

Курсова работа

Софтуерни архитектури и разработка на софтуер

VeloCity

**Изготвено от: Кристияна Николова 62372,
Евтим Иванов 62418**

1. Въведение

1.1 Обща информация за текущия документ

1.1.1 Предназначение на документа

Предназначението на документа е да се представи архитектурата на системата за споделени пътувания, с цел по-лесното и разбиране.

1.1.2 Описание на използваните структури на архитектурата.

1.1.2.1 Декомпозиция на модулите

Декомпозицията на модулите показва системата разделена на подмодули, като това помага за лесното разбиране на това коя функционалност в кой модул се намира, както и къде точно се намира в системата. Модулите биват DB, VeloCity Web, VeloCity Server, Velocity Mobile. DB съхранява цялата база от данни за системата. VeloCity Web и VeloCity Mobile са интерфейсите на потребителите използващи системата, в зависимост от това с какви роли са влезли, като системния администратор, поддръжката и наблюдателите влизат през уеб приложението, а обикновените потребители през мобилното приложение. VeloCity Server е връзката между базата от данни и уеб или мобилното приложение.

1.1.2.2 Структура на внедряването

Чрез структурата на внедряването лесно се разбира разпределението на хардуерните компоненти в цялостната система.

1.1.2.3 Структура на процесите

Нагледно се вижда изпълнението на процесите в системата, както и последователността на тяхното изпълнение.

1.1.3 Структура на документа

Документът се състои от 4 секции:

Секция 1: Въведение в документа

Секция 2: Съдържа структура от тип “Декомпозиция на модулите”

Секция 3: Съдържа допълнителни структури от тип “Структура на процесите” и “Структура на внедряването”.

Секция 4: Обосновка на архитектурата, архитектурните драйвери както и причините за тяхното избиране.

1.2 Общи сведения за системата

VeloCity е приложение за лесно наемане на електрически велосипеди в рамките на определен град. В приложението служителите използват приложението през уеб сайт, а потребителите през мобилното приложение. Уеб приложението е разработено с цел подпомагане на служителите. Чрез уеб приложението те могат да следят диагностиката на всеки велосипед, неговия бивш маршрут, къде се намира в момента, а също така и да получават нотификации за случили се пътни инциденти по време на използването на услугата. Всеки потребител може да търси най-близкият

до него велосипед със зададена от него поне X% батерия, могат да проверяват оставащите проценти батерия на велосипеда, също така и да заплащат за дадената услуга чрез СМС, талон или кредитна карта. Всеки велосипед има смарт сензори за самодиагностика, а също така и батериите на всеки от велосипедите се зарежда след оставянето му на определените места (стоянки).

1.3 Терминологичен речник

талон - документ с код върху себе си, чрез който може да се плати използването на дадената услуга чрез сканиране или ръчно въвеждане

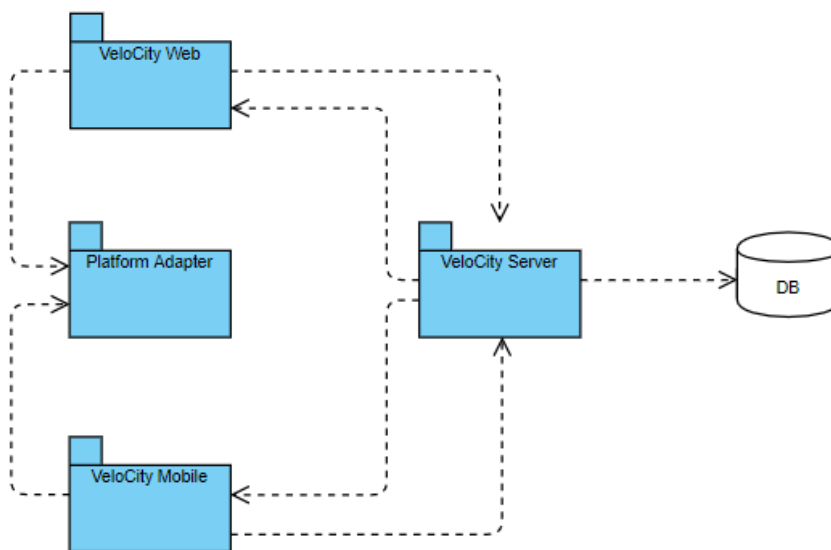
смарт-сензор - устройство, което изпраща сигнали за диагностиката на велосипеда в момента

нотификация - съобщение

2. Декомпозиция на модулите

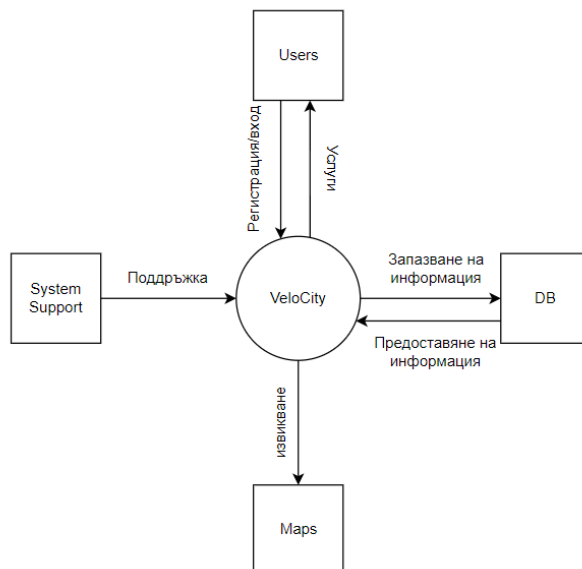
Общ вид на декомпозицията на модули за системата

Елементите на декомпозицията на модулите са мобилното приложение, уеб приложението, сървърът, чрез който системната комуникация е възможна, платформеният адаптер и базата данни.



Контекстна диаграма

Това е т. нар. описание на обкръжението. Съдържа информация за това как описаните тук модули си взаимодействат с други системи или подсистеми или модули, интерфейси, протоколи и т. н.



Подробно описание на всеки модул

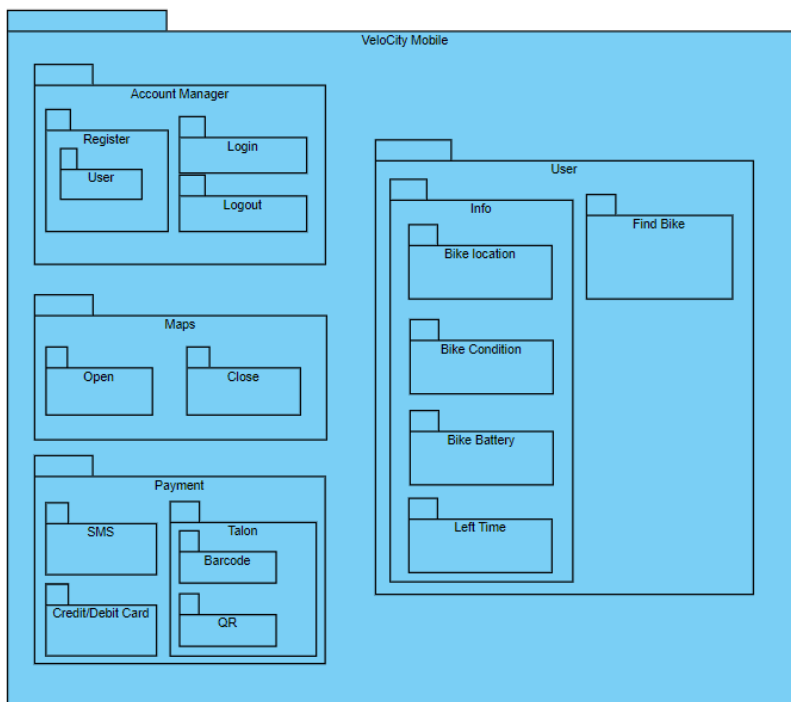
Предназначение на модула

Основни отговорности на модула в системата

Описание на интерфейсите на модула ⁴. Описанието на всеки интерфейс съдържа:

- Идентификация (име)
- Предоставени ресурси (методи)
 - Синтаксис, включително използвани типове данни
 - Входни данни
 - Изходни данни
 - Семантика
- Други (ако е необходимо)
 - Ограничения при употребата
 - Грешки и изключения
 - Зависимости от други елементи
 - Пример за употреба

VeloCity мобилно приложение:



- 1. Предназначение на модула** - чрез мобилното приложение потребителите получават достъп до всички функционалности на системата. Тези функционалности и техните характеристики са представени в модула VeloCity Mobile.
- 2. Основни отговорности на модула в системата** - Модулът трябва максимално да улесни своите потребители, които използват системата. Това трябва да стане с подходящ интерфейс, който е лесен за разбиране и използване. Всички функционалности трябва да са представени по подходящ начин.
- 3. Описание на интерфейсите на модула:**

3.1. Account Manager - представя функционалността за възможност за вход и регистрация в приложението.

3.1.1. Регистрация - форма за регистрация, в която потребителят предоставя задължително своето име, ЕГН и данни за обратна връзка. При коректно въведени данни, системата изпраща заявка към сървъра за добавяне на съответния потребител към базата данни на системата.

bool checkName(string name) - проверява дали подаденото име е коректно, т.е. отговаря на формата "<собствено име> <фамилия>" и съдържа само букви.

void setName(string name) - записва подаденото име

void setUsername(string uname) - записва подадения username

bool checkEGN(string EGN) - проверява дали ЕГН-то е във валиден формат

bool checkPhoneNumber(string phoneNum) - прави проверка дали въведеният телефонен номер е във валиден формат

bool createNewUser(string name, string uname, string EGN, string phoneNum) - прави проверка в себе си дали името, ЕГН-то и телефонният номер са валидни. При валидни входни данни и успешно добавяне на новия потребител към системата, функцията връща положителен резултат.

3.1.1.1 User - подмодул, който включва в себе си само регистрираните наематели на велосипеди.

3.1.2. Login - форма за влизане в системата, която очаква въвеждане на валидни входни данни на вече съществуващ потребител.

3.1.3. Logout - възможност за излизане от текущия акаунт

3.2. User - модул, който включва в себе си всички функционалности, които един регистриран потребител може да използва.

3.2.1. Info - връща информация относно текущия потребител - показва наетия от него велосипед, неговото състояние, нивото на батерията, както и оставащото време за ползване.

3.2.1.1. Bike Location - предоставя информация за текущите координати на велосипеда и изобразява неговото местоположение на картата.

3.2.1.2. Bike Condition - предоставя информация за техническата изправност на превозното средство чрез вградения сензор в него. Потребителят може да види състоянието на спирачките, гумите, както и да получи информация за някаква евентуална повреда.

3.2.1.3. Bike Battery - предоставя информация в проценти за нивото на батерията на велосипеда, който в момента се използва от потребителя.

3.2.1.4. Left Time - предоставя информация на потребителя за оставащото му време за използване на велосипеда.

`Time getLeftTime()` - връща информация за оставащото време във формат чч:мм:сс

3.2.2. Find Bike - потребителят подава цяло число X, след което системата показва на картата най-близкия велосипед с ниво на батерията, което е равно или по-голямо от числото X, въведено от потребителя.

`Coordinates findBike(int X)` - връща координатите на най-близкия велосипед, които да покажат местоположението му на картата.

3.3. Maps - модул, който включва в себе си функционалности, които дават възможност за използване на картите, предоставени от външни услуги

3.3.1. Open

3.3.2. Close

3.4. Payment - модул, който включва в себе си всички функционалности, които позволяват на потребителя да заплати чрез най-удобния за него начин.

3.4.1. SMS - когато потребителят избере този начин на плащане, таксата от наетия велосипед се начислява на сметката му към мобилния оператор.

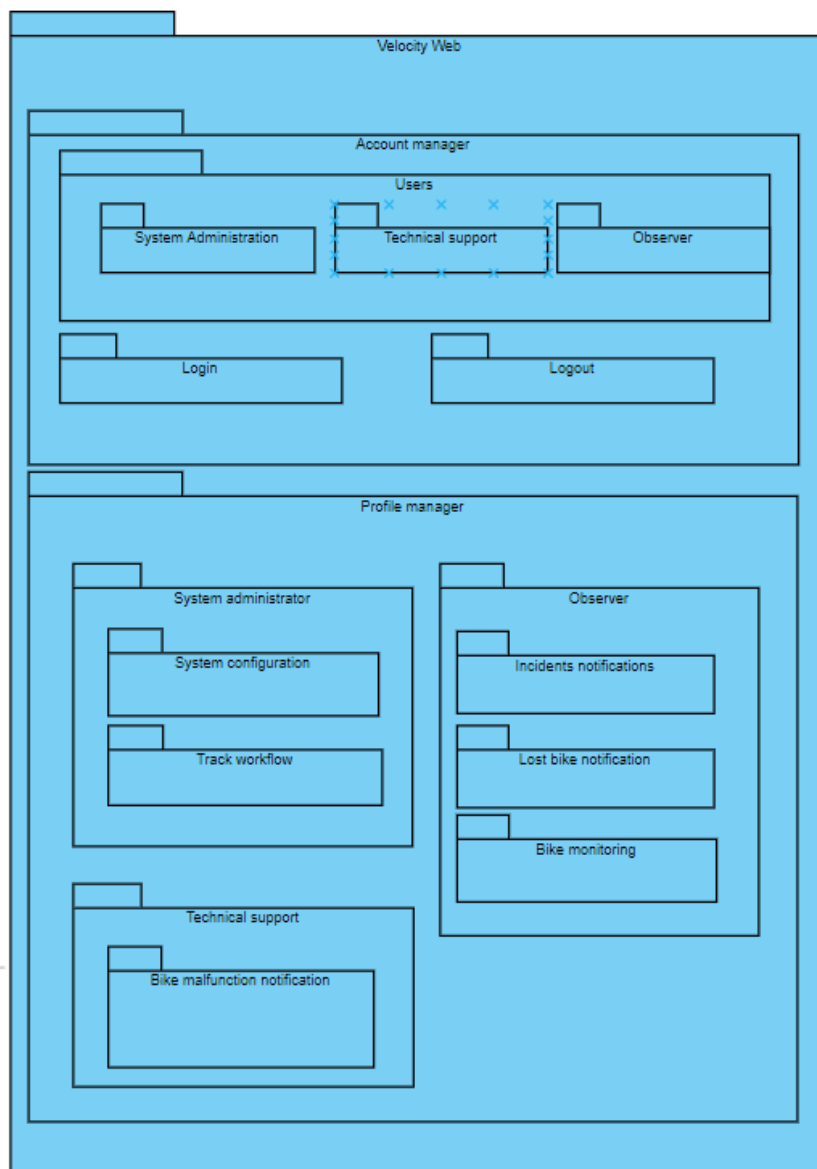
3.4.2. Credit/Debit Card - при избиране на този начин на плащане, системата предоставя форма, в която потребителят се подканва да въведе необходимата информация - номер на картата, име на картодържача, дата на валидност, CVV/CVC код. След въвеждането на данните се изпраща заявка към сървъра за осъществяване на банковата транзакция, която следва да бъде обработена.

3.4.3. Talon

3.4.3.1. QR - предоставя възможност за плащане чрез сканиране на талон с QR код, след използването му, талонът не може да бъде използван повторно.

3.4.3.2. Barcode - предоставя възможност за плащане чрез ръчно въвеждане на кода от талона, след което ваучерът не може да бъде използван повторно.

VeloCity уеб приложение:



1.Предназначение на модула - чрез него работата на хората, поддържащи приложението е обединена.

2.Основни отговорности на модула в системата - модулът трябва максимално да улесни наблюдателите, системния администратор и техническата поддръжка, така че максимално бързо да може да се усвои и използва.

3.Описание на интерфейсите на модула:

3.1 Account manager - Осигурява вход и изход в системата за потребителите

3.1.1 Users - след коректно въведени данни, модулът позволява достъп до функционалностите на една от групите, в зависимост от въведените данни:

3.1.1.1.Observer

3.1.1.2 System Administrator

3.1.1.3 Technical support

3.1.2 Login - предоставя полета за попълване на данни от потребителя, като след коректно въвеждане, потребителят има достъп до функционалности единствено до неговата област(системен администратор,наблюдател и т.н.)

3.1.3 Logout - излиза от потребителския акаунт.

3.2 Profile manager - преустановява различните функционалности между различните служители.

3.2.1 System Administrator - регламентират се дейностите на системния администратор, както и функционалностите до които той е достъпен

3.2.1.1 System configuration - позволява на системния администратор лесно да поддържа системата в изправност

3.2.1.2 Track workflow - проследява натовареността на системата, както и коректността на работата и

3.2.2 Observer - следи за пътно-транспортни произшествия, както и при крайни случаи има достъп до данните на служителите, с цел по-бързо установяване на местоположението на потребителя

3.2.2.1 Incident notification - изпраща извештие за инцидент свързан с даден потребител на наблюдателя

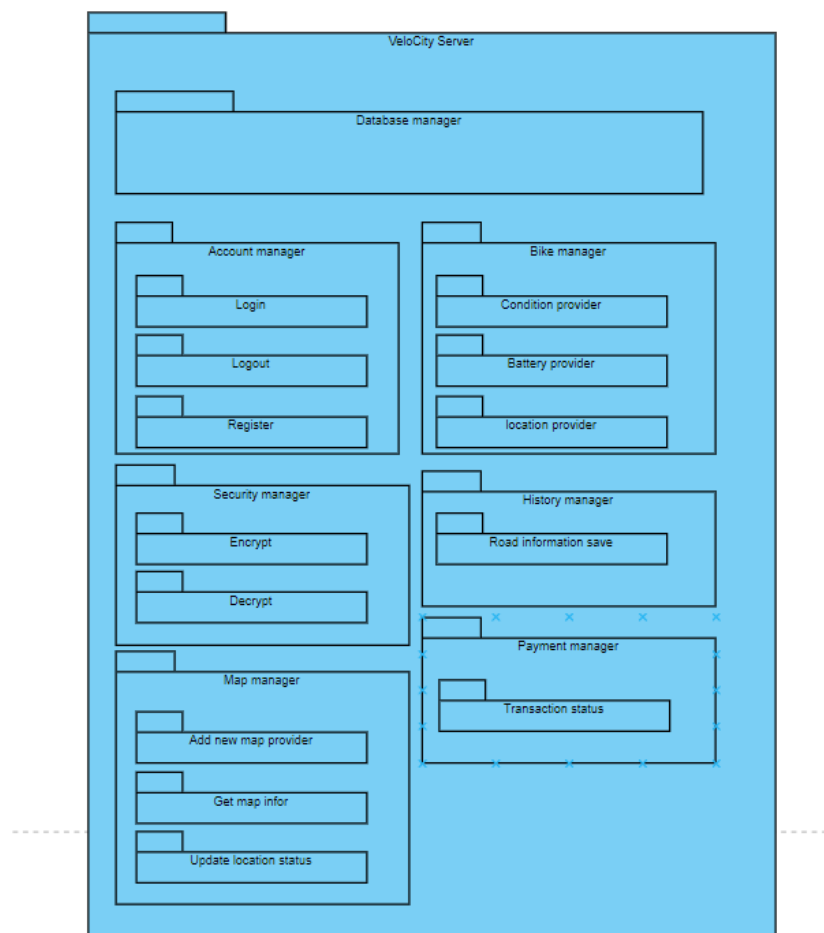
3.2.2.2 Lost bike notification - изпраща известие на наблюдателя за загубено колело или за колело излязло извън позволените предели за ползване

3.2.2.3 Bike monitoring - позволява наблюдение върху използваните колела, както и тяхното местоположение в конкретен период от време

3.2.3 Technical support

3.2.3.1 Bike malfunction notification - изпраща известие до екипа по поддръжка за проблеми с колелото(като спукана гума,повреда и т.н.)

VeloCity Server:



1.Предназначение на модула - явява се като връзка между базата данни и уеб приложението или мобилното приложение.В него се съхранява основната логика на системата.

2.Основни отговорности на модула в системата - модула приема заявки от потребителите, като при нужда чете информация от базата данни,записва или трие от нея.

3.1 VeloCity Server

3.1.1 Database manager- отговаря за връзката между различните модули и базата данни

3.1.2. Account manager - управлява входа и изхода в системата за потребителите

3.1.2.1 Login - проверява дали съществува двойка (потребител,парола), който да отговарят на въведените

```
public boolean CheckAccount(string username, string password)
```

3.1.2.2 Logout - премахва правата за достъп на потребителя(премахване на token)

```
public void Logout()
```

3.1.2.3 Register - проверява дали потребител със същото име не е регистриран вече. Ако няма потребител регистриран със същото име регистрира нов такъв, ако има връща грешка за съществуващ вече такъв потребител.

```
public void Register (string username, string password)
```

```
public boolean CheckIfAlreadyNameExists(string username)
```

3.2 Bike manager

3.2.1 Condition provider- осигурява информация за състоянието на велосипедите, като има добро техническо състояние, а при неизправност, бива разделено на различни подкатегории, за максимално улеснение на техническия екип

```
public String GetConditionOfBike()
```

3.2.2 Battery provider - осигурява информация за % запълнена батерия на велосипедите

```
public int GetBatteryLevel() - връща число между 0- 100 %, в зависимост от заряда на батерията
```

3.2.3 Location provider - при подаване на заявка, осигурява информация за местоположението на велосипеда в даден момент от време

```
public Location GetLocation() - връща обекто от класа Location , с координати (x,y) -
```

координатите на велосипеда в даден момент от време

3.3 Security manager- този модул играе роля за криптиране на личните данни на потребителите, за да могат да бъдат максимално защитени

3.3.1 Encrypt- криптира данните на потребителя (дебитна карта, номер и т.н.)

```
public String Encrypt(String DataToBeEncrypted) - връща криптирани подадените данни
```

3.3.2 Decrypt - декриптира вече криптираните данни на потребителя при нужда за заплащане и т.н.

```
public String Decrypt (String DataToBeDecrypted) - връща декриптирани подадените данни
```

3.4 History manager- модул за запазване на маршрутите изминати от велосипедите при даден потребител

3.4.1 Road information save - запазва данните за всеки потребителски маршрут в продължение на 1 седмица, след което те автоматично биват изтрети от базата данни

```
public void SaveRoadInfo()
```

3.5 Transaction manager - модул отговарящ за правилното обработване на картовите транзакции от потребителя към системата

3.5.1 Transaction status - връща статуса на дадена транзакция, по подадено ID

```
public String GetTransactionStatus(String TransactionID)
```

3.6 Map manager - модул който спомага за връзката между различните географски карти, които потребителя използва и дадената система

3.6.1 Add new map provider - позволява да се добави нова система за управление на географски карти към вече съществуващите

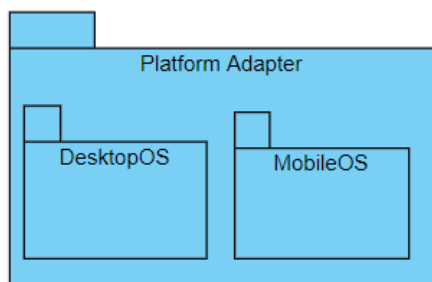
```
public void AddNewMapProvider(Map MapProviderName) - добавя обект от тип Map към системата
```

3.6.2 Get map info - връща информация за това къде точно се намира велосипеда на географската карта по координати (x,y)

```
public Location GetMapInfo()
```

3.6.3 Update location status - модул за постоянно актуализиране на местоположението на потребителите. На всеки 10 секунди се изпраща заявка за актуализиране на местоположението на всички потребители
 public void UpdateLocationStatus(Location UserLocation) - актуализира локацията на потребителя.

Platform Adapter:



1. **Предназначение на модула и основни отговорности на модула в системата** - модулът отговаря за осигуряването на хомогенност и обобщеност от гледна точка на системния интерфейс, така че потребители от различни операционни системи да имат достъп.
2. **Описание на интерфейсите на модула**
 - 2.1. **DesktopOS** - Windows 7, Windows 8, Windows 10, Linux, MacOS X
 - 2.2. **MobileOS** - Android 8.0 or later, iOS 10 or later
 - 2.3. **Други**
 - 2.3.1. **Зависимости от други елементи** - модулът е зависим от операционната система. При излизане на нови версии на дадената операционна система, необходимо е да се направят подходящите промени, така че приложението да работи и на по-новите версии.

Описание на възможните вариации (ако е необходимо)

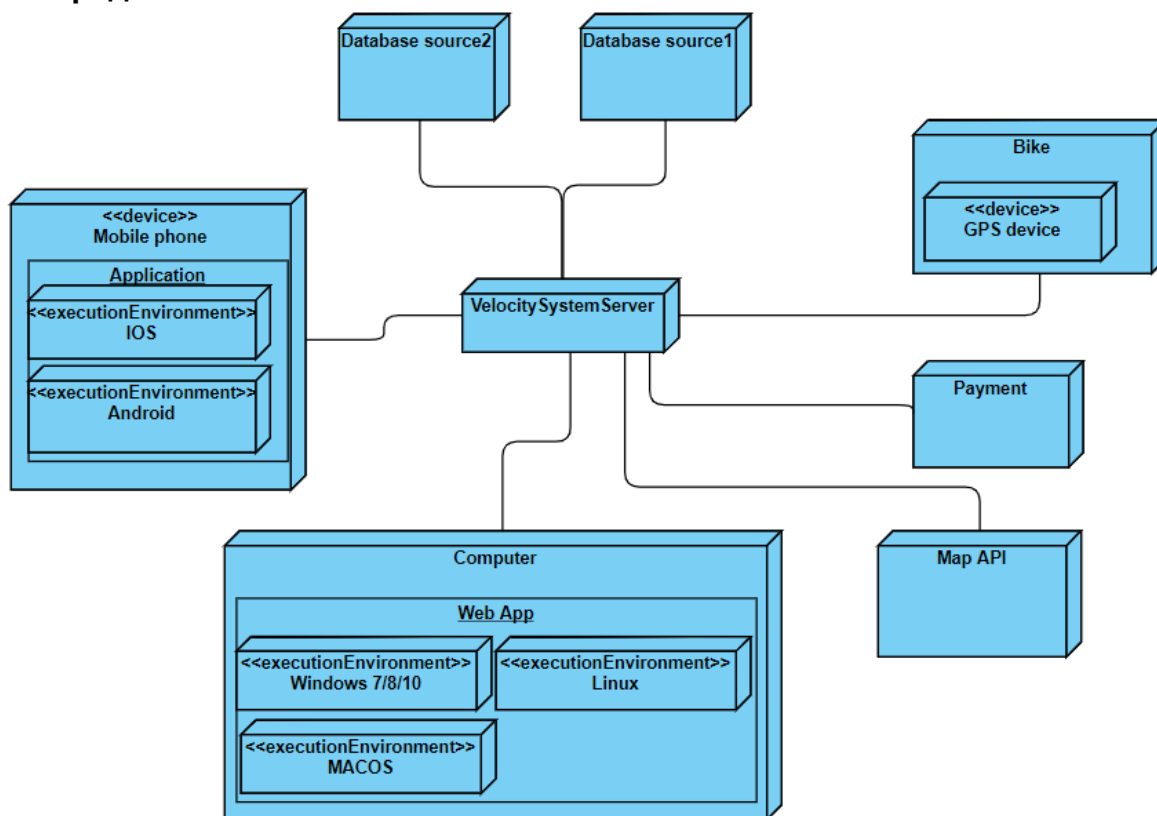
Описват се възможностите за промени на някои от модулите, например: възможности за параметризация на даден модул; условия за избор на протоколи за обмен на информация и др.

3. Описание на допълнителните структури

3.1. Структура на внедряването

Тъй като приложението VeloCity е силно зависимо от огромна част хардуерни компоненти, за ефикасното представяне на такъв тип системата, зависеща от голям брой хардуерни компоненти е нужна структура на внедряването. По този начин ще бъде по-лесно възприето разпределението на хардуерните компоненти в цялата система.

3.1.1 Представяне



3.1.2 Описание на елементите и връзките

Database source1/2 са сървърни копия на които се запазват всичките данни за системата. Тъй като системата трябва да е 99.999% налична, наличието на копие на сървъра позволява да се избегнат и предотвратят крашове на системата.

VelocitySystemServer е сървърът, върху който е разположена цялата логика на системата. Той бива отговорен за отговор на заявките както от потребители, така и от системни администратори и техническия отдел.

Mobile phone е отделно физическо устройство, което позволява достъпа на потребителите до системата. Мобилните версии на системата са според операционните системи, а те биват Android или iOS.

Computer е физическото устройство, позволяващ достъпа на наблюдателите, администраторите и техническия отдел. Достъпа става през уеб приложение, което бива поддържано на операционните системи MACOS, LINUX и Windows 7/8/10.

Bike е физическо устройство, в което има GPS device, който постоянно изпраща данни на сървъра за местоположението на потребителя използващ устройството. Данните биват запазени за определен срок от време, за да може при евентуално загубване маршрута на потребителя да бъде запазен, с цел по-лесно намиране на потребителя.

Payment е системата за разплащане с която потребителя се свързва. Тя поддържа различни видове разплащания - SMS, Credit card, talons.

3.1.3 Описание на обкръжението

Payment бива свързан по различен начин с различните системи, при СМС с различни мобилни оператори а при Credit card с различни банкови институции

Map API е връзката между различните системи за карти и сървъра

3.1.4 Описание на възможните вариации

При нуждата на добавяне на друг онлайн източник на географски карти се използва Map API, което помага за лесното интегриране.

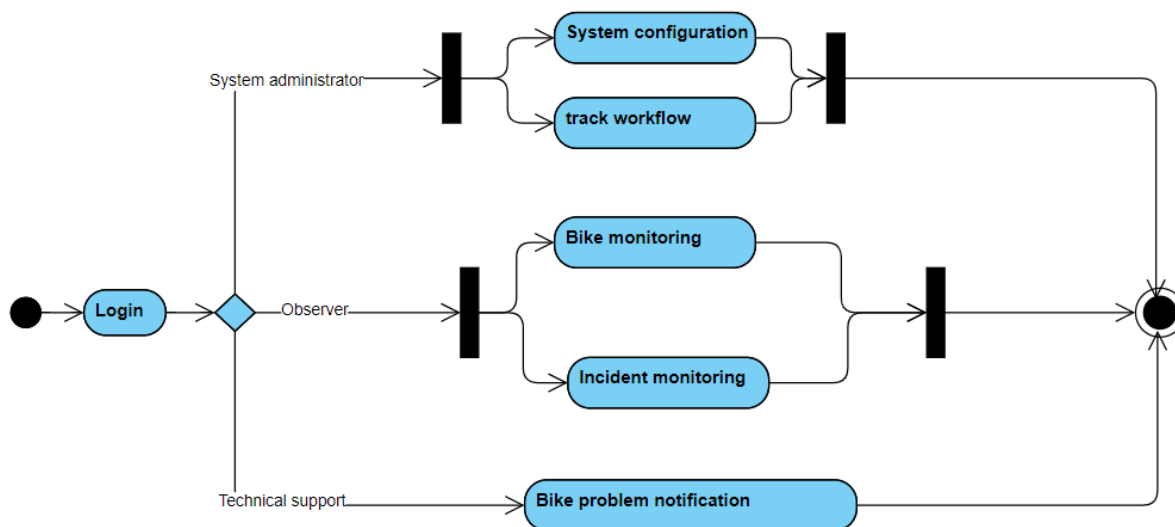
При огромно нарастване на потребителите е възможно добавяне на допълнителни сървъри, за да може наличието на 99.999% да бъде спазено, като сървърите трябва да бъдат синхронизирани.

3.2 Структура на процесите

Структурата на процесите ни показва различните взаимодействия между всеки отделен процес, тяхната последователност и начин на изпълнение. По този начин придобиваме ясна представа за работата на системата в различни ситуации при различните потребители.

3.2.1 Структура на процесите на VeloCity Web

3.2.1.1 Представяне



3.2.1.2 Описание на елементите и връзките

Достъп до уеб приложението на VeloCity имат всички, които имат вече съществуващ профил, който е добавен от системния администратор. Когато потребителят влезе в системата със своя профил, той може да изпълнява специфични дейности, спрямо заемащата от него длъжност. Системният администратор отговаря за конфигурирането на системата и проследяването на работния процес. Наблюдателят следи за безопасността на наемателите на велосипедите. Техническата поддръжка отговаря за състоянието и изправността на велосипедите.

3.2.1.3 Описание на обкръжението

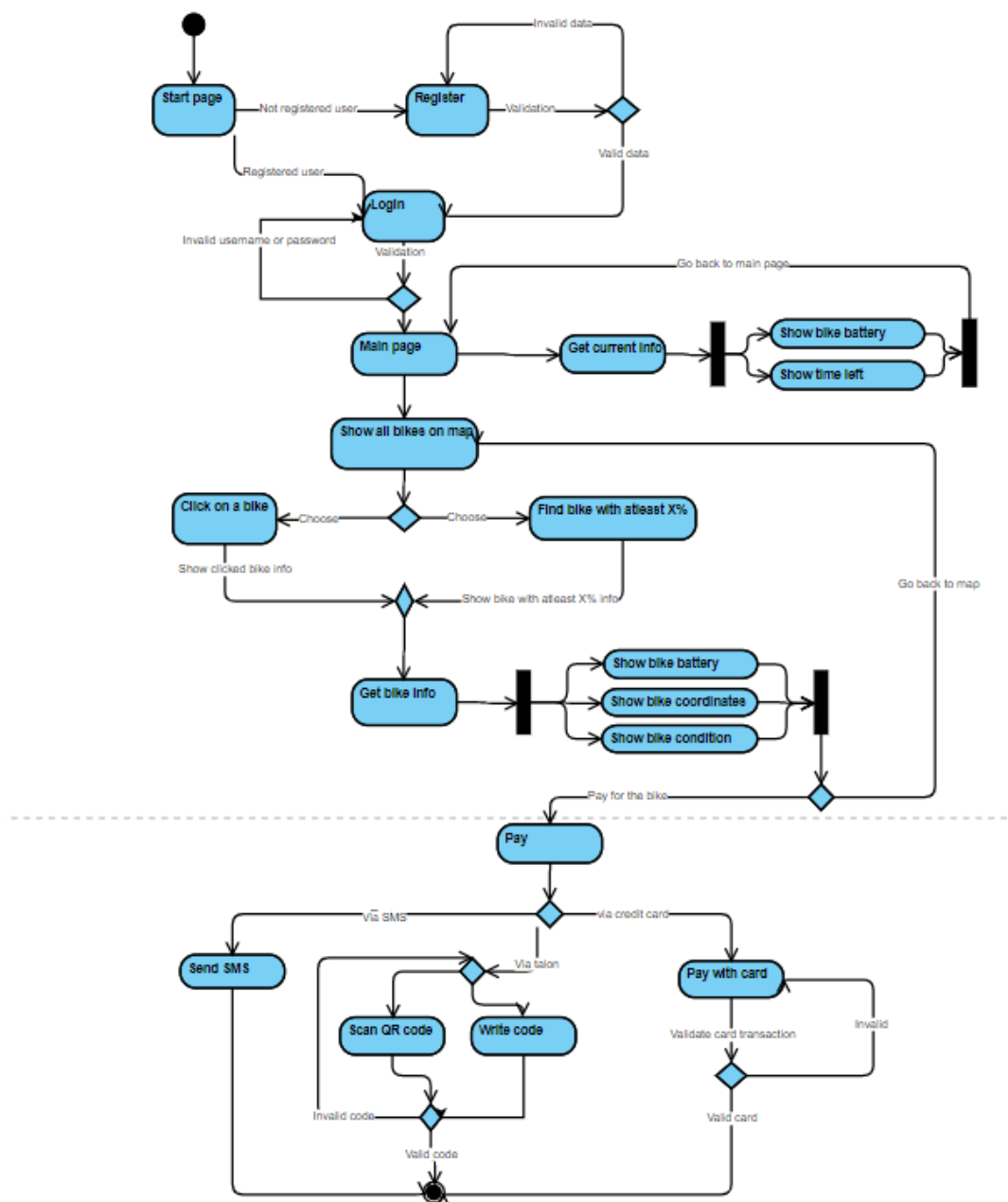
За предоставяне на функционалностите на уеб приложението се използва информация от външни обекти. Сензорите на велосипедите дават информация за състоянието, изправността и местоположението им.

3.2.1.4 Описание на възможните вариации

Системата е проектирана така, че да е възможно да се разширяват възможностите на системния администратор, наблюдателите, техническата поддръжка, както и на други бъдещи длъжности. Целта е максимално да се улесни работата.

3.2.2 Структура на процесите на VeloCity Mobile

3.2.2.1 Представяне



3.2.2.2 Описание на елементите и връзките

При отваряне на приложението потребителя избира дали да регистрира нов акаунт или да влезе с вече съществуващ. При регистрация, данните на потребителя се проверяват и при коректно въведени данни и несъществуващ вече акаунт се регистрира новият акаунт. След което потребителят бива препратен към Login скрийна. При коректно въведени данни потребителят влиза в системата. При влизане потребителят има избор да види информацията за момента (ако е взел колело, да види оставащото време и батерия на колелото) или да отвори картата и да види всички свободни колела, както и техните местоположения. При натискане на случайно свободно колело излиза цялата

информация за него(батерия,къде се намира, състояние и т.н.) като има възможност да го активира с плащане. При натискане на търсене на колело с поне X% батерия се въвежда исканото X и при наличие на поне едно такова колело се показва това, което се намира на най-късо разстояние до потребителя. След което при избор на заплащане на колело потребителят има 3 възможности да заплати - кредитна карта,смс или чрез талон.При избор на смс, потребителя изпраща смс и така активира даденото колело. При избор чрез талон, потребителят избира дали да въведе кода ръчно или чрез сканиране на QR код, след което кода се проверява и при коректен код се активира колелото, а при некоректно се връща отново в въвеждането на кода. При избор чрез дебитна карта, при изпращане на заявката за плащане, се проверява за коректността на дебитната карта и при грешка се връща отново за въвеждането на данни за дебитната карта, а при правилно въведени данни и при наличие на достатъчна сума в сметката на потребителя, се активира избраното колело.

3.2.2.3 Описание на обкръжението

В дадената система се използва външен доставчик за географски карти, валидирането на банковите карти отново биват изпращани до външен източник. Също така за четене на QR кодовете се използват вече съществуващи компоненти.

3.2.2.4 Описание на възможните вариации

Гореспоменатата система е съобразена с нуждата за добавяне на нови източници на географски карти, както и при нужда на различни модели банкови карти.

4. Архитектурна обосновка

4.1 Функционални изисквания

4.1.1 Наемателите на велосипед се регистрират чрез мобилното приложение, като в профила им се включват следните данни: имена, ЕГН, както и данни за връзка.

Регистрацията на потребителите е най-важната функционалност, която позволява на клиентите използването на останалите функционалности на системата. Всеки потребител може да използва своя личен профил да проследява свободни велосипеди спрямо неговото текущо местоположение, както и да проследи заряда на батерията на всеки един велосипед. Понеже устройствата могат да се използват определено време, след това те трябва да се оставят на стоянката, потребителят може да проследява колко време точно му остава. При евентуални проблемни ситуации, като загуба на връзка с даден велосипед, инцидент и др., именно профилът е връзката между потребителя и лицата, които стоят зад продукта – наблюдателите, техническата поддръжка, системните администратори и др

4.1.2 Системните администратори и наблюдателите използват системата

през уеб приложение.

По този начин те ще имат непрекъснат достъп до всеки един от велосипедите, да следят за евентуални повреди или нередности, и при възникване на грешки, от страна на потребителя, да могат бързо да ги отстраняват, като проблеми с плащането чрез банкова карта/смс, при евентуална загуба на връзка с велосипед, при индикация за случил се инцидент с потребителя. Уеб приложението е от изключителна важност за екипа, понеже така те могат да извършват своите задължения.

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

Всеки велосипед разполага с вградена система, чрез която се следи точното местоположение, както и заряда на батерията, на даденото устройство. Потребителят получава информация за всички свободни устройства в неговото обкръжение със заряд, който е по-голям или равен на зададения заряд в заявката за търсене. Тази информация помага на клиента да избере най-подходящия велосипед за него според нуждите му, както и да проследи къде има стоянки при евентуална нужда от зареждане на велосипеда или замяната му с друг.

4.1.4 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).

Потребителите могат да изберат най-удобния за тях начин на плащане. Налична система за разплащане, която улеснява потребителя максимално. Тази система трябва да е в разрез с изискването за сигурност, затова е необходимо тя да може да гарантира, че личните данни на всеки потребител, като номер на банкова карта, телефонен номер и др., са напълно защитени. Така ще се осигури едно сигурно плащане. Системата трябва да разполага и с четец за QR-кодове, което би улеснило значително дадения човек, използващ приложението. Чрез всички тези вариации за плащане, потребителят има възможност да избере този начин, който е най-удобен за него, и да бъде сигурен, че личните му данни остават защитени.

4.1.5 При 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

Сигнализирането на загубена връзка с велосипед може да бъде основополагащ фактор за откраднат велосипед или тежка катастрофа. По преценка на наблюдателя могат да бъдат сигнализирани органите на реда за грабеж или за злополука. Системата пази информация за изминатия път на потребителя (с дадения велосипед) за последните 30 мин., което помага на наблюдателите да установят евентуалното текущо положение на устройството. При излизане на велосипед от рамките на града, трябва да се сигнализира наблюдателя в рамките на 1 мин, като се изпратят данни за движението на велосипеда в последните 30 мин. При засичане на пътен или друг инцидент с велосипеда, се изпраща автоматично сигнал до спешна помощ (112), в рамките на 1 сек след засичане на инцидента. В рамките на 5 сек се известява и наблюдателя на системата. Тази информация се предава на

наблюдателите след известяване за загубена връзка. Известие се изпраща при изчезването на велосипед от радара или при сигнализация за възможен инцидент.

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

Възможността за интегриране с различни географски карти позволява улесняване на потребителите да могат по-лесно да разучат приложението с наличните им онлайн услуги за географски карти, без да е необходимо да се съобразяват с точно определена онлайн услуга. По този начин се получава усложняване на системата в замяна на по-добър “user experience”

4.2 Качествени изисквания

4.2.1 Личните данни на потребителите трябва да са абсолютно защитени от външна намеса. Достъпни са единствено до наблюдателя на правомерното използване на велосипедите. (Security)

Сигурността и защитата на личните данни е с най-висок приоритет. Тя е от голяма необходимост за системата. Потребителите трябва да бъдат уверени, че, предоставяйки своята информация, те няма да станат жертва на някоя измама или кражба на лични данни. Достъп до информацията имат само наблюдателите на правомерното използване на велосипедите.

4.2.2 Допускат 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. (Availability)

Поради спад в заинтересоваността от приложението в интервала 2:30-5:30, се допуска профилактика на системата в този часови диапазон, като поне един час преди профилактиката се изпраща известие до всички потребители. Така броят на засегнатите потребители от профилактиката ще бъде минимален и потребителите ще бъдат максимално добре информирани.