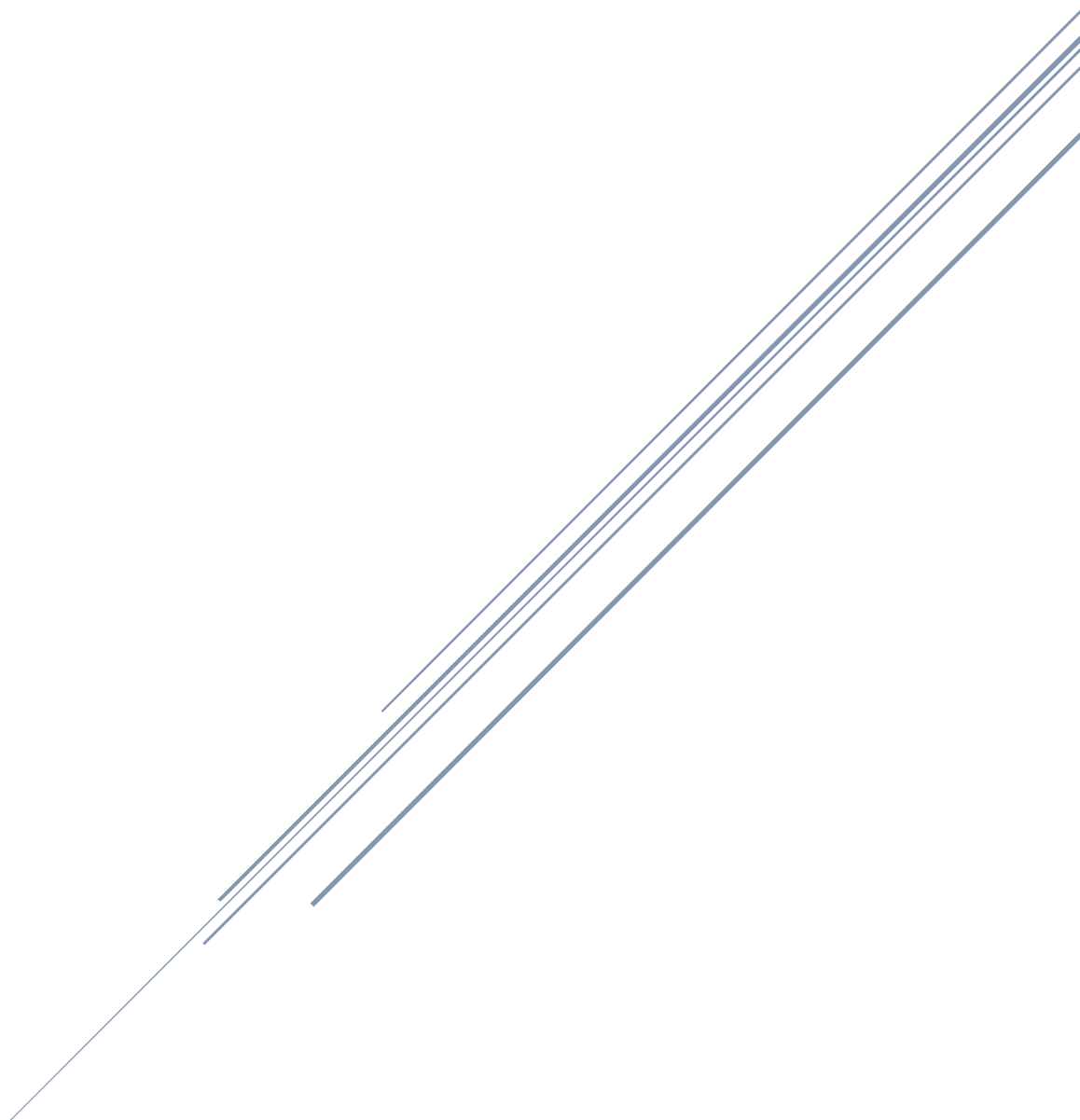


VeloCity

КУРСОВА РАБОТА ПО СОФТУЕРНИ АРХИТЕКТУРИ И
РАЗРАБОТКА НА СОФТУЕР



Веселина Послийска 62498
Ирина Христова 62473

Съдържание

<i>VeloCity</i>	0
1. Въведение	5
1.1. Обща информация за текущия документ	5
1.1.1. Предназначение на документа	5
1.1.2. Описание на използваните структури на архитектурата	5
1.1.3. Структура на документа	5
1.2. Общи сведения за системата	5
1.3. Терминологичен речник	6
2. Декомпозиция на модули	7
2.1. Общ вид на декомпозиция на модулите за системата	7
2.2. Контекстна диаграма	8
2.3. Velo Manager	9
2.3.1. Предназначение на модула	9
2.3.2. Основни отговорности на модула в системата	9
2.3.3. Описание на интерфейсите на модулите	10
2.3.3.1. Location manager	10
2.3.3.1.1. Locate	10
2.3.3.1.2. Send	10
2.3.3.2. Diagnostics manager	10
2.3.3.2.1. Diagnose	10
2.3.3.2.2. Send	10
2.3.3.3. Signal manager	10
2.3.3.3.1. Send	10
2.3.4. Други	11
2.3.4.1. Ограничения при употреба	11
2.3.4.2. Грешки и изключения	11
2.3.4.3. Зависимости от други модули	11
2.4. Database	11
2.4.1. Предназначение на модула	11
2.4.2. Основни отговорности на модула в системата	11
2.5. MapApps	12
2.5.1. Предназначение на модула	12
2.5.2. Основни отговорности на модула в системата	12
2.5.3. Описание на интерфейсите на модулите	13

2.5.3.1.	Location manager	13
2.5.3.1.1.	Locate	13
2.5.3.1.2.	Send	13
2.5.3.2.	Map manager	13
2.5.3.2.1.	Add	13
2.5.3.3.	Connection manager	13
2.5.3.3.1.	Connect.....	13
2.6.	WebApp.....	14
2.6.1.	Предназначение на модула.....	14
2.6.2.	Основните отговорности на модула в системата.....	14
2.6.3.	Описание на интерфейсите на модулите	14
2.6.3.1.	Account Manager	14
2.6.3.1.1.	Login	14
2.6.3.1.2.	Logout.....	15
2.6.3.2.	Profile manager	15
2.6.3.2.1.	System Administrators	15
2.6.3.2.1.1.	Signals.....	15
2.6.3.2.2.	Observers.....	15
2.6.3.2.2.1.	Signals.....	15
2.6.3.2.2.2.	Violations.....	15
2.6.4.	Други.....	16
2.6.4.1.	Ограничения при употреба	16
2.6.4.2.	Грешки и изключения.....	16
2.6.4.3.	Зависимости от други елементи.....	16
2.7.	MobileApp.....	16
2.7.1.	Предназначение на модула.....	16
2.7.2.	Основни отговорности на модула в системата	16
2.7.3.	Описание на интерфейсите на модулите	17
2.7.3.1.	Account Manager	17
2.7.3.1.1.	Login	17
2.7.3.1.2.	Logout.....	17
2.7.3.2.	Registration Manager	17
2.7.3.2.1.	User	17
2.7.3.2.2.	Data	17
2.7.3.3.	Profile Manager	17

2.7.3.3.1.	Renters	17
2.7.3.3.1.1.	Parking Spots	17
2.7.3.3.1.2.	Time	17
2.7.3.3.1.3.	Battery %	17
2.7.3.3.1.4.	Payment.....	17
2.7.3.3.2.	Technical Support Group.....	17
2.7.3.3.2.1.	Signals.....	17
2.7.4.	Други.....	17
2.7.4.1.	Ограничения при употреба	17
2.7.4.2.	Грешки и изключения.....	17
2.7.4.3.	Зависимости от други елементи.....	18
2.8.	Computation.....	18
2.8.1.	Предназначение на модула.....	18
2.8.2.	Основни отговорности на модула в системата	18
2.8.3.	Описание на интерфейсите на модулите	19
2.8.3.1.	Velo manager	19
2.8.3.1.1.	Data receiver	19
2.8.3.1.2.	Sensor activator.....	19
2.8.3.1.3.	Notifier.....	19
2.8.3.2.	Maps manager.....	19
2.8.3.2.1.	Data receiver	19
2.8.4.	Други.....	19
2.8.4.1.	Ограничения при употреба	19
2.8.4.2.	Грешки и изключения.....	19
2.9.	Server.....	20
2.9.1.	Предназначение на модула.....	20
2.9.2.	Основни отговорности на модула в системата	20
2.9.3.	Описание на интерфейсите на модулите	21
2.9.3.1.	Account manager	21
2.9.3.2.	Access Manager.....	21
2.9.3.3.	Signal Dispatcher	21
2.9.3.4.	Update Manager.....	21
2.9.3.5.	Database manager.....	21
3.	Описание на допълнителните структури.....	21
3.1	Структура на процесите.....	21

3.1.1	Първично представяне	21
3.1.2	Наемане на велосипед	22
3.1.2.1	Описанието на елементите и връзките	22
3.1.2.2	Описание на обкръжението	23
3.1.2.3	Описание на възможните вариации	23
3.2	Структура на внедряването.....	23
3.2.1	Първично представяне.....	23
3.2.2	Описанието на елементите и връзките	24
3.2.3	Описание на обкръжението	24
3.2.4	Описание на възможните вариации	24
4.	Архитектурна обосновка	24

1. Въведение

1.1. Обща информация за текущия документ

1.1.1. Предназначение на документа

В този документ е представена архитектурата на приложението за отдаване на велосипеди под наем - *VeloCity*.

1.1.2. Описание на използваните структури на архитектурата

- **Декомпозиция на модулите:** Показва логическото разделение на приложението на модули и подмодули.
- **Структура на внедряването:** Избрахме „Структура на внедряването“ като втора допълнителна структура, защото тя показва особеностите относно бързодействието, надеждността, сигурността на нашата система.
- **Структура на процесите:** Избрахме „Структура на процесите“ като допълнителна структура, понеже тя илюстрира по един много удобен и лесен за разбиране начин протичането на основните процеси в нашата система.

1.1.3. Структура на документа

- Въведение (секция 1)
- Декомпозиция на модулите (секция 2)
 - Общ вид на декомпозицията на модулите
 - Контекстна диаграма
 - Подробно описание
 - Описание на възможните вариации
- Структура на внедряването (секция 3)
 - Първично представяне
 - Описание на елементите и връзките
 - Описание на обкръжението
 - Описание на възможните вариации
- Структура на процесите (секция 4)
 - Първично представяне
 - Описание на елементите и връзките
 - Описание на обкръжението
 - Описание на възможните вариации
- Архитектурна обосновка (секция 5)
- Допълнителна информация (секция 6)

1.2. Общи сведения за системата

VeloCity е софтуерна система за отдаване под наем на велосипеди в рамките на града . Основната цел на системата е потребителите през мобилно приложение да се информират за най-близките свободни велосипеди, с които може да се придвижват само в рамките на града. Велосипедите са снабдени с електрическа система за задвижване, но може да се задвижват и със собствена сила на колоездача, чрез педали.

Качествени изисквания:

- **Наличност** – Допуска се ремонт и профилактика в интервала от 2:30 до 5:30 ч. В останалата част на деня, системата трябва да е 99,999% налична. Системата трябва да е устойчива към пикови натоварвания в най-натоварените в денонощието.
- **Изправност** – При или загуба на връзка с даден велосипед, , трябва да се сигнализира наблюдателя в рамките на 10 сек, като се изпратят данни за движението на велосипеда в последните 30 мин, заедно с най-вероятната му позиция на която се е намирал в момента на изгубване на връзката.
- **Използваемост** - Максималното време Т за използване на един велосипед е конфигурационен параметър на сървъра на системата. След изтичане на максималното време се изпраща съобщение на наемателя и той трябва да остави велосипеда на найблизката стоянка и да го замени с друг ако му е необходимо.

1.3. Терминологичен речник

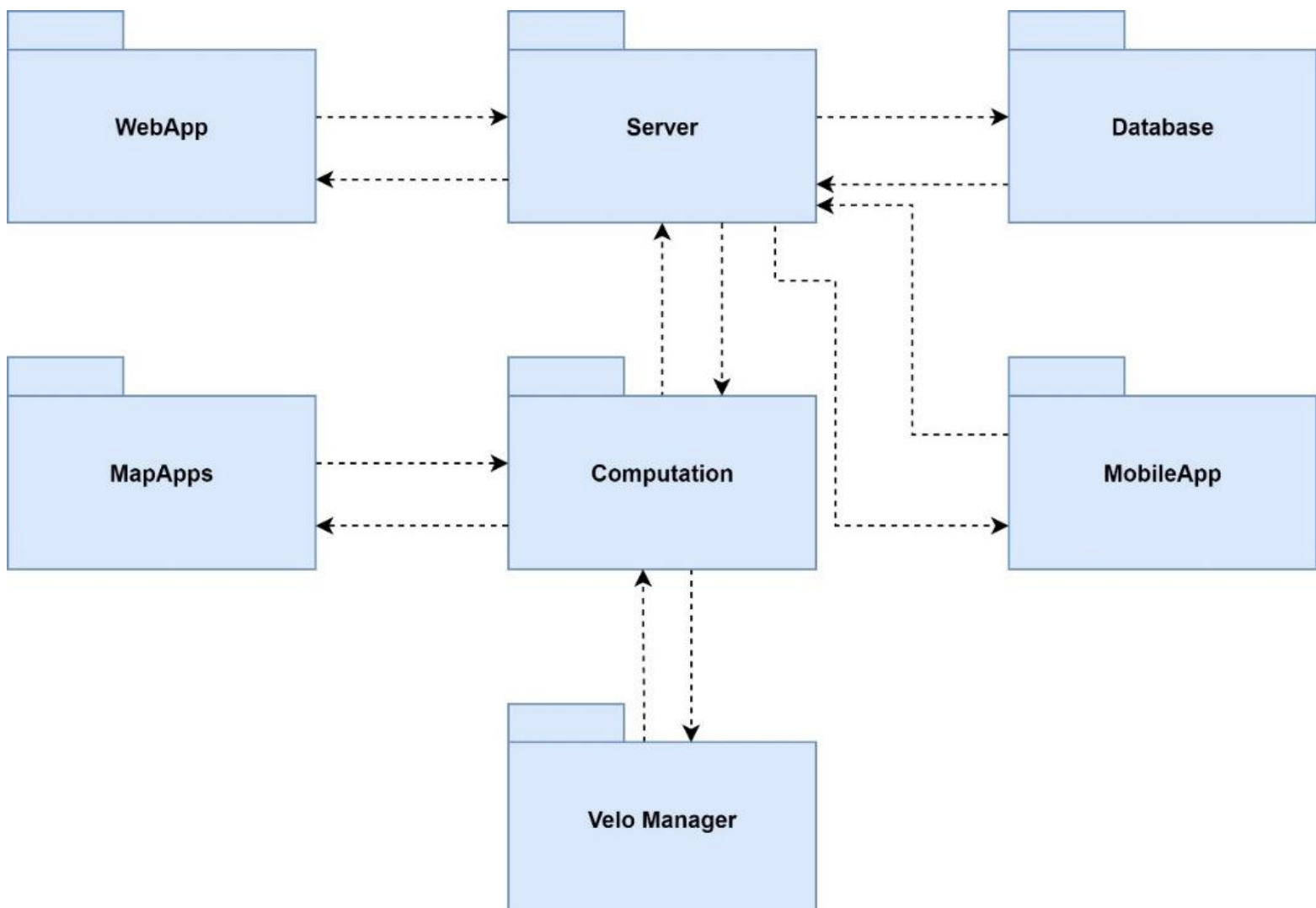
- Velo Manager
 - Location manager
 - Diagnostics manager
 - Signal manager
- Database
- MapApps
 - Map manager
 - Connection manager
- WebApp
 - Account Manager
 - Login/Logout
 - Profile manager
 - System Administrators
 - Signals
 - Observers
 - Signals/Violations
- MobileApp
 - Account Manager
 - Login/Logout
 - Registration Manager
 - User/Data
 - Profile Manager
 - Renters
 - Parking Spots/Time/Battery %/Payment
 - Technical Support Group
 - Signals
- Computation
 - Velo manager
 - Data receiver
 - Sensor activator
 - Notifier
 - Maps manager

- Data receiver
- Server
 - Signal Dispatcher

2. Декомпозиция на модули

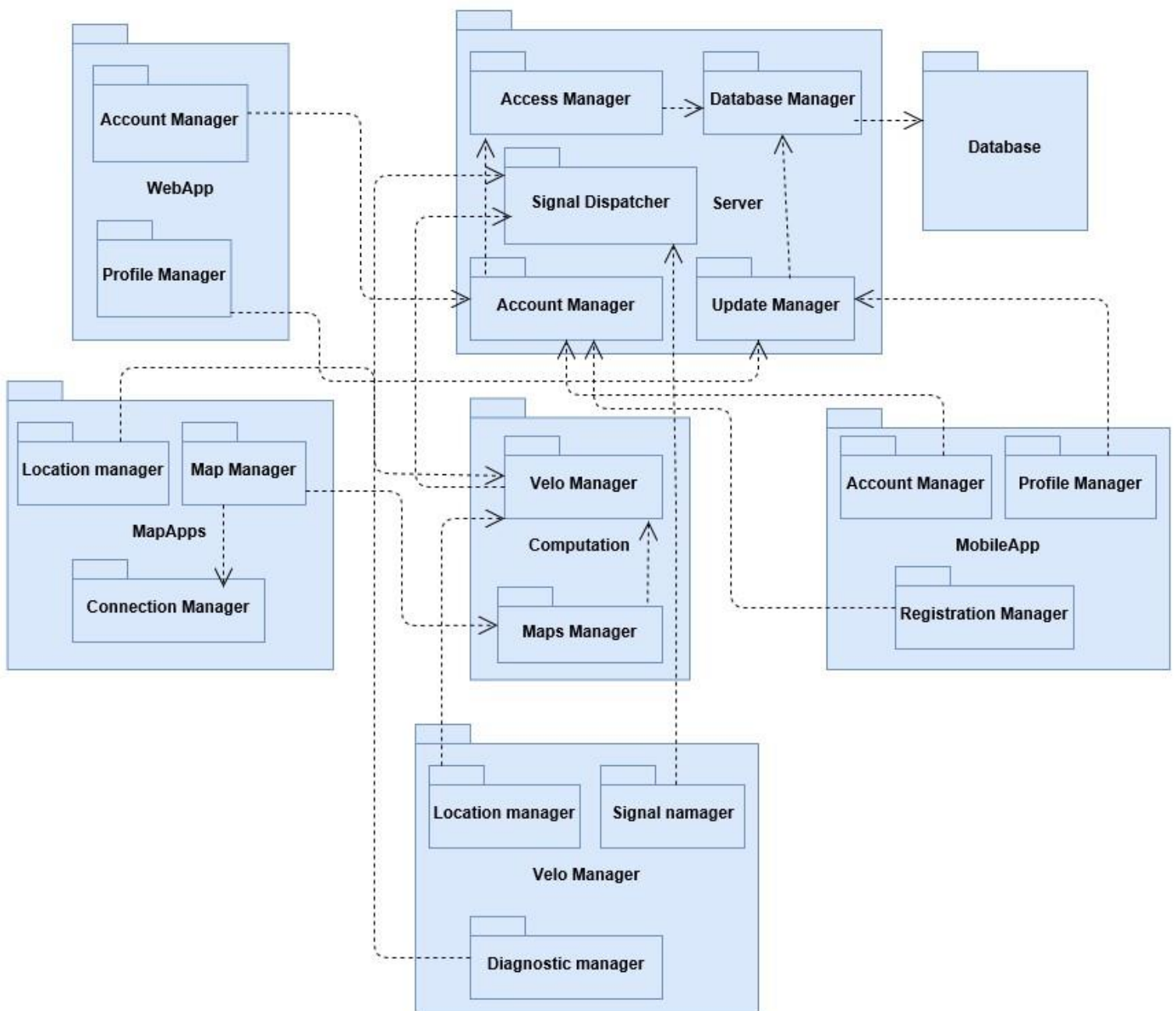
2.1. Общ вид на декомпозиция на модулите за системата

В декомпозицията на модулите са показани и връзките между отделните модули. Основните модули са: VeloManager, WebApp, Mobile app, Database, MapApps, Computation, Server. Server осъществява връзката между всички модули в системата. Съществува отделен модул Computation, който свързва Server, Velo Manager и външните приложения за карти. Mobile App е потребителското приложение, WebApp е модула, предназначен за техническия екип. Velo Manager представлява съвкупността от хардуер и софтуер върху всяко колело.

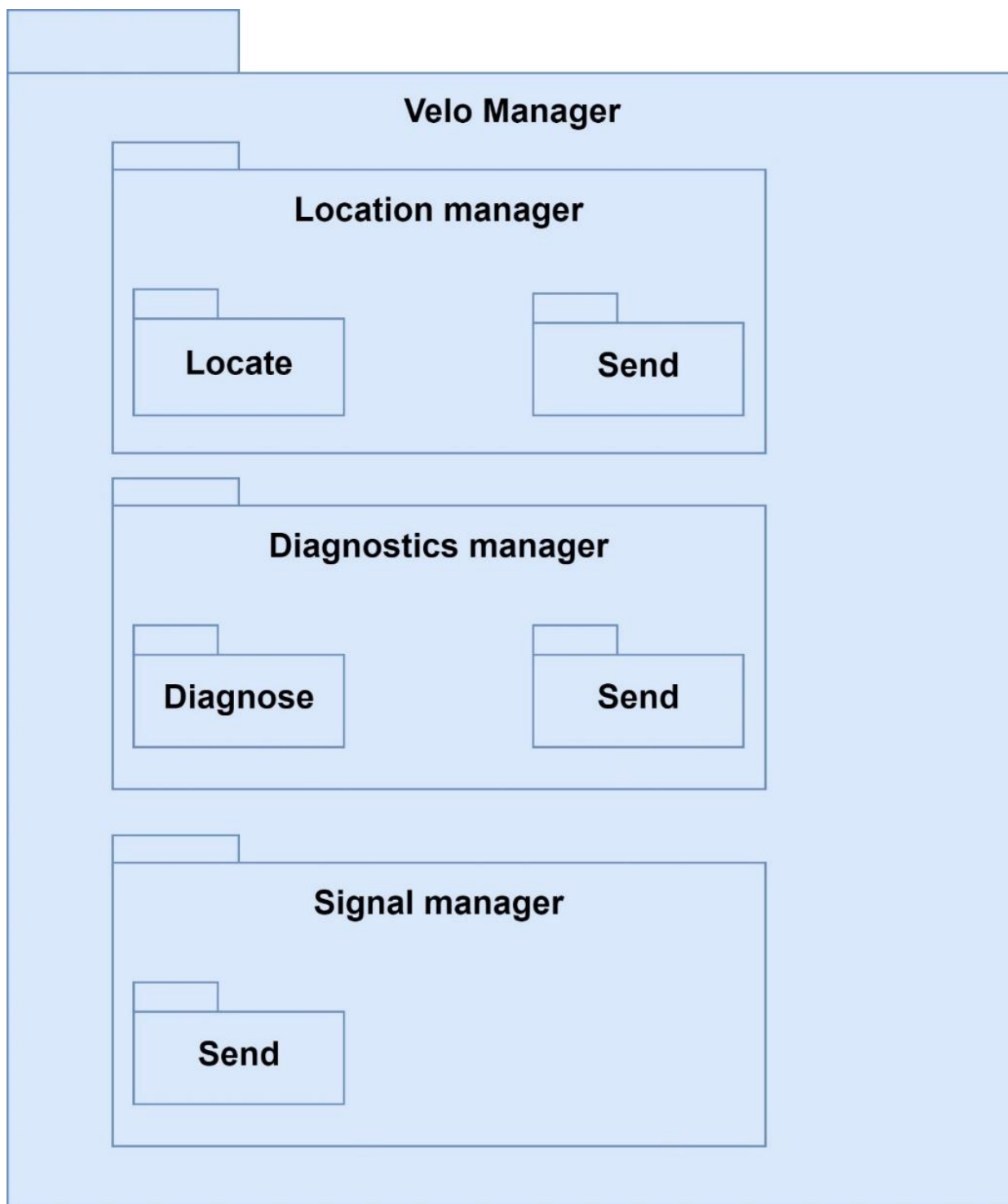


2.2. Контекстна диаграма

Показани са основните връзки между всеки един от модулите.



2.3. Velo Manager



2.3.1. Предназначение на модула

Получаване на информация за местоположението на велосипедите.

2.3.2. Основни отговорности на модула в системата

Локализиране на велосипедите. Изпращане на данни за обработка. Следи и сигнализира за изправността на grs-ите велосипедите.

2.3.3. Описание на интерфейсите на модулите

2.3.3.1. Location manager

Този модул отговаря за локацията на всеки велосипед.

2.3.3.1.1. Locate

Този модул отговаря за намирането на стоянките със свободни и заредени велосипеди. Това става с намирането на локацията на наемателя.

public bool LocateVelo(string location);

вход: локация на наемателя

изход: true, ако се е намерил свободен велосипед успешно и false, ако не се

2.3.3.1.2. Send

Този модул изпраща локацията на намереният велосипед към Computation модула.

private LocationVelos[] GetLocationVelos();

вход: void

изход: списък с всички локации на свободни велосипеди

public bool SendLocationVelos();

вход: void

изход: true, ако се е изпълнила успешно и false, ако не е

2.3.3.2. Diagnostics manager

Този модул отговаря за самодиагностиката на всеки велосипед.

2.3.3.2.1. Diagnose

Отговаря за самодиагностицирането на всеки велосипед.

private GetVeloInfo[] GetVeloInfo(string VeloID);

вход: id на велосипеда

изход: информация за състоянието на даден велосипед

2.3.3.2.2. Send

Този модул изпраща диагностиката на намереният велосипед към Computation модула.

public bool SendDiagnostics();

вход: void

изход: true, ако се е изпълнила успешно и false, ако не е

2.3.3.3. Signal manager

2.3.3.3.1. Send

Този модул отговаря за изпращането на сигнали през интервали от време към Server (Heartbeat tactic). Ако сигнали спрат ще бъдат взети определени мерки.

public bool SendSignal(string idVelo);

вход: id на велосипед

изход: true ако се е изпълнила успешно и false ако не е

2.3.4. Други

2.3.4.1. Ограничения при употреба

Сензорите не могат да изпращат сигнали на големи разстояния, затова при позиция, засечена от gps-ите, която е близо до максималния му обсег изпраща незабавен сигнал. Не могат да изпращат сигнали да супер малък интервал като 1 милисекунда.

2.3.4.2. Грешки и изключения

При възникване на грешка или изключение, дрона ще спре да изпраща сигнал и ще бъдат взети съответните мерки.

2.3.4.3. Зависимости от други модули

VeloManager зависи много от Computation, защото до него се изпраща за обработка всичката извлечена информация.

2.4. Database

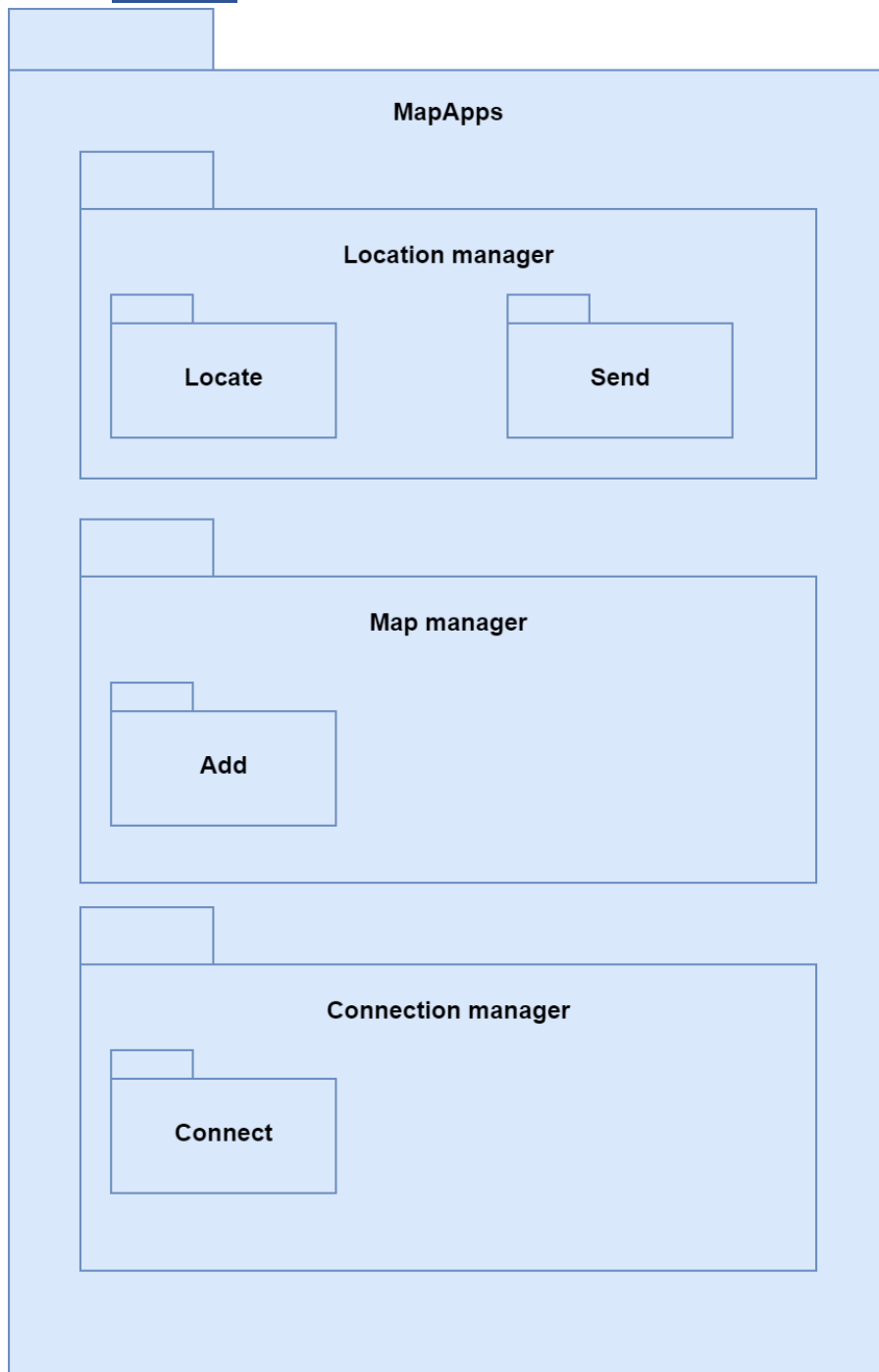
2.4.1. Предназначение на модула

Основната база от данни за системата.

2.4.2. Основни отговорности на модула в системата

Съдържа данни за състоянието на велосипедите, личните данни на потребителите и архив с локациите на велосипедите.

2.5. MapApps



2.5.1. Предназначение на модула

Снабдява системата с географски карти от всички познати онлайн услуги (Google maps, BG maps, Open Street maps и т.н.), като може и да се добавят нови карти.

2.5.2. Основни отговорности на модула в системата

Локализиране на велосипедите. Изпращане на данни за обработка. Добавя нови карти. Свързва се с мобилното приложение.

2.5.3. Описание на интерфейсите на модулите

2.5.3.1. Location manager

2.5.3.1.1. Locate

Този модул отговаря за намирането на локациите на стоянките със свободни велосипеди.

public string getVeloLoc();

вход: void

изход: връща списък с локациите на велосипедите

2.5.3.1.2. Send

public void sendLoc(string location);

вход: списък с локациите на велосипедите

изход: изпраща списъка с локациите на велосипедите до външните приложения за карти

2.5.3.2. Map manager

2.5.3.2.1. Add

public void addMap(string locations);

вход: списък с велосипедите и техните местоположения

изход: изпраща заявка за добавяне на нова карта във външното приложение

2.5.3.3. Connection manager

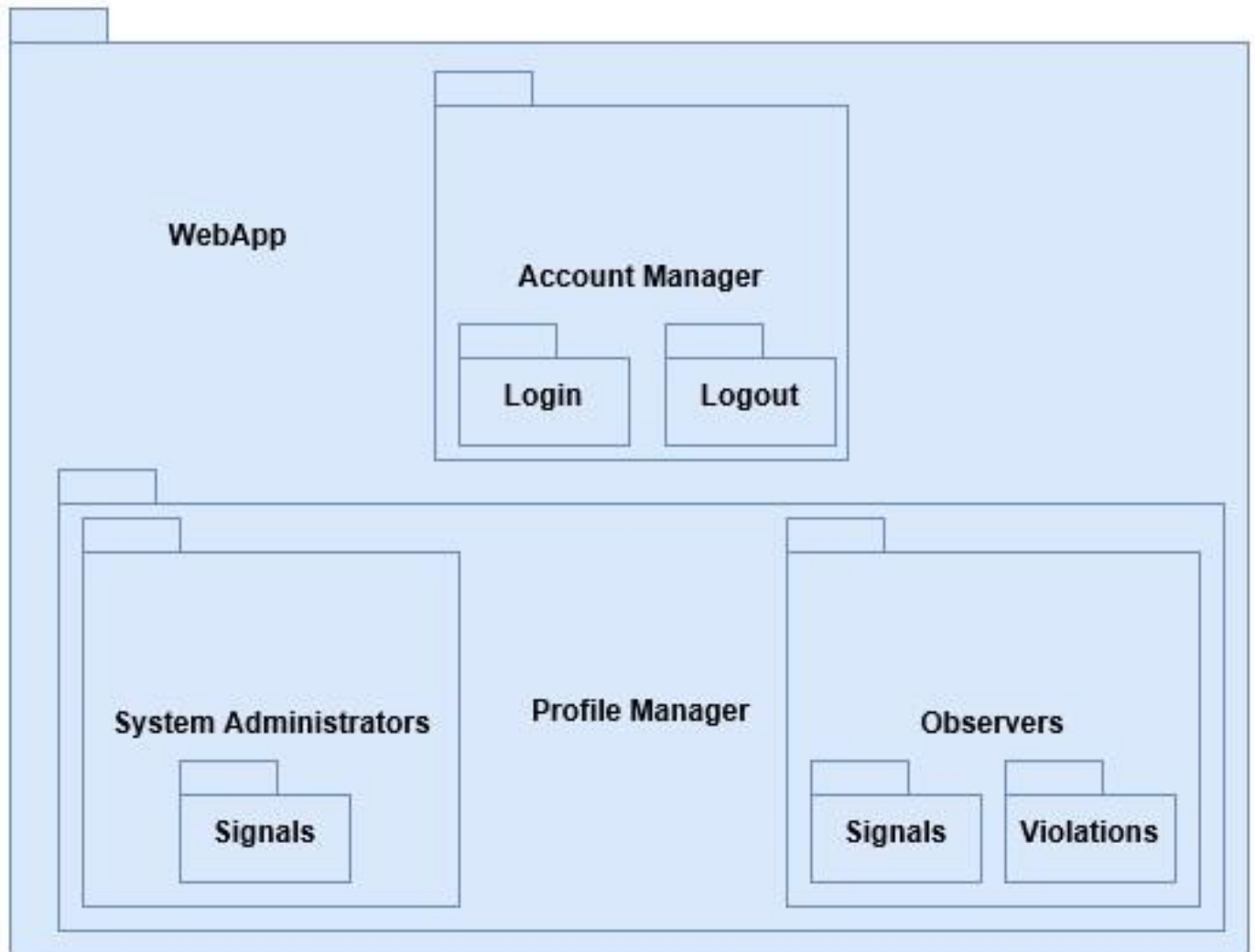
2.5.3.3.1. Connect

public void Connect();

вход: void

изход: MapApps се свързва с модула Computation

2.6. WebApp



2.6.1. Предназначение на модула

Публичен сайт, където са достъпни данните на системата, нарушенията, инциденти и т.н.

2.6.2. Основните отговорности на модула в системата

След автентикация на нарушителя при излизане от рамките на града, загуба на велосипед, инцидент и други проблеми да се праща съобщение на наблюдателите. В допълнение системните администратори имат достъп до системата и гледат да работи правилно, при проблем получават сигнал.

2.6.3. Описание на интерфейсите на модулите

2.6.3.1. Account Manager

2.6.3.1.1. Login

Отговорността на този модул е да предостави подходящ потребителски интерфейс за вход в системата, както и да препрати получената информация към Server модула, където тя се проверява и валидира. Интерфейсът се състои от полета, в които потребителят въвежда нужната информация и бутони, с които да подаде реална заявка към сървъра.

```
public string getCurrentUserName();
public string getHashedPassword();
```

```
private string hash(string pInitialPassword, HashingAlgoorythm pAlgoorythm)
public bool sendLoginRequest(string pUsername, string pPassword)
```

вход: потребителско име и парола

изход: съобщение за резултата от операцията - успешно или неуспешна или успешна, като при неуспех се отчита причината за него (сгрешени данни или неуспешно осъществена връзка със сървъра).

2.6.3.1.2. Logout

Този модул е отговорен за безопасния изход от системата на потребителите. Предоставя на потребителите подходящ интерфейс, чрез който се изпраща заявка към Server модула. Резултатът от тази заявка се следи, като при успех се изчиства информацията за текущия потребител и сайтът се завръща в изходно състояние. В зависимост от настройките на потребителския браузър, може да запази информация в Cookies.

```
public void sendLogoutRequest();
```

вход : void

изход : резултат от операцията - успех/неуспех и при неуспех конкретната причина.

2.6.3.2. Profile manager

Отговаря за разграничаването на отделните лица използващи приложението

2.6.3.2.1. System Administrators

2.6.3.2.1.1. Signals

При проблем със системата, администраторите получават сигнал от сървъра.

```
public void sendSignaltoSA();
```

вход : void

изход : резултат от операцията - успех/неуспех и при неуспех конкретната причина.

2.6.3.2.2. Observers

2.6.3.2.2.1. Signals

Този модул е отговорен за известяване на определените лица със съобщение. Предоставя на потребителя интерфейс, чрез който след въведен код на нарушение, се изпраща заявка към Server модула. ако потребителят е оторизиран да достъпва тази информация заявката се обработва и при успех се показва съобщение за нарушението.

```
public void sendSignaltoOB();
```

вход : void

изход : резултат от операцията - успех/неуспех и при неуспех конкретната причина.

2.6.3.2.2.2. Violations

Отговорността на този модул е да предостави съобщение от нарушение на текущия потребител, ако той е оторизиран да достъпва тази информация. Предоставя интерфейс, чрез който се изпраща заявка към Server модула.

```
public bool requestViolationInfo( string violation );
```

вход: нарушение

изход: съобщение за възникналите грешки.

2.6.4. Други

2.6.4.1. Ограничения при употреба

За да се достъпи сайта, потребителят трябва да има достъп до интернет мрежа и уеб браузър.

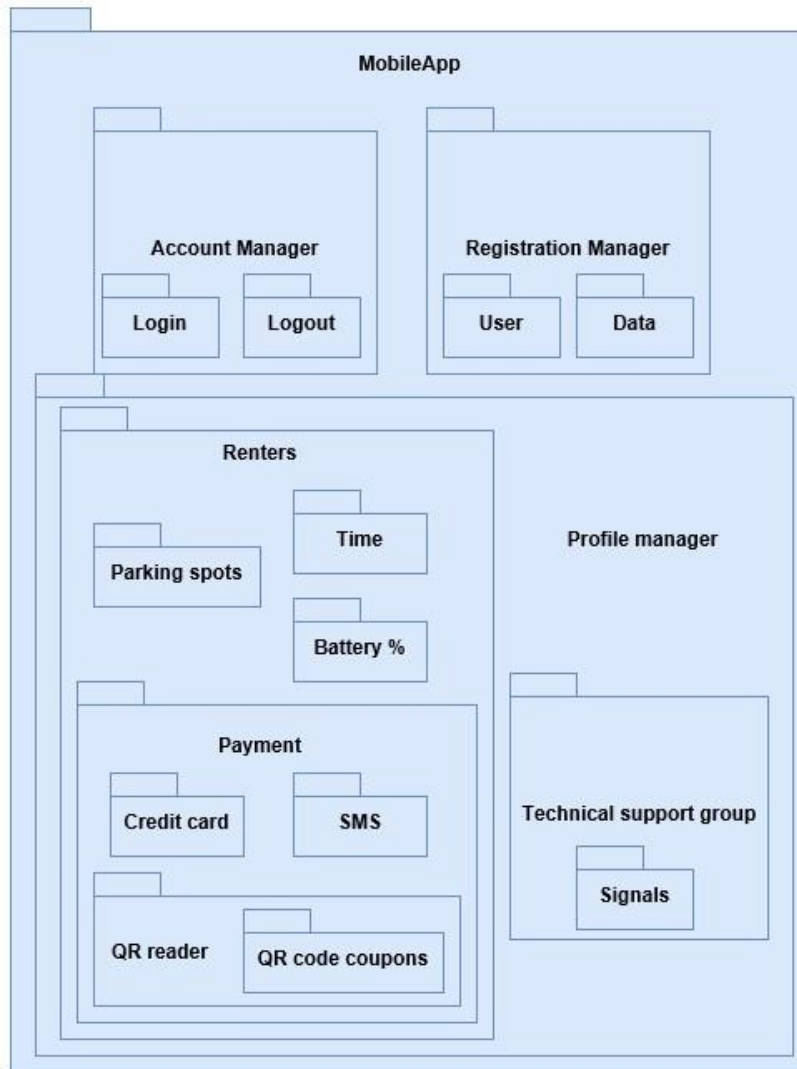
2.6.4.2. Грешки и изключения

Възникналите грешки по време на изпълнение, свързани със заявки от страна на потребителя, се визуализират по подходящ начин за него.

2.6.4.3. Зависимости от други елементи

Този модул зависи изключително от Server модула.

2.7. MobileApp



2.7.1. Предназначение на модула

Мобилното приложение, чрез което потребителите използват системата

2.7.2. Основни отговорности на модула в системата

Осъществява входа на лицата, които използват системата, като им предоставя определени функционалности. Осъществява плащания. Управява потребителските акаунти.

Предоставя информация за свободните места, време и батерия на регистрираните. Чрез

смарт сензори се прави диагностика на велосипедите и информира групата по техническа поддръжка.

2.7.3. Описание на интерфейсите на модулите

2.7.3.1. Account Manager

2.7.3.1.1. Login

Предоставя полета за попълване на необходимите данни и ги изпраща на сървърния модул Access Manager.

2.7.3.1.2. Logout

Изпраща заявка до сървъра за излизането на потребителя от системата изтриване на всякаква локална дата за него.

2.7.3.2. Registration Manager

2.7.3.2.1. User

Предоставя полета за попълване на нужната информация за регистрация на обикновен потребител, и изпраща заявка със съответната информация до сървъра.

2.7.3.2.2. Data

Предоставя полета за попълване на нужната информация за регистрация на член на групата по техническа поддръжка, и изпраща заявка със съответната информация до сървъра.

2.7.3.3. Profile Manager

Отговаря за разграничаването на отделните лица използващи приложението

2.7.3.3.1. Renters

2.7.3.3.1.1. Parking Spots

Показва стоянките със свободни велосипеди в областта, в която се намира. Местата се обновяват от сървъра.

2.7.3.3.1.2. Time

Показва времето, с което разполага наемателя.

2.7.3.3.1.3. Battery %

Показва батерията на велосипеда.

2.7.3.3.1.4. Payment

Този модул се свързва с външна система (кредитна карта, SMS, закупени талони с уникален код). При успешна транзакция и получаване на сумата се изпраща информацията до сървъра. Кодът може да се въвежда ръчно или автоматично.

2.7.3.3.2. Technical Support Group

2.7.3.3.2.1. Signals

При получаване на сигнал от сървъра, известява групата за проблема.

2.7.4. Други

2.7.4.1. Ограничения при употреба

Трябва да има инсталирано приложението, както и връзка със интернет за да може да бъде използвано.

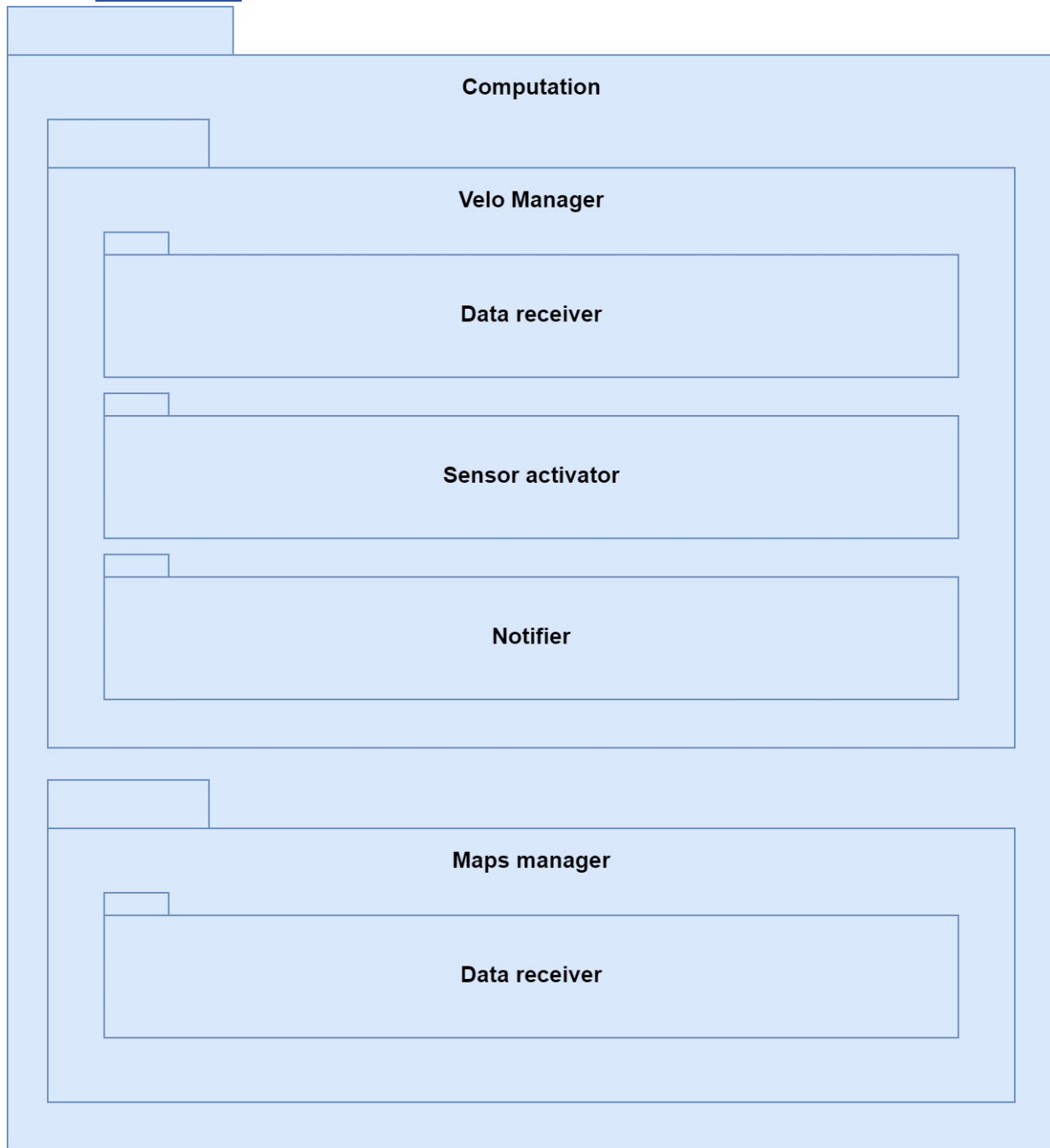
2.7.4.2. Грешки и изключения

При появата на грешка или изключение, да се покаже и извести чрез съобщение по подходящ начин.

2.7.4.3. Зависимости от други елементи

MobileApp модула зависи изключително много от сървър модула. Почти всичко се извършва чрез заявки от и към него.

2.8. Computation



2.8.1. Предназначение на модула

Обработка на данните получени от Velo Manager и външната система за времето.

2.8.2. Основни отговорности на модула в системата

Определя местоположението на грсите. Комуникира с Мар приложенията. Сигнализира при повреди на велосипедите.

2.8.3. Описание на интерфейсите на модулите

2.8.3.1. Velo manager

Основната отговорност на този модул е осъществяването на връзка велосипедите и сензорите им.

2.8.3.1.1. Data receiver

Отговорността на този модул е да получи и обработи информацията от велосипедите. Обработката се състои в преобразуване на локацията им в подходящ формат за изчисление от Maps Manager и за запазване в Database.

public Location[] getLocation(string idVelo);

-вход: идентификационен номер на велосипед

-изход: локацията, получена от gps-ите прикрепени за велосипедите

2.8.3.1.2. Sensor activator

Този модул отговаря за получаване на информация от сензорите за диагностика в даден момент от време.

public void string[] getVeloInfo();

-вход: void

- изход: информация за текущите дата и час, район

2.8.3.1.3. Notifier

Изпраща списък за повредените велосипеди до тиймът по техническа поддръжка

public void string[] getBrokenVelo();

- вход: void

- изход: информация списък за повредените велосипеди

2.8.3.2. Maps manager

Този модул получава информация за картите.

2.8.3.2.1. Data receiver

Този модул получава информация за картите от външна система, обработва тази информация и я изпраща в подходящ вид към Velo Manager.

2.8.4. Други

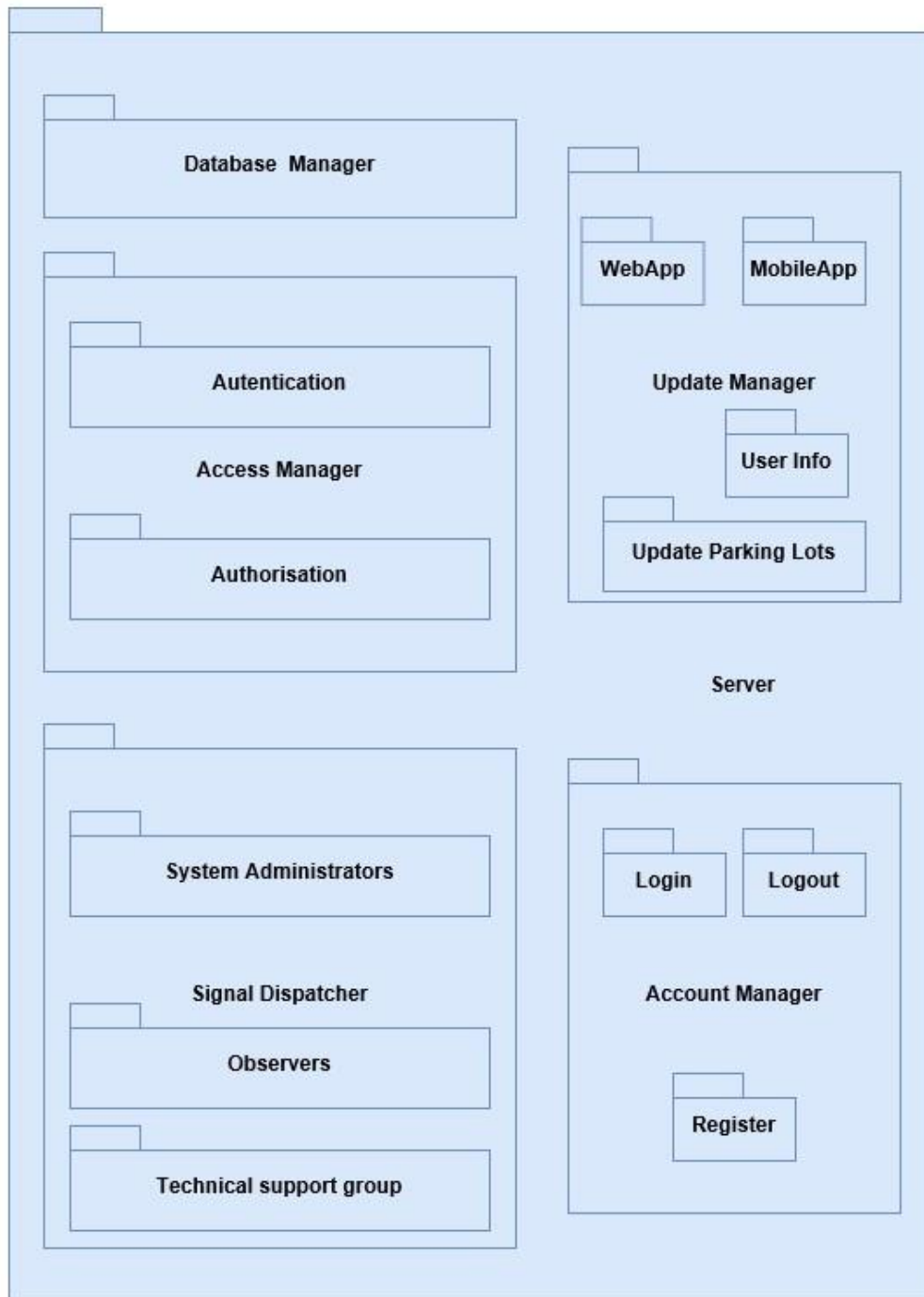
2.8.4.1. Ограничения при употреба

Техническият екип на системата се известява при възникнали грешки по време на изпълнение и взема мерки за отстраняването им.

2.8.4.2. Грешки и изключения

Този модул зависи от VeloManager, Server и от външните системи за карти.

2.9. Server



2.9.1. Предназначение на модула

Този модул осъществява връзката между всички модули в системата.

2.9.2. Основни отговорности на модула в системата

Предава информация за свободните велосипеди на стоянките. Обновява информацията към операторите и групата за техническа поддръжка, свързана със състоянието на системата. Изпълнява основните заявки към базата от данни – пише и извлича информация. Обновява уеб сайта и приложението. Осъществява връзка между приложението и потребителската база данни във връзка с плащанията. Обновява информацията за потребителите.

2.9.3. Описание на интерфейсите на модулите

2.9.3.1. Account manager

Получава заявки и нужната информация от MobileApp и WebApp и я изпраща в Access manager модула.

2.9.3.2. Access Manager

Получава заявки и нужната информация от Account Manager и я изпраща в Database manager модула.

2.9.3.3. Signal Dispatcher

Получава сигнали от Velo Manager и Computation и ги изпраща към съответните потребители.

2.9.3.4. Update Manager

Актуализира съответните приложения и информация, която е нужна, за да се изпълняват функционалните изисквания.

2.9.3.5. Database manager

Изпраща цялата събрана информация към Database модула.

3. Описание на допълнителните структури

3.1 Структура на процесите

3.1.1 Първично представяне

Структурата на процесите е насочена главно към потребителите и към други заинтересовани лица без технически знания. Основните процеси в нашата система са процесите на: обновяване на местоположенията на велосипедите, които потребителят може да види; наемане на велосипед; подаване на сигнал при неизправност на велосипеда.

Тези три процеса използват почти всички модули в системата и са лесен начин да се придобие добра представа за това как работи тя. Още един плюс на нашият избор за допълнителна структура е това, че може да се види последователността на протичане на процесите.

3.1.2 Наемане на велосипед



3.1.2.1 Описанието на елементите и връзките

Наемането на велосипед се осъществява от потребителя чрез MobileApp. Потребителя може да наеме велосипед при отварянето на приложението и като задейства съответен механизъм. Изпраща се заявка до Velo Manager за проверка дали има свободни велосипед. Ако няма, се показва съобщение за грешка. Ако пък има, системата се обръща към Parking spots и показва най-близко намиращите се велосипеди. Ако е избран велосипед, системата се обръща към Update Parking Spots. Потребителят се свързва с избраната външна система за плащане. След това проверяваме дали е успешна транзакцията. При неуспех се прекратява плащането. При успех се изпраща заявка към Velo Manager(Computation) за осъществено плащане. Всичко се запазва в Базата от данни чрез Database Manager. Процеса завършва успешно.

3.1.2.2 Описание на обкръжението

Наемането на велосипед се осъществява чрез външна система за плащане.

3.1.2.3 Описание на възможните вариации

Може да се интегрират различни външни системи за плащане.

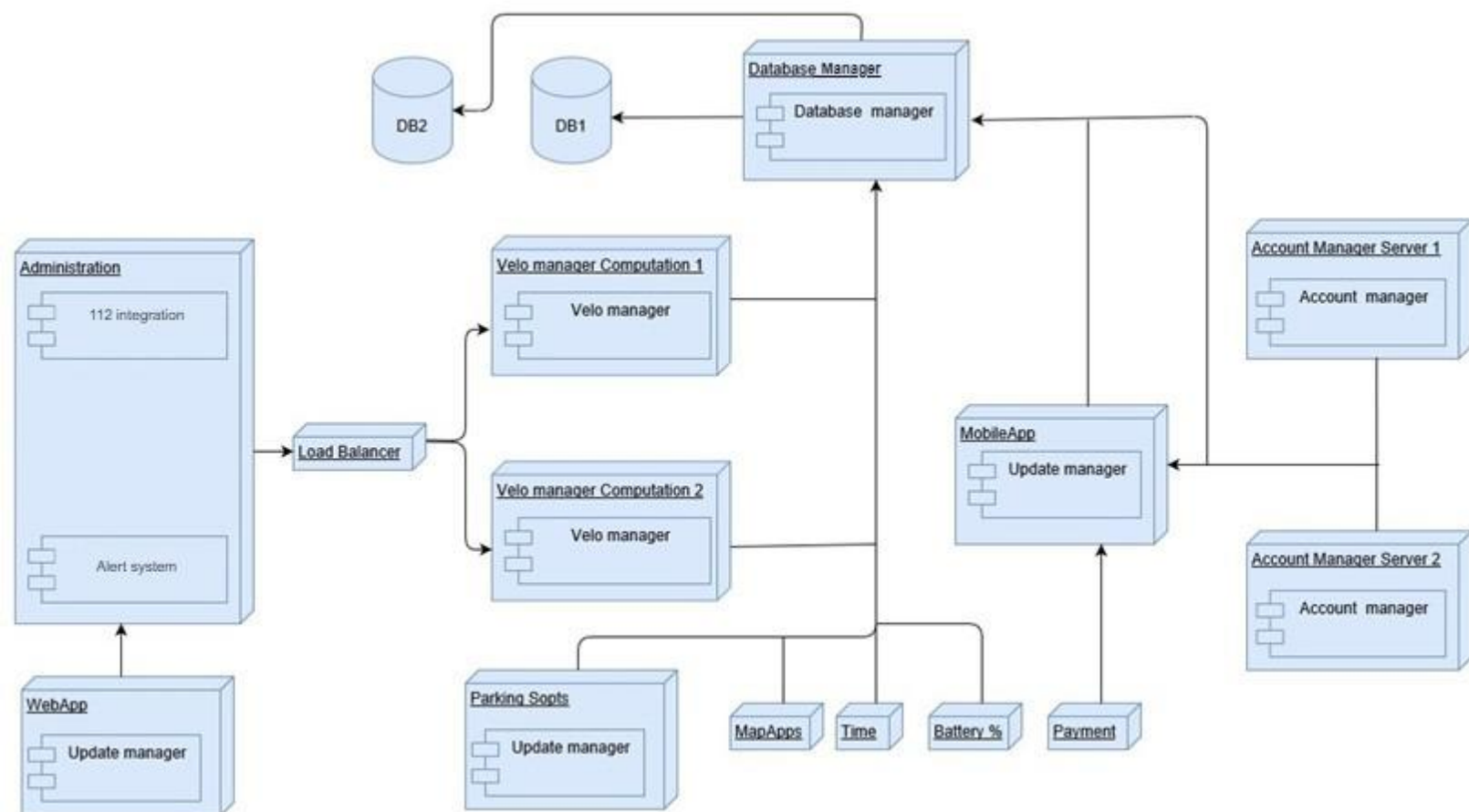
3.2 Структура на внедряването

3.2.1 Първично представяне

Структурата на внедряването е изключително важна за VeloCity, защото всеки от модулите му са върху отделно устройство. За да е надеждна системата и да е достъпна в пикове часове, по-важните подмодули са на отделни сървъри.

Системата трябва да е устойчива към пикови натоварвания в най-натоварените в денонощието, часове за придвижване – от това изискване следва, че Server, Computation и Database трябва да имат backup сървъри в случай на претоварване или повреда на основния, за да може да продължи нормалното функциониране на системата.

При или загуба на връзка с даден велосипед, трябва да се сигнализира наблюдателя в рамките на 10 сек, като се изпратят данни за движението на велосипеда в последните 30 мин, заедно с най-вероятната му позиция на която се е намирал в момента на изгубване на връзката – от тук следва, че информацията предадена от Velo Manager на Computation трябва да е изчислена в рамките на секунди, за да бъде известен наблюдателя и да предприеме мерки. Signal Manager не трябва да е на един сървър с Velo Manager, въпреки, че е негов подмодул, заради голямото количество информация, която трябва да бъде обработена и изпратена.



3.2.2 Описанието на елементите и връзките

Payment е интерфейс за плащане, който независимо от начина на плащане е еднакъв към Mobile App.

MapApps е сървър, който е един интерфейс за работа с всички външни карти, които са свързани със системата.

Database Manager е сървърът, който разпределя заявките към двете отделни бази данни.

Той контролира получените заявки и балансира натоварването на сървърите с бази данни и когато друг модул от системата се нуждае от информация от базата данни изпраща заявка към този модул.

DB1 и DB2 а сървърите с копие на базата данни. Два са , за да могат да отговарят на голямо натоварване. Обработват директни заявки единствено от Database Manager. Външния свят няма директен достъп с тях, защото се намират на локална мрежа.

Account Manager е сървърът , който обработва профилите на всички потребители, затова са две инстанции на този сървър, за да може да издържи на пикови натоварвания.

Mobile App са всички отделни смартфони, които имат приложението на VeloCity. Хардуерът на устройствата може да е Adroid/IOS и не е от значение за системата. Този модул е връзката между различните сървъри в системата, за да може потребителите да използват цялата функционалност на приложението.

VeloManager е вградена система на всяко колело, както и софтуера. Поддържа всички сензори и GPS-и и изпраща информация от тях на другите модули. Също така през тези два сървъра минава цялата обработка на информация за велосипедите, търсенето на такива и тн. Очаква се голямо натоварване и поради тази причина са два и преди тях има load balancer, който разпределя заявките между двете инстанции.

Administration сървърът не обработва голямо количество заявки, затова е един и към него се изпращат съобщенията за проблеми и заявките от WebApp - от администраторите и техническия екип.

3.2.3 Описание на обкръжението

Payment Server е свързан с множество външни системи и комуникира с тях по определен от тях начин.

MapApps сървърът са различните видове карти, които могат да се ползват през системата и е свързан с външните системи, които са услугите на тези карти.

3.2.4 Описание на възможните вариации

Възможно е DatabaseManager да е интегриран отделно във всеки от двата DB, но тогава ще има нужда от Load Balancer на негово място.

4. Архитектурна обосновка

В тази секция ще представим основните архитектурни драйвери.

- **Велосипедите се намират на стоянки, разположени на предварително определени позиции в рамките на града. Батериите на велосипедите се зареждат по време на престоя им на стоянката.**

Смятаме, че това е архитектурен драйвер, понеже е най-основното и най-важно функционално качество. Около него ще се върти цялата функционалност на системата (какво прави системата).

- **Системата поддържа следните групи потребители: а. Наемател на велосипед (обикновен потребител) б. Член на група по техническа поддръжка на велосипедите с. Системен администратор (техническа софтуерна поддръжка) d. Наблюдател/отговорник по използването на велосипедите**

Важно е, защото показва всички лица, които ще използват системата. Също така, обособява основните им роли и правата, които те ще имат когато я използват.

- **При заявка от потребителя за търсене на велосипед, приложението му показва най-близката стоянка, където има свободен велосипед с поне 60% заредена батерия. 60 е конфигурационен параметър на системата.**

Трябва да се гарантира, че системата ще може да изпълни това изискване – тоест ще може да намери най-близката стоянка със свободен велосипед до потребителя стига да е зареден с поне 60%. Това е възможно, защото велосипедът може да се зарежда и механично чрез педалите.

- **Всеки велосипед има уникален идентификационен номер в системата и е снабден с GPS устройство, както и със смарт-сензори за самодиагностика. При наличие на технически проблем по велосипеда (спукана/спаднала гума, повреда, и т.н.) да се изпраща известие до групите по техническа поддръжка, които в рамките на половин час трябва да диагностицират повредата и да вземат мерки за отстраняването ѝ**
- **При или загуба на връзка с даден велосипед, трябва да се сигнализира наблюдателя в рамките на 10 сек, като се изпратят данни за движението на велосипеда в последните 30 мин, заедно с най-вероятната му позиция на която се е намирал в момента на изгубване на връзката. Т.нар. най-вероятна позиция се определя със специален алгоритъм (напр. чрез изкуствен интелект)**

Смятаме, че това са архитектурни драйвери, защото са доста важни за системата. От тях зависи дали системата ще може да се използва. Без велосипедите да бъдат в изправност и без да могат да бъдат намерени, особено ако често се случва, това намалява шанса да се използва приложението.

- **Потребителите може да заплащат услугата чрез кредитна карта, СМС или чрез предварително закупени талони, които съдържат уникален код. Кодът може да се въвежда ръчно или автоматично (QR-code).**

Това е архитектурен драйвер, понеже показва какви са основните услуги, които предлага системата на потребителите, това също така е основен източник на пари и съответно това ще е причината системата да продължи да съществува, да се развива и да помага на потребителите със своите услуги.

- **Наемателите на велосипед се регистрират през мобилното приложение, като в профила им се включват следните лични данни: имена, ЕГН, както и данни за връзка.**
- **Системните администратори и наблюдателите използват системата през Уеб приложение**

Тези са ключови за системата, показващи как системата ще бъде използвана от наемателите – чрез мобилно приложение и системните администратори – чрез уеб приложение.

- **Системата трябва да може да се интегрира с всички познати онлайн услуги за географски карти (Google maps, BG maps, Open Street maps и т.н.), като има възможност за бъдещо добавяне на нови карти.**

Това е драйвер, понеже помага на потребителите да намират по-лесно стоянките за велосипеди, да се ориентират и да следят да не излязат извън зоната за използването им. Без тези карти става много по-трудно и може да доведе до недоволни потребители.

- **Системата трябва да е устойчива към пикови натоварвания в най-натоварените в денонощието, часове за придвижване.**

Това е архитектурен драйвер, защото системата трябва да работи в основните часове, когато потребителите най-много ще я използват и ще разчитат на нея. Защото може да доведе до недоволни потребители, дори и до загубата им. Затова трябва системата да се проектира, така че в тези часове да няма проблеми и да е налична през по-голямата част от времето.