблем. Тези проблеми може да не се срещат често, но дори една тяхна поява може да доведе до срыв в системата, ако не бъдат уведомени съответните групи, които се грижат за тяхното отстраняване. Именно поради тази причина ние имаме нужда от модула „Мениджър на изправност“, който да уведоми съответните групи при нарушение в маршрутите на дроновете и при затруднения, свързани с метеороголичните условия.

##### **2.4.3.5.1 Уведомяване при нарушение в маршрута**

Броят на дроновете и техните маршрути се определят на базата на натрупаните данни за динамиката на паркиране, които се съхраняват в модула „Конфигурация база данни“. Когато някой дрон излезе от строя, то трябва да

се уведомят аварийните групи, за да могат те да отидат в района на съответния дрон и да отстранят повредата. Поради тази причина се нуждаем от постоянно наблюдение върху движенията на дроновете – дори едно малко отклонение от маршрута трябва да бъде сигнализирано.

#### **2.4.3.5.2 Уведомяване при метеорологични затруднения**

Ако заснемането на паркоместата е затруднено поради метеорологични условия, то тогава трябва да се уведомят операторите на системата. Излишно е да се заснемат изображения при намалена видимост. Това няма да е ефективно за потребителите, защото няма да получат необходимата им информация относно паркоместата.

#### **2.4.3.6 Извършване на плащане**

След като потребител е избрал определено паркомясто или регистриран потребител желае да закупи абонамент, е необходимо той да заплати. Това се извършва посредством външна система за извършване на плащане, като тя предоставя четири начина – СМС, кредитна карта, дебитна карта или PayPal, като в бъдеще могат да бъдат добавени още възможности. Когато някой потребител иска да извърши плащане, той избира начин, като съответно се свързва с подмодула „Информация от потребители“ на модула „Мениджър на данни“, като подадената информация от потребителите се обработва и съответно се записва в модула „Конфигурация база данни“.

##### **2.4.3.6.1 СМС**

При избор на тази опция за плащане потребителите получават телефонен номер и текст. Те трябва да изпратят получения текст на получения номер, като автоматично ще бъдат таксувани на цената за избраното паркомясто или на цената за закупения абонамент – втората опция не е достъпна за нерегистрираните потребители.

##### **2.4.3.6.2 Дебитна карта**

При избор на тази опция за плащане се зарежда форма, която потребителите трябва да попълнят с определени данни като например номер на карта, имена, телефон за връзка и други. При възникване на проблем системата уведомява потребителя и му се дава възможност да въведе отново своите данни, а ако операцията е успешна, потребителят се таксува за избраното паркомясто или за закупения абонамент (втората опция не е достъпна за нерегистрираните потребители) и необходимите данни се изпращат в главния модул „Сървър“, като се съхраняват в неговия подмодул „Конфигурация база данни“.

#### 2.4.3.6.3 Кредитна карта

Процесът при избор на тази опция за плащане е аналогичен на процеса при извършване на плащане чрез дебитна карта.

#### 2.4.3.6.4 PayPal

При избор на тази опция за плащане потребителите се пренасочват към външната система за плащане PayPal.

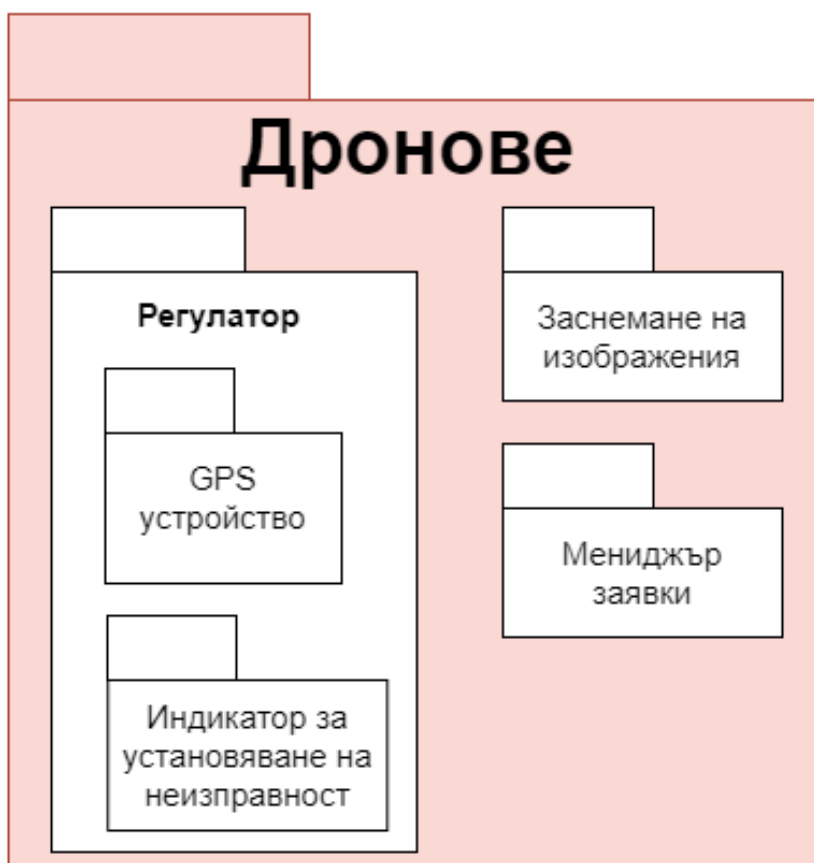
#### 2.4.3.7 Мениджър за разпознаване на свободни паркоместа

Модулът използва специфичен алгоритъм за разпознаване на свободни паркоместа на база на получените от дрона изображения. Разграничават се свободните от заетите паркоместа и необходимата информация се обновява в модула „Конфигурация база данни“. Отделните паркоместа се идентифицират с уникален номер.

#### 2.4.3.8 Мениджър заявки

Модулът отговаря за обработката на всички получени заявки в главния модул „Сървър“, като той може да ги одобри за изпълнение или да ги отхвърли. През този модул минават и всички заявки, които трябва да бъдат изпратени към останалите компоненти на системата, като се проверяват и верифицират при необходимост.

## 2.5 Дроне



### **2.5.1 Предназначение на модула**

Това е друг основен модул на нашата система. Благодарение на него се извършва заснемането на паркоместа по определен маршрут, като направените изображения се съхраняват в архив за определен срок.

### **2.5.2 Основни отговорности на модула в системата**

Дроновете имат две основни задачи, свързани със заснемане и следене на маршрут. Изключително важно е заснетите изображения да са ясни. При проблем с това изискване е необходимо да се уведомят операторите на системата. Също така всеки дрон следва генериран маршрут и отклоняването от него е сигурен знак за неизправност. При тази ситуация е необходимо да се уведомят аварийните групи на системата.

### **2.5.3 Описание на интерфейсите на модула**

#### **2.5.3.1 Заснемане на изображения**

Този модул отговаря за правилното фокусиране и заснемане на изображения. Изключително важно е снимките да бъдат ясни и снимани по начина, който е зададен.

#### **2.5.3.2 Мениджър заявки**

Този модул отговаря за всички пристигащи към дрона заявки, както и за изпращането на такива. Чрез обработването на заявки те могат бързо да бъдат пренасочени към конкретните модули за изпълнение. Заснетите изображения се изпращат в подмодула „Информация от дрона“ в модула „Мениджър на данни“, който се намира в главния модул „Сървър“. След това те се прехвърлят на модула „Мениджър за разпознаване на свободните паркоместа“, като се разграничават свободните от заетите места. Необходимата информация се запазва в модула „Конфигурация база данни“. Подмодулът „Информация, предназначена за дрона“ в модула „Мениджър на данни“ съдържа информация като например нов генериран маршрут, който трябва да бъде предоставен на определен дрон. Изпращането на такъв вид информация се осъществява именно чрез комуникацията на модулите „Мениджър заявки“ в главните модули „Сървър“ и „Дронове“.

#### **2.5.3.3 Регулатор**

Основната цел на регулатора е да следи за неизправности в конкретни ситуации и при наличието на такива веднага да подаде сигнал за уведомяване на необходимите групи.

#### **2.5.3.3.1 GPS устройство**

Дроновете имат вградено GPS устройство, чрез което могат да бъдат проследявани, и ако се получи отклоняване от определения им маршрут, да могат да бъдат уведомявани аварийните групи за района, в който се намират дроновете.

#### **2.5.3.3.2 Индикатор за установяване на неизправност**

Този модул отговаря за следене на общи неизправности от вида на проблем/невъзможност за фокус, възникнал технически проблем по дрона, невъзможност за заснемане на изображение поради наличие на мъгла например или други метеорологични условия и други. В такива ситуации се изпраща заявка до модула „Мениджър на заявки“ в главния модул „Дронове“, след това се прехвърля в модула „Мениджър на заявки“ в главния модул „Сървър“, след което модулът „Мениджър на изправност“ уведомява аварийните групи или операторите на системата в зависимост от вида на неизправността.

## **3. Описание на допълнителни архитектури**

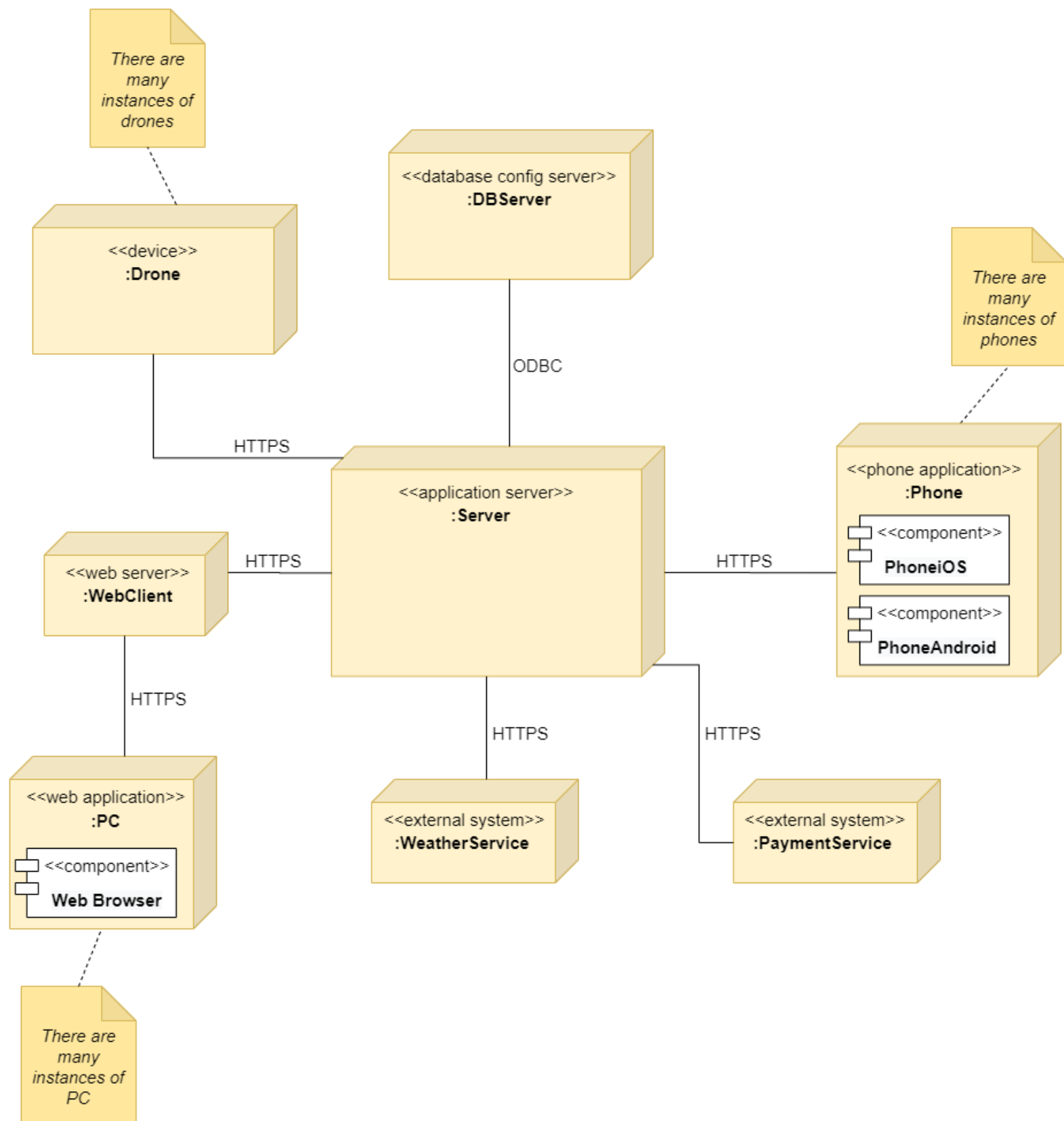
### **3.1 Структура на внедряването**

Избрахме една от допълнителните архитектурни структури да е структура на внедряването, защото за нас е важно да покажем своята представа за разположението на елементите върху хардуера. Елементите са процеси, хардуерни устройства и комуникационни канали.

#### **3.1.1 Първично представяне**



# Структура на внедряването



## **3.1.2 Описание на елементите и връзките**

### **3.1.2.1 Сървър (Server)**

Server е софтуерът, който изпълнява бизнес логиката на приложението. Той трябва да бъде на собствена машина. Необходимо е да бъде постоянно под наблюдение – да се следи поведението му при наличие на повече заявки, да се следи комуникацията му с потребителите и дроновете, да се наблюдава скоростта му при обработване на заявки, както и при обмен на данни с базата данни.

### **3.1.2.2 База данни**

Базата от данни е логически отделна изчислителна единица и като такава тябва бъде отделена на машина, различна от тази, която се занимава с бизнес логиката. Базата данни си комуникира със сървъра чрез ODBC.

### **3.1.2.3 Дронове (Drones)**

Дроновете са устройства, които са главна част от системата. Те си комуникират постоянно с главния сървър чрез HTTPS протокол. Тяхната задача е да заснемат изображения, след което да ги изпратят в базата данни чрез сървъра, за да могат да се запазят в архива и да се обнови информацията, свързана с паркоместата – кои са свободни и кои са заети. Дроновете разполагат с GPS устройства и когато се появи отклонение от техния маршрут, е необходимо да се сигнализируют аварийните групи. Те имат и индикатор за установяване на неизправност, който служи да се установи дали има метеорологични затруднения, които да попречат на заснемането на изображения, в следствие на което е необходимо да бъдат уведомени операторите на системата.

### **3.1.2.4 Мобилни устройства (Phones)**

Phone е мобилно устройство, което има възможност да изтегли мобилното приложение към системата. Този софтуер се инсталира от потребителя върху мобилно устройство. Двата компонента са мобилните приложения за Android и iOS. Трябва да се отбележи, че те използват един и същ програмен код. Мобилните приложения си взаимодействат със сървъра чрез HTTPS.

### **3.1.2.5 Компютри (PC)**

PC е компютър (устройство), което притежава приложен софтуер, наречен уеб браузър (web browser), с помощта на който се предоставя възможност за разглеждане на информационни ресурси в Уеб. Уеб браузърът може да достъпи уеб потребителския интерфейс към системата, който е разположен на уеб сървър. Връзката се осъществява чрез HTTPS протокол.

### **3.1.2.6 Уеб сървър (WebServer)**

WebServer съдържа приложението, осигуряващо уеб потребителски интерфейс. Той ни е необходим заради изискването снимките и фишовете на нарушителите да са достъпни и през публичен сайт, който се зарежда с уеб браузър. WebServer получава заявки по HTTPS от WebBrowser и комуникира пряко с главния сървър.

## **3.1.3 Описание на обкръжението**

### **3.1.3.1 Външна система за прогноза за времето (WeatherService)**

WeatherService е външна система, която определя прогнозата за времето, като по този начин се оказва влияние върху броя и маршрута на дроновете.

### **3.1.3.2 Външна система за извършване на плащане (PaymentService)**

PaymentService е външна система, с помощта на която се извършва плащането при закупуване на абонамент или при заемане на свободно паркомясто.

## **3.1.4 Описание на възможни вариации**

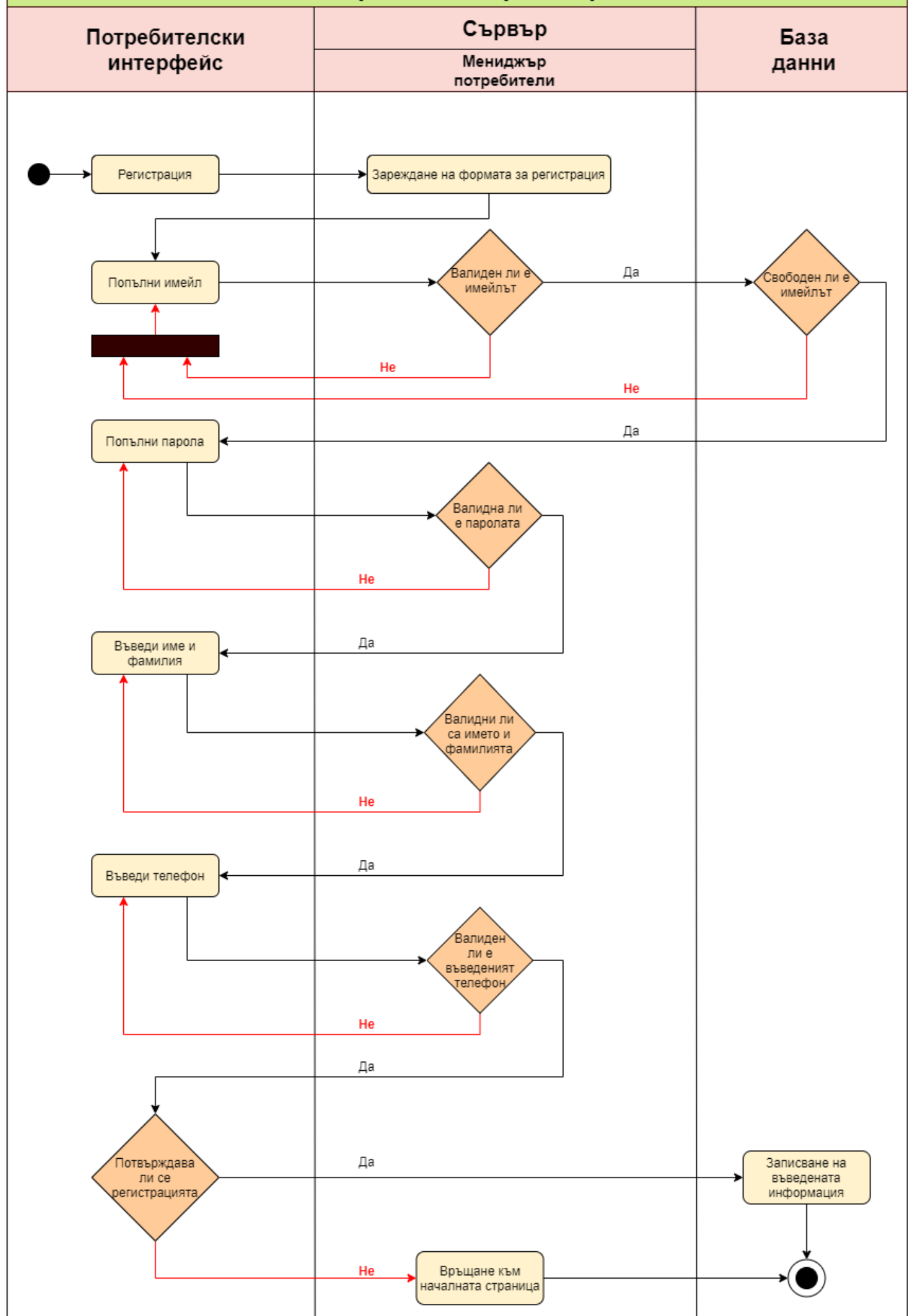
Възможно е да се появят моменти, в които сървърът да е натоварен с множество заявки от страна на потребителите. В такива случаи е удобно да се измисли вариант, който да измести главния фокус от сървъра. Възможно е да бъде добавен допълнителен компонент, който да получава заявките от потребителите, да ги обработва и да изпраща към сървъра само одобрените. Освен външна система за прогноза за времето и външна система за извършване на плащане, от полза би била и външна система, която да навигира потребителите до избраното от тях паркомясто, както и да показва пътищата, по които има трафик. Такава подходяща външна система е Google Maps.

## **3.2 Структура на процесите**

### **3.2.1 Извършване на регистрация**

#### **3.2.1.1 Първично представяне**

## Извършване на регистрация



### 3.2.1.2 Описание на елементите и връзките

Диаграмата показва последователността от процеси, които се извършват при регистриране на потребител. За да се регистрира, потребителят трябва да избере опция „Регистрация“ в мобилното приложение. Мениджър потребители в сървъра зарежда формата, която трябва потребителят да попълни. В нея има полета за въвеждане на имейл, парола, име и фамилия, както и телефон за контакт. След като потребителят въведе имейл, мениджърът потребители проверява дали имейлът е валиден – например за валиден имейл може да се приеме въведеният низ да съдържа поне един символ, след което да съдържа символа „@“, след това да има поне още един символ, след което да съдържа символа „.“ и накрая да завършва с поне един символ (разбира се, валидацията за имейл може да съдържа много други критерии). Ако въведеният имейл е валиден, то тогава трябва да се направи проверка дали той вече не е зает от друг регистриран потребител – прави се справка в базата данни. Ако въведеният имейл не е нито валиден, нито свободен, то тогава се извежда грешка на потребителя и не му се дава възможност да продължи с попълването на регистрационната форма. Ако въведеният имейл е едновременно валиден и свободен, то тогава се дава възможност на потребителя да попълни полето за парола, като отново се прави проверка в мениджър потребители в сървъра дали тя изпълнява определени критерии – например да съдържа минимум 8 символа, които да включват поне една малка латинска буква и цифри от 0 до 9 (валидацията за парола може да съдържа и други критерии). Ако въведената парола не изпълнява определените изисквания, то тогава потребителят не може да завърши своята регистрация. В противен случай се преминава към полето за въвеждане на име и фамилия, като отново се прави проверка в мениджър потребители в сървъра дали са въведени валидни имена, като тук критерият може да бъде да се съдържа интервал във въведения низ. Ако въведените име и фамилия не отговарят на изискванията, то тогава потребителят трябва да удовлетвори наложените критерии за това поле, преди да продължи нататък. В противен случай, ако въведените име и фамилия са валидни, то тогава потребителят може да премине към поле за въвеждане на телефонен номер за контакт. Тук отново се извършва проверка дали въведеният телефон е валиден – например дали започва с „0“, дали съдържа 10 цифри. Ако въведеният телефонен номер не е валиден, то тогава не се дава възможност за завършване на регистрацията. Ако всички полета изпълняват въведените ограничения, потребителят може да завърши своята регистрация или да се откаже. Ако не иска да завърши своята регистрация, то тогава се връща на началната страница и приключва с регистрацията, а ако желае да я завърши, то тогава въведената дотук информация се запазва в базата данни и регистрацията може да се счита за завършена.

## 3.2.2 Известяване на групи при поява на неизправност

### 3.2.2.1 Първично представяне



### 3.2.2.2 Описание на елементите и връзките

Диаграмата представя процеса на известяване на групи при поява на неизправност. Докато следват определения си маршрут, дроновете заснемат паркоместа. Всяко заснето изображение се изпраща в сървъра. В подмодула „Информация от дрона“ на модул „Мениджър на данни“ се проверява дали е получено заснетото от дрона изображение.

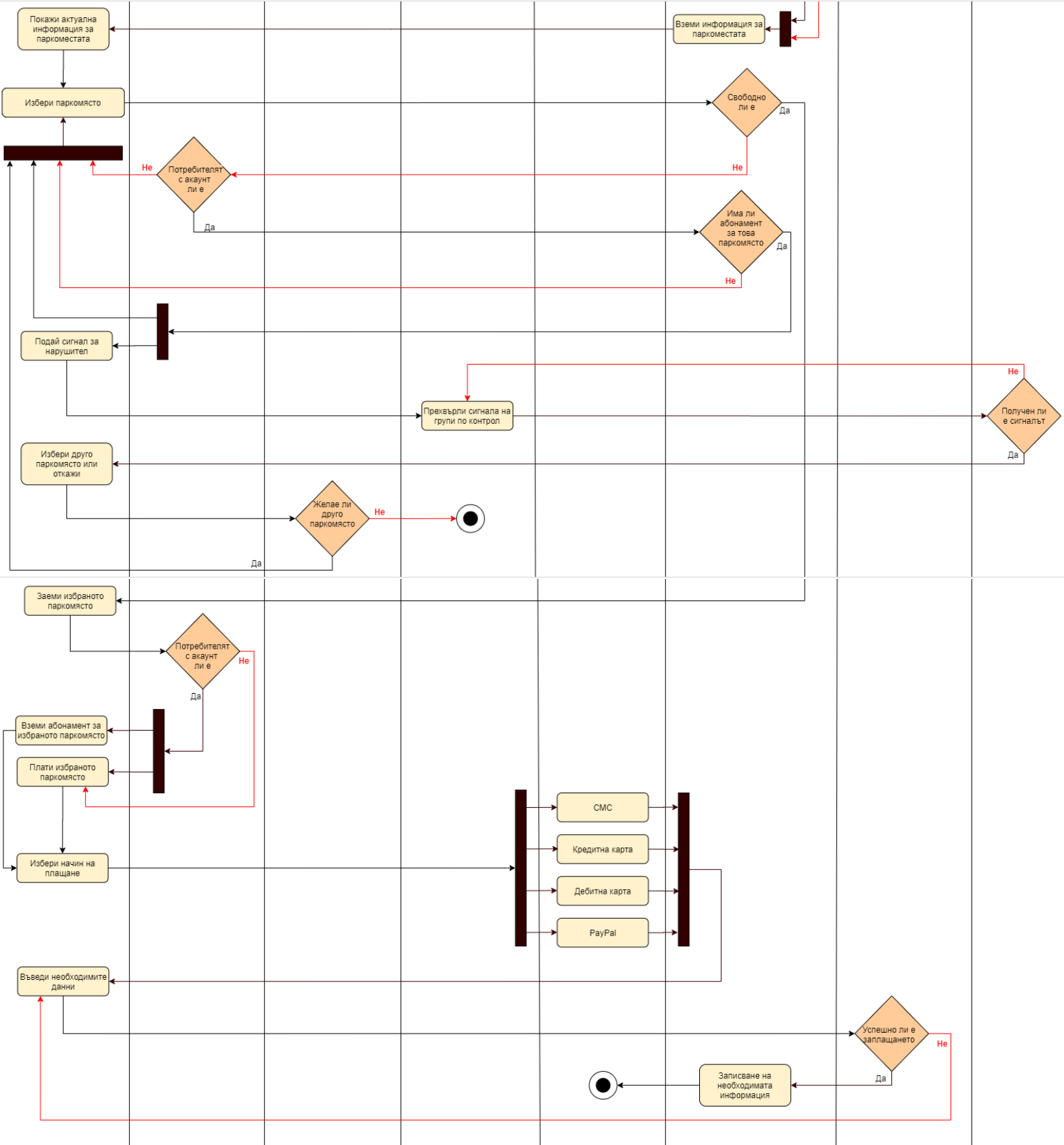
- Ако **не е получено**, то тогава мениджърът на изправност трябва да прехвърли сигнала на група оператори, защото причината най-вероятно е свързана с метеорологичните условия. Трябва да се провери дали групата от оператори е получила сигнала – ако не е получен, то тогава мениджърът на изправност отново трябва да направи опит да изпрати сигнал, но ако няколко пъти тази дейност е неуспешна, то тогава вероятно има проблем в системата. Ако група оператори получи сигнала за неизправност, то тогава процесът приключва, защото операторите трябва да поемат отговорността за отстраняване на появилата се неизправност.
- Ако **изображението е получено**, то тогава мениджърът за разпознаване на свободни места използва готов алгоритъм, с помощта на който се разграничават свободните места от заетите. В базата данни се записва необходимата информация за свободните и заетите паркоместа.

Благодарение на информацията от снимките, от базата данни се извлича информация за данни за паркиране (това включва натоварени райони, не толкова натоварени райони и т.н.) и по този начин подмодулът „Информация, предназначена за дрона“ на модула „Мениджър на данни“ в сървъра има възможност да генерира нов маршрут на дрона, след което съответният дрон трябва да промени своя маршрут. Дроновете имат вградени GPS устройства, с помощта на които е възможно да се следи за правилното им движение по определения маршрут, както и да се извлича информация за районите, в които се намират. В подмодула „Информация от дрона“ в модула „Мениджър на данни“ в сървъра се проверява дали дронът се движи по генерирания му маршрут. Ако се установи, че дронът се движи по определения му маршрут, тоест няма отклонение, то тогава процесът приключва. Ако обаче се установи отклонение от определения маршрут, то тогава е необходимо мениджърът на изправност да уведоми аварийните групи за поява на неизправност. Проверява се дали аварийните групи са получили сигнала – ако той не е получен, то тогава мениджърът на изправност отново трябва да направи опит да го изпрати, но ако няколко пъти тази дейност е неуспешна, то тогава вероятно има проблем в системата. Ако аварийните групи са получили сигнала за неизправност, то тогава процесът приключва, защото те трябва да поемат отговорността за отстраняване на появилата се неизправност.

### 3.2.3 Заемане на паркомясто

#### 3.2.3.1 Първично представяне





### 3.2.3.2 Описание на елементите и връзките

Диаграмата представя как протича процесът при заемане на паркоместо. Когато потребителят желае да заеме определено паркоместо, той влиза в приложението и избира как да продължи – ако има акаунт, може да влезе чрез него в системата, а ако не разполага с такъв, може да продължи и без акаунт. В сървъра в модул „Мениджър потребители“ се проверява дали потребителят е



избрал да продължи нататък с акаунт. Ако е избрал тази опция, той трябва да въведе своя имейл, като трябва да се провери в базата данни дали съществува такъв имейл (дали е направена регистрация с въведения имейл). Ако въведеният имейл не е валиден (не е записан в базата данни), то тогава потребителят не може да влезе в системата чрез този имейл и му се дава възможност да направи опит с друг имейл. В противен случай, ако въведеният имейл е валиден, то тогава потребителят трябва да въведе своята парола. Ако тя е неправилна (не съществува в базата данни и не съответства на въведения имейл), то тогава се дава възможност на потребителя да направи опит с друга парола. В противен случай, ако въведената парола е коректна, то тогава потребителят може да влезе в своя профил и да прегледа актуална информация за паркоместата, която се извлича от базата данни и която се обновява през определен период. Ако потребителят не е избрал да влезе със свой акаунт (избрал е опцията „Продължи без акаунт“), то тогава той може да продължи и да прегледа актуална информация за паркоместата, която се извлича от базата данни и която се обновява през определен период. След като потребителят прегледа информацията, която го интересува за паркоместата, то тогава той може да избере паркомясто. В базата данни се проверява дали избраното паркомясто е свободно.

- Ако то **не е свободно**, в мениджър потребители в сървъра се проверява дали потребителят е продължил с акаунт. Ако не е влязъл със своя профил, то тогава той може да избере друго паркомясто. Ако обаче е продължил със свой профил, в базата данни се проверява дали той има абонамент за избраното паркомясто. Ако няма, то тогава той може да избере друго паркомясто. В противен случай има две опции – да подаде сигнал за нарушител или отново да се върне на опцията за избор на паркомясто. При опцията за подаване на сигнал за нарушител мениджърът на изправност в сървъра прехвърля сигнала на групи по контрол. Проверява се дали групите по контрол са получили изпратения сигнал – ако не са, то тогава мениджърът на изправност в сървъра прави опит да изпрати сигнала отново, а ако са го получили, то тогава се дава възможност на потребителя да избере друго паркомясто или да откаже (необходимо е време, докато групите по контрол вземат необходими мерки относно съответния нарушител, и ако потребителят желае да заеме друго паркомясто, той може да го направи). В мениджър на данни в сървъра се проверява за избраната опция от потребителя – ако не желае да заеме друго паркомясто, то тогава процесът приключва, а в противен случай той се връща на опцията за избор на паркомясто.
- Ако избраното паркомясто **е свободно**, то тогава потребителят може да избере опция „Заеми избраното паркомясто“. Отново мениджърът потребители в сървъра проверява дали потребителят е влязъл в системата

чрез профил. Ако е влязъл с акаунт, то тогава има две опции – да вземе абонамент за избраното паркомясто или да плати избраното паркомясто. Ако не е влязъл с акаунт, единствената опция, с която разполага, е да плати избраното паркомясто. Следващата стъпка е да се избере начин на плащане за избраното паркомясто или за абонамент към него (ако регистриран потребител е избрал тази опция) – за момента има четири начина на плащане – чрез СМС, чрез кредитна карта, чрез дебитна карта и чрез PayPal. Който и начин да се избере, е необходимо да се въведат определени данни в зависимост от начина на плащане – например като номер на автомобил, номер на карта и други. Проверява се във външната система за плащане дали е успешно заплащането за избраното паркомясто или за абонамент към него. Ако не е успешно, то тогава потребителят би трябвало да въведе отново необходимите данни, защото може да е настъпила грешка при тяхното попълване. Ако заплащането е извършено успешно, то тогава в базата данни се записва необходимата информация – например отразено заплащане, закупен абонамент за съответния акаунт, както и маркиране на избраното паркомясто като заето в периода на закупения абонамент и други. С това процесът приключва.

### **3.2.3.3 Описание на обкръжението**

Този процес изисква да се направи проверка във външната система за плащане дали успешно е извършено заплащането на абонамент (за регистрираните потребители, ако са избрали да закупят абонамент), или за избрано паркомясто (за всички потребители).

### **3.2.4 Описание на възможни вариации**

С течение на времето и с развитието на технологиите е възможно да се добавят нови компоненти в системата, както и да се използват допълнителни външни системи, които ще доведат до промяна на представените диаграми. Разбира се, описаните три процеса не са единствените. Наясно сме с това, че системата е огромна и предлага много повече услуги. Изпълнението на всяка една от тях може да се представи чрез диаграма и да бъде описана по подробен начин.

## **4. Архитектурна обосновка**

Създаването на една система се базира на определени изисквания (функционални и качествени), които трябва да бъдат удовлетворени. В основата на нейното изграждане стоят такива изисквания, които се считат за основни. Те се наричат още архитектурни драйвери. В следващите точки по-долу сме описали всички архитектурни изисквания към нашата система и сме отбелязали кои според нас са архитектурните драйвери.

## **4.1 Архитектурни изисквания и обосновка**

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

Целта е да се разработи софтуерна система за следене и управление на свободните паркоместа. Това изискване ни дава конкретни насоки как това би могло да се осъществи – чрез система от дроне, които обикалят града и да заснемат отгоре зоните за паркиране, като идентифицират свободните паркоместа и дават информация за тях. Достатъчно е да се създаде основа, върху която да се изградят останалите изисквания, за да може да се разработи успешно необходимата система. Това изискване е изпълнено от нашата система чрез отделните модули. Дроновете обикалят по генериран маршрут и заснемат изображения, които изпращат на сървъра. Модулът „Мениджър за разпознаване на свободни паркоместа“ в главния модул „Сървър“ използва специфичен алгоритъм за разпознаване на свободните паркоместа, като информацията за тях се обновява в модула „Конфигурация база данни“, а след това се предоставя и на потребителите.

### **2. Броят на летящите в момента дроне 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**

Системата предлага възможност за съхранение на заснетите от дроновете снимки. По този начин се предоставя и възможност за наблюдение в кои дни от седмицата и в кои часове има интензивно използване на системата. Чрез направена статистика може да се прецени броят на летящите дроне за всеки ден и час от седмицата, както и да се избере подходящ маршрут за всеки един поотделно. Чрез такъв подход се постига икономия на ресурси както от страна на дроне, така и от страна на групи потребители, които се занимават главно с наблюдението на дроне. Преди да се приложи напълно такъв метод на действие, е необходимо системата да се подложи на изпитаемост – дали дроновете ще отчетат поставената им бройка за деня, дали ще успеят да възприемат възложения им маршрут, дали ще успеят да осъществят връзка с регистрираните или обикновените потребители, ако предния ден не са били активни. Интересува ни дали динамичната обстановка ще се отрази добре на нашата система и именно поради тази причина е необходима предварителна проверка не само на променените функции, а и на непроменените, за да се установи връзката между тях и отделните компоненти.

Реализацията на това изискване включва няколко модула. Първият от тях, „Мениджър за разпознаване на свободни паркоместа“ от главния модул „Сървър“, отговаря след разпознаване на паркоместата информацията за тях да се записва в модула „Конфигурация база данни“. Подмодулът „Данни за паркиране“ от модула „Архив“ в модула „Конфигурация база данни“ включва като информация динамиката на паркиране в съответния ден и час от седмицата, което ще помогне да се определи броят на летящите дроне, както и маршрутът на всеки от тях. Ако се налага промяна в маршрутите и броя на дроновете се включва и третият модул – „Мениджър на данни“ и по-конкретно „Информация, предназначена за дрона“, чиято цел е да информира всички засегнати от промяната дроне и да изпрати новите им маршрути. Комуникацията между отделните модули се извършва чрез изпратени заявки, които се обработват от „Мениджър заявки“ и се изпращат до съответните модули.

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

Бройката на дроновете и техният маршрут се определя не само от деня и часа, но и от метеорологичните условия. Чрез статистика, направена на база на запазените изображения, може да се уточни бройката за съответния ден и час, но не може чрез нея да се определят и метеорологичните условия, защото те постоянно се менят. Използвайки външна услуга за прогноза за времето, ние можем да помогнем на дроновете да разпознават климатичните условия и например да уведомяват операторите, ако остават броени минути до валежи, до наличие на мъгла или до достигане на определена температура, които да окажат влияние върху работата на дроновете. По този начин е възможно да се предупредят всички групи потребители за настоящо или предстоящо време и да се предприемат действия, които да са благоприятни за тях. Благодарение на индикаторите за установяване на неизправност, с които разполагат дроновете, се дава възможност на модула „Мениджър на изправност“ в главния модул „Сървър“ да уведоми операторите на системата при невъзможност за заснемане на изображения поради наличие на мъгла например или при други метеорологични затруднения.

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

Чрез специфичен алгоритъм за разпознаване на свободните паркоместа от предоставените изображения от дроновете системата има възможност успешно да обработи получените заявки на обикновените или регистрираните потребители. В главния модул „Сървър“ е разположен

подмодулът „Мениджър за разпознаване на свободните паркоместа“, който използва именно този алгоритъм. Този алгоритъм ни служи като гаранция за това, че работи изрядно и че е изпробван за странични ефекти. Не е изключено да се появят дефекти по време на работа, но заради това има определени групи потребители, които се занимават с отстраняването на неизправности. Чрез специфичния алгоритъм се спестяват средства и се спомага за по-бързото обслужване на чакащите потребители. След използването му от модула „Мениджър за разпознаване на свободните паркоместа“ информацията за свободните и заетите паркоместа се обновява в модула „Конфигурация база данни“, извлича се необходимата информация за тях, предоставя се на подмодула „Достъпна информация за потребители“ в модула „Мениджър на данни“, след което потребителите имат възможността да я прегледат и да продължат да използват останалите функционалности на системата.

## **5. Системата поддържа следните групи потребители:**

- Администратор**
- Оператор**
- Аварийни групи**
- Групи по контрол на паркирането (т.нар. „паяци“)**
- Регистрирани потребители**
- Обикновени потребители**

Необходимо ни е да знаем кои всъщност ще са потребителите на нашата система, за да можем да съставим подходящи функции за всяка една отделна група потребители, както и различен изглед, през който да имат достъп до необходимите за тях функции. Въпреки че отделните групи потребители на практика извършват различни дейности, тях ги свързва самата система, през която ще могат да комуникират помежду си. Чрез това разделение на групи се създава възможност за промяна на точно конкретизирани функции, без това да повлияе пряко на останалите групи и без да се спира работата на цялата система. Промяната всъщност може да се осъществи във всички аспекти на системата – във функциите ѝ, в обкръжението ѝ, в капацитета ѝ на потребители или клиенти. Тя може да се случи и по-нататък във времето, когато е необходима поява на нови елементи например. Поради тези причини нашата система е неизбежно свързана с изменяемостта. Когато се стремим един продукт да е успешен, ние не може да разчитаме само и единствено на текущия му изглед и на сегашните му способности, а трябва да се стремим да го подобрим не само за да запазим активните потребители, а и за да привлечем нови. Реализацията на това изискване се осъществява посредством няколко модула. Първо, модулът „Мениджър заявки“ от главния модул „Потребителски интерфейс“ изпраща заявки за разпознаване на потребителя

при опит за влизане в системата. Началният интерфейс, свързан с осигуряване на вход в системата, е еднакъв за всички потребители. Необходимо е да се въведат имейл и парола. След като модулът „Мениджър на данни“ от главния модул „Сървър“ получи заявка, тя се изпраща към подмодула „Информация от потребителите“ на модула „Мениджър на данни“. Заявката се пренасочва към модула „Мениджър потребители“. Този модул отговаря за разпознаването на статуса на потребителя и за зареждането на съответния интерфейс. Всяка от групите Администратори, Оператори, Аварийни групи и Групи по контрол на паркирането се считат също за регистрирани потребители, тоест те също могат да избират паркомясто, да закупуват абонамент и да подават сигнали. Избрали сме да ги обособим по такъв начин, защото за тях е необходимо да имат акаунт, за да могат да изпълняват задълженията си към системата, но също така те могат да го използват и като потребители, вместо да създадат допълнителен акаунт и да използват и него. **(Вижте 2.3.1 Предназначение на модула „Потребителски интерфейс“)**

**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 дронът, и да отстранят повредата**

Дроновете са част от системата. На тях им се подава информация – колко на брой да бъдат през дните от седмицата, в кои райони заснемат изображения, на каква височина да се намират спрямо метеорологичните условия, а други групи потребители получават информация от главния модул „Сървър“, благодарение на тях – например кои паркоместа са свободни и кои са заети, местоположение на паркинг. Никога обаче не можем да бъдем сигурни, че ще липсва негативно поведение. Ако някой от дроновете се отклони от подадения му маршрут, то тогава настъпва дефект в системата ни, който трябва да бъде замаскиран за възможно най-кратко време. Именно поради тази причина аварийните групи трябва да получат необходима информация за излезлите от строя дронове, за да могат да отстранят появилата се повреда. По този начин е голяма възможността да се избегне срыв, който да бъде забелязан от крайните потребители и който да доведе до негативни отзиви относно системата.

Дроновете имат вградени GPS устройства, които дават информация на системата дали се следват генерираните маршрути и кои са местоположенията, на които се намират в даден момент. Модулът „Регулатор“ в главния модул „Дронове“ съдържа подмодули „GPS устройство“ и „Индикатор за установяване на неизправност“. Неговата основна задача е да следи за неизправности, свързани както с отклоняване

на определен дрона от маршрута, така и при възникнала негова техническа неизправност. При наличие на проблем се изпраща заявка до модула „Мениджър на заявки“ от главния модул „Дронове“ до модула „Мениджър на заявки“ в главния модул „Сървър“. При получена такава заявка модулът „Мениджър на изправност“ уведомява Аварийните групи, като в заявката е изпратено последно местоположение на дрона, както и засечената неизправност. Тези заявки се изпращат, докато заявката не стане „Одобрена“, за да сме сигурни, че необходимите групи са забелязали открития проблем и са поели отговорност за отстраняването му. Ако обаче дори в самия модул „Регулатор“ съществува някакъв проблем, е необходимо да има механизъм за бързото му откриване. Когато дроновете изпращат заснети изображения, се използва тактиката за „Ехо“, която ако през определен период не получи информация от даден дрон, информира аварийните групи, като изпраща последното засечено местоположение.

**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**

Обновяването на информация е периодично събитие, което в случая трябва да се случва най-малко през 1 минута. Най-често такива периодични събития се наблюдават в real-time системите, а нашата трябва да е именно такава. Предмет на производителността е времето, за което системата реагира на възникващите събития. Потребителите имат нужда да бъдат уведомени за най-скоро настъпилите събития – дали мястото, където искат да паркират в момента, разполага с изправни и свободни дронаве, които съответно да ги уведомят при наличие на свободно паркомясто, а ако имат абонамент за дадено паркомясто, дали тяхното запазено паркомясто е заето от друго превозно средство, или не.

За реализирането на това изискване е необходимо сървърът, който използва системата, да разполага с високоскоростна връзка и интернет, за да се избегне забавяне в актуализирането на необходимата информация.

**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**

Докато се извършва заснемане на свободните паркоместа, е възможно да се появят пречки, които да намалят или да ограничат тяхната видимост. В такъв случай е необходимо да се уведоми операторът на системата, за да може да се вземат мерки, които да доведат до успешно заснемане на паркоместата и

след това да се върне отговор на изчакващия потребител. Ако в такава ситуация не се уведоми операторът, а вместо това дронът извести потребителя, че не е в състояние да заснеме заявените паркоместа, то тогава автоматично се отказва обслужване на клиента. Цялото това положение може да доведе до намаляване на броя на обикновените или регистрираните потребители, чиято заявка е била отхвърлена поради странични причини. За да се избегне такъв тип недоразумение, добър подход е да бъдат уведомени операторите на системата, за да могат да предотвратят загубата на клиенти, като например променят височината на проблемните дроне при наличие на мъгла или променят маршрута им, за да могат да имат по-ясен изглед, който да им предостави възможност за заснемане на паркоместата.

Това изискване реализираме благодарение на вградените в дроновете индикатори за установяване на неизправности и на външната услуга за прогноза за времето. Задачата на подмодула „Индикатор за установяване на неизправност“ в главния модул „Дроне“ е да сигнализира, ако се създаде невъзможност за заснемане на изображения поради наличието на мъгла например. Ако е установен проблем, подмодулът „Мениджър заявки“ в главния модул „Дроне“ изпраща заявка, която след обработване се очаква да бъде получена от модула „Мениджър на изправност“ в главния модул „Сървър“, за да се уведомят операторите на системата. Междувременно операторите на системата следят стриктно при получени заявки дали все пак ще е възможна работата на дроновете, за да може при първа възможност те отново да бъдат активни. Външната услуга за прогноза за времето комуникира с модулите от сървъра и на база на данните, които изпраща, се преценява дали да бъдат уведомени операторите на системата. Възможно е да се появи ситуация, в която е необходимо да се промени височината на летене на всички активни дроне, за да се избегне затруднение при заснемането на изображения поради метеорологични условия.

**9. Регистрираните потребители могат да заплащат абонамент за определено паркомясто, 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**

Освен заплащане за определено паркомясто, системата предлага и закупуване на абонамент за определен период от време. Ако даден потребител иска да закупи абонамент за дадено паркомясто, то е необходимо той да влезе или да се регистрира в системата. След като вече е влязъл със свой акаунт, той има възможност да направи избор за паркомясто и да закупи абонамент за него. Необходимо е системата да получи потвърждение за извършване на плащане, както и за начин на плащане, за



да може потребителят да бъде прехвърлен във външната система за извършване на плащане. Получена заявка за заплащане на паркомясто или за закупуване на абонамент се обработва от модула „Мениджър заявки“ на главния модул „Сървър“. След това информацията от нея се прехвърля и запазва в подмодулите „Отразени заплащания“, „Потребители“ и „Данни за паркоместа“ в модула „Конфигурация бази данни“. Именно от модула „Данни за паркоместа“ се извлича информация за свободните и заетите паркоместа и след като в него е записано, че дадено място е заето за определен период, то задължително се показва като заето за този период.

**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**

След като изображенията, получени от дрона, се обработват от „Мениджър за разпознаване на свободни места“ и информацията от него се запише в „Данни за паркоместата“, модулът „Заемане на паркомясто“ използва обработената информация. За паркоместа, които са с платен абонамент, те не се обработват. За паркоместата, които не са с платен абонамент, се преминава през модула „Генериране на цена спрямо зоната“. Цената се генерира спрямо информацията от подмодула „Данни за паркиране“ в модула „Архив“. На базата на тази информация се изготвя статистика за конкретния регион и на базата на натовареността се определя и цената, която се съхранява и обновява и в модула „Данни за паркоместа“. Дадено паркомясто може да бъде безплатно в даден ден или времеви период. Ето защо модулът „Генериране на цената спрямо зоната“ е важно да може да генерира правилно цените на незаетите паркоместа.

**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**

Необходимо е по някакъв начин да се извършва плащане в изградената софтуерна система за заемане на определени паркоместа. Като пример могат да се вземат сините и зелените зони, както и местата на платените паркинги. Паркирането на тези места има конкретна цена за точно определено време. Нашата система е създадена с цел улесняване на хората, които вместо да обикалят и да търсят свободни паркоместа, да получат информация за налични такива от летящите дроне. Не може по никакъв начин да се заобиколи процедурата по заплащане за заемането им. Ако хората не използват тази система, те отново са задължени да платят за синя и зелена зона или за платените паркинги. Поради тази причина е

необходимо да се осигури метод на заплащане – чрез дебитна/кредитна карта, PayPal или СМС, като в бъдеще е възможно да се добавят и допълнителни начини.

Това изискване е дефинирано чрез външна система за плащане. Сървърът комуникира с нея чрез модула „Извършване на плащане“ и съответните подмодули. След като модулът „Мениджър заявки“ получи заявка от външната система за успешно извършено плащане, то той изпраща заявка със съответната информация до подмодулите „Отразени заплащания“ и „Потребители“ в модула „Конфигурация база данни“. В зависимост от избрания начин на плащане от страна на потребителя, се зарежда конкретна форма за попълване на необходими данни. След това те се изпращат до външната система за обработка, след това се извършва плащането и се потвърждава.

## **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**

Изискването е изпълнено чрез разработване на мобилно приложение, което предлага различен интерфейс и функционалности спрямо типа потребител, който го използва. Обикновените потребители имат право да прегледат информация за свободните паркоместа, да изберат свободно паркоместо, на което да паркират своя автомобил, да платят за него и да прегледат информация за оставащо време. Регистрираните потребители освен правата, които имат обикновените потребители, имат право да закупят абонамент за определен период от време, както и да прегледат информация за заетите паркоместа, за да могат да подадат сигнал за нарушител, ако съответно забележат такъв. Аварийните и групите по контрол освен изброените права на по-горните потребители, могат да получават сигнали за неизправности в системата. При описаната по-горе допълнителна архитектурна структура „Структура на внедряването“ се забелязва, че мобилното приложение е достъпно на операционните системи Android и iOS, като е необходимо да се отбележи, че те използват един и същ програмен код.

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

Тези две групи потребители разполагат с информация, свързана с управление на дроновете и съхранение на заснетите изображения и издадените фишове за глоби. Достъпът до този тип данни трябва да е забранен за външни лица, за да се избегне злоупотреба, която да доведе до

проникване в системата. Само един пропуск в сигурността на системата може да доведе до нейния пълен срив, а той може да се постигне, като например да се пуснат в действие абсолютно всички налични дроне, да се изтрие специфичният алгоритъм за разпознаване на заснетите изображения, да се изтрият архивите, съдържащи заснети изображения и фишове за глоби, да се пусне фалшива информация на останалите групи потребители, да се променят маршрутите на дроновете и много други. Всъщност не е трудно да се направи пробив в система, която има липса на сигурна защита. Тогава случаите на злоупотреба биха били доста повече. Именно за да се избегне такъв тип ситуация, е нужно да се вземат предварителни мерки за ограничаване на достъпа на системата.

За реализацията на това изискване използваме различни тактики за сигурност. Основната и най-често срещаната е автентикацията на потребителите чрез парола. Тя е от най-ниско ниво и е задължителна. Допълнително към нея сме сложили оторизация на всеки потребител. Това включва при всяка заявка за достъп до ресурси да се прави проверка дали този потребител има достъп до тях. Тъй като системата съхранява голямо количество чувствителна информация, използваме криптиране както на постоянната памет, така и на комуникационните канали. Също така имаме хеш-алгоритми и чек суми, които са механизми на излишък. Ограничили сме и експозицията на системата. Тези места са ограничени и допълнително защитени от външен достъп. За да има достъп до тях, е необходимо да бъдат изпълнени определени условия.

**14. Групите по контрол на паркирането следят дали няма нарушители (неплатили 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**

Това изискване е реализирано, чрез публичен сайт, който е достъпен за потребителите. При описаната по-горе допълнителна архитектурна структура „Структура на внедряването“ се забелязват компонентите „Сървър“, „Компютри“ и „Уеб сървър“, които комуникират помежду си чрез протоколи и обменят информация за наличното уеб приложение (**Вижте Описание на Структура на внедряването**). При установяване на нарушител по време на принудителното преместване на автомобила се прави заснемане на събитието от групите по контрол. След това те изпращат заснетото събитие до главния модул „Сървър“ чрез заявки. Тези заявки се обработват от модула „Мениджър заявки“ и информацията от тях се пренасочва и записва в подмодулите на модула „Архив“ – „Фишове за глоба“ и „Снимки на

нарушители“. На базата на тези снимки, които потребителите могат да прегледат единствено през уеб-сайта, модулът „Фишове за глоби“ издава електронен фиш на съответния нарушител и му го изпраща. Издадените фишове се запазват в подмодула „Архив“ на модула „Конфигурация база данни“ и могат да се достъпят публично на уеб-сайта.

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

Когато даден регистриран потребител установи, че абонаментното му паркомясто е заето неправомерно, то той може да подаде сигнал.

Подаването на сигнал включва създаване на заявка. След като се попълни, следва да се потвърди и изпрати до сървър, който да уведоми групите по контрол на паркирането. След като заявката е одобрена, регистрираният потребител има две възможности – да изчака групите по контрол да пристигнат и да преместят автомобила на нарушителя, или да избере друго свободно паркомясто, където да паркира своя автомобил.

**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)**

В интервала на работното време се очаква, че системата ще получава най-много заявки и ще регистрира най-много активност. Точно в този момент трябва да се гарантира на потребителите безотказно и по възможност бързо обслужване. Системата ни трябва да е надеждна и да не прекъсва процеса на функциите при голяма натовареност. В случай, че няма налични дроне, които да приемат дадена заявка, е силно препоръчително системата да уведоми съответните потребители и да ги помоли за изчакване, като при първа възможност освободени дроне да започнат да приемат и да обработват изчакващите заявки. В такава ситуация е от голямо значение дроновете да са в пълна изправност и да успяват бързо да отговорят на обикновените или регистрираните потребители. Останалите групи потребители – администрацията, операторите, аварийните групи и групите по контрол на паркирането, също трябва да са в готовност да се заемат с постъпилите заявки, а това би могло да стане, ако системата е надеждна и издръжлива на голяма натовареност. Тя трябва да поддържа и комуникацията между отделните групи потребители, за да не се губи връзката между тях и за да може всички те да са удовлетворени както от своята работа, така и от работата на системата.

За да осигурим 100% наличност на системата използваме различни тактики за това. Една от основните тактики е тази на Ехо (Пингване). Благодарение на нея контролираме дали дроновете са активни и дали изпълняват задачите

си. Друга тактика, която използваме в нашата система, е тази на активен излишък. Тази тактика включва дублиране на важните компоненти в системата и поддържането им в едно и също състояние. В нашия случай използваме активен излишък за сървър. Той е един от ключовите компоненти и служи за комуникация между всички останали в системата. Ето защо, благодарение на тази тактика, успяваме да осигурим максимална наличност на системата дори и при спад на някой от дублираните компоненти. Друга тактика, която използваме, е пасивен излишък. Той е използван за дроновете и потребителския интерфейс. Така при проблем с дрона веднага може да се включи нов на негово място, докато проблемът не бъде отстранен и дронът не бъде въведен отново в употреба. Разбира се, за всеки от дублираните компоненти използваме разнородност при тяхното създаване. Тя включва различни програмни езици, алгоритми, компилатори, ограничен контакт и други.

**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**

Съхранението на данни е една добра възможност, която може да предложи дадена система, стига само да не се злоупотребява със запазената информация. За да се избегне това, системата трябва да предлага сигурност на тези архиви от данни и да не допуска външни и неквалифицирани лица да имат достъп до тях. Съхранените данни биха послужили за доказателство при разследване на определен случай от страна на МВР например. Необходимо е данните да са запазени и защитени за определен период от време, защото не се знае кога съхранената информация ще послужи за добри цели.

Специално за изпълнението на това условие главния модул „Сървър“ са разположени модулът „Конфигурация база данни“ и един от неговите подмодули – „Архив“. В него се съхранява цялата изискана информация за период от посочените срокове. За да бъде максимално защитена, има ограничен достъп до този модул. Също така е дублиран – за използването на тактиката пасивен излишък. За да бъде достатъчно надеждна системата, всяка вечер, когато задължителното условие за наличност не важи, бива извършвана смяна между дубликатите. Така се осигурява максимална актуалност и синхронизираност на отделните компоненти.

## 4.2 Архитектурни драйвери

Архитектурните драйвери, които избрахме при изграждането на системата, са:

- ❖ **Свободните паркоместа се идентифицират от система от дроне, които обикалят града и заснемат зоните за паркиране (отгоре)**
- ❖ **Броят на летящите в момента дроне 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**
- ❖ **Системата поддържа следните групи потребители:**
  - Администратор
  - Оператор
  - Аварийни групи
  - Групи по контрол на паркирането (т.нар. „паяци“)
  - Регистрирани потребители
  - Обикновени потребители
- ❖ **Ако някой дрон излезе от строя, незабавно трябва да се уведомят аварийните групи, които да получат информация за предполагаемия район, в който се намира дронът, и да отстранят повредата**
- ❖ **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**
- ❖ **При трайно намалена видимост (напр. мъгла), която води до невъзможност да се заснемат паркоместата, да се вземат мерки за известяване на оператора на системата**
- ❖ **Плащането може да се извършва чрез дебитна/кредитна карта, PayPal или СМС, като в бъдеще може да се добавят и други начини на разплащане**
- ❖ **Администраторите и операторите трябва да имат 100% защитен от външна намеса достъп до системата**
- ❖ **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)**
- ❖ **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**