

Курсов проект на тема VeloCity

ФН 62121 Даниел Димитров

ФН 62167 Иван Чучулски

Съдържание

1. Въведение

- 1.1. Организация на текущия документ
 - 1.1.1. Предназначение на документа
 - 1.1.2. Списък на структурите
 - 1.1.3. Структура на документа
- 1.2. Общи сведения за системата
- 1.3. Разширен терминологичен речник
 - 1.3.1. Списък на софтуерните елементи
 - 1.3.2. Други специфични термини

2. Декомпозиция на модулите

- 2.1. Общ вид на декомпозицията на модулите
- 2.2. Application Module
 - 2.2.1. Предназначение на модула
 - 2.2.2. Отговорности на модула
 - 2.2.3. Елементи на модула
 - 2.2.4. Други
 - 2.2.4.1. Ограничения при употреба
 - 2.2.4.2. Грешки и изключения
 - 2.2.4.3. Зависимости от други елементи
 - 2.2.4.4. Описание на възможните вариации
- 2.3. Bicycle Module
 - 2.3.1. Предназначение на модула
 - 2.3.2. Отговорности на модула
 - 2.3.3. Елементи на модула
 - 2.3.4. Други
 - 2.3.4.1. Зависимости от други елементи
- 2.4. Computation Module
 - 2.4.1. Предназначение на модула
 - 2.4.2. Отговорности на модула
 - 2.4.3. Елементи на модула

2.4.4. Други

2.4.4.1. Ограничения при употреба

2.4.4.2. Грешки и изключения

2.4.4.3. Зависимости от други елементи

2.5. Server Module

2.5.1. Предназначение на модула

2.5.2. Отговорности на модула

2.5.3. Елементи на модула

2.5.4. Други

2.5.4.1. Ограничения при употреба

2.5.4.2. Грешки и изключения

2.5.4.3. Зависимости от други елементи

2.6. User Manager

2.6.1. Предназначение на модула

2.6.2. Отговорности на модула

2.6.3. Елементи на модула

2.6.4. Други

2.6.4.1. Ограничения при употреба

2.6.4.2. Грешки и изключения

2.7. Web Site

2.7.1. Предназначение на модула

2.7.2. Отговорности на модула

2.7.3. Елементи на модула

2.7.4. Други

2.7.4.1. Ограничения при употреба

2.7.4.2. Грешки и изключения

2.7.4.3. Зависимости от други елементи

3. Описание на допълнителни структури

3.1. Употреба на модулите

3.1.1. Първично представяне

3.1.2. Мотивация за избор

3.1.3. Описание на елементи и връзки

3.2. Структура на процесите

3.2.1. Мотивация на избор

3.2.2. Процес на диагностициране на информация от сензорите

3.2.2.1. Първично представяне

3.2.2.2. Описание на елементите и връзките

3.2.2.3. Описание на обкръжението

3.2.3. Процес на промяна на конфигурационните параметри

3.2.3.1. Първично представяне

3.2.3.2. Описание на елементите и връзките

3.2.3.3. Описание на обкръжението

4. Архитектурни драйвери

4.1. Функционални изисквания

4.2. Качествени изисквания

1. Въведение

1.1 Организация на текущия документ

1.1.1 Предназначение на документа

Целта на този документ е да опише и представи софтуерната архитектура на системата VeloCity.

1.1.2 Списък на структурите

- Декомпозиция на модулите - Системата е разделена на Server, Bicycle, Computation, App, Web Site, User Manager и Database. Модулът Bicycle отговаря за осигуряването на основните абстракции на сензорите във велосипедите и съответно в Computation се извършва изчисляването на текущите стойности на параметрите на всеки велосипед и пресмятането на маршрута му. В Server се случва разпределение на уведомление и сигнали към всички останали модули и в него се прави връзка с основната база данни. App и Web Site служат за взаимодействие със системата на съответните типове потребители, а в User Manager се помещават личните данни на всеки потребител, като те са защитени от външна намеса и достъпът до тях се извършва след автентикация и авторизация на лицата.
- Допълнителни структури
 - Употреба на модулите – разглеждат се основните комуникационни връзки между модулите от декомпозицията.
 - Структура на процесите – представени са някои от най-важните процеси, протичащи в системата и

съответните действия на модули при конкретни входни данни.

1.1.3 Структура на документа

- Секция 1 – Въведение. Съдържа общи сведения за системата, описва как е структуриран документа и речник на използваните понятия.
- Секция 2 - Декомпозиция на модулите. Описано е разпределението на системата на модули според архитектурните драйвери и останалите изисквания към системата, като за всеки модул са описани неговото предназначение и отговорности.
- Секция 3 - Допълнителни структури. Представени са употреба на модулите и две структури на процесите, чрез които се дава по-ясна представа за работата на системата по време на изпълнение.
- Секция 4 - Архитектурна обосновка. Това са най-важните изисквания към работата на системата, като те са разделение на две категории – функционални и качествени изисквания и за всеки драйвер е дадена обосновка, защо е избран.

1.2 Общи сведения за системата

VeloCity е система за отдаване и наемане на велосипеди снабдени с електрическа система за задвижване, като идеята е потребителите да намират лесно и бързо велосипеди чрез мобилно приложение, да могат да интегрират услуги за географски карти и да имат гъвкави опции за заплащане. Също така е осигурено и уеб приложение за системните администратори, които се грижат за поддържане и обновление на системата, наблюдатели, които следят за правомерното използване на системата и екипи по техническа поддръжка, които трябва да бъдат уведомени при настъпване на счупвания и неизправности във велосипедите.

1.3 Разширен терминологичен речник

1.3.1 Списък на софтуерните елементи

Номерата пред елементите са съответните секции, в които те се намират

2.2 App

2.2.3.1 Access Manager

2.2.3.1.1 Login

2.2.3.1.1.1 Authenticate

2.2.3.1.1.2 Authorise

2.2.3.1.2 Register

2.2.3.1.3 Edit Account Info

2.2.3.2 Payment

2.2.3.2.1 Credit Card

2.2.3.2.2 Voucher

2.2.3.2.3 SMS

2.2.3.3 Notifications

2.2.3.4 Navigation

2.2.3.5 Find Nearest Bikes

2.2.3.6 Give Feedback

2.3 Bicycle Module

2.3.3.1 Sensor Manager

2.3.3.1.1 GPS

2.3.3.1.2 Self Diagnostic

2.3.3.1.2.1 Smart Sensors

2.3.3.2 Configuration Parameters

2.3.3.2.1 Battery Indicator

2.3.3.2.2 Time To Live

2.3.3.3 Signal Sender

2.3.3.3.1 Sensor Data

2.3.3.3.2 Configuration Data

2.4 Computation Module

2.4.3.1 Route Tracker

2.4.3.1.1 GPS Database

2.4.3.2 Processing Unit

2.4.3.2.1 Bike Parameters

2.4.3.3 Communication Manager

2.4.3.3.1 Communication With Server

2.4.3.3.1.1 Route Data

2.4.3.3.1.2 Sensor Fault

2.4.3.3.1.3 Server CFG

2.4.3.3.2 Communication with Bikes

2.4.3.3.2.1 GPS Data

2.4.3.3.2.1 Sensor Data

2.4.3.3.2.2 Bike CFG

2.5 Server Module

2.5.3.1 Scheduler

2.5.3.1.1 Data For Processing

2.5.3.2 Update Manager

2.5.3.2.1 Web Site

2.5.3.2.2 App

2.5.3.2.3 User Manager

2.5.3.3 Configuration Settings

2.5.3.4 Signal

2.5.3.4.1 Notify Observer

2.5.3.4.2 Emergency Call

2.5.3.5 Database manager

2.6 User Manager

2.6.3.1 User Data

2.6.3.1.1 User History

2.6.3.1.2 Profile Details

2.6.3.2 Access Manager

2.6.3.2.1 Authentication

2.6.3.2.2 Authorisation

2.7 Web Site

2.7.3.1 Profile Manager

2.7.3.1.1 Administrators

2.7.3.1.1.1 System Maintenance

2.7.3.1.2 Observers

2.7.3.1.2.1 Incident Record

2.7.3.1.3 Technical Support

2.7.3.1.3.1 Signals

2.7.3.2 Access Manager

2.7.3.2.1 Authentication

2.7.3.2.2 Authorisation

2.8 Database

1.3.2 Други специфични термини

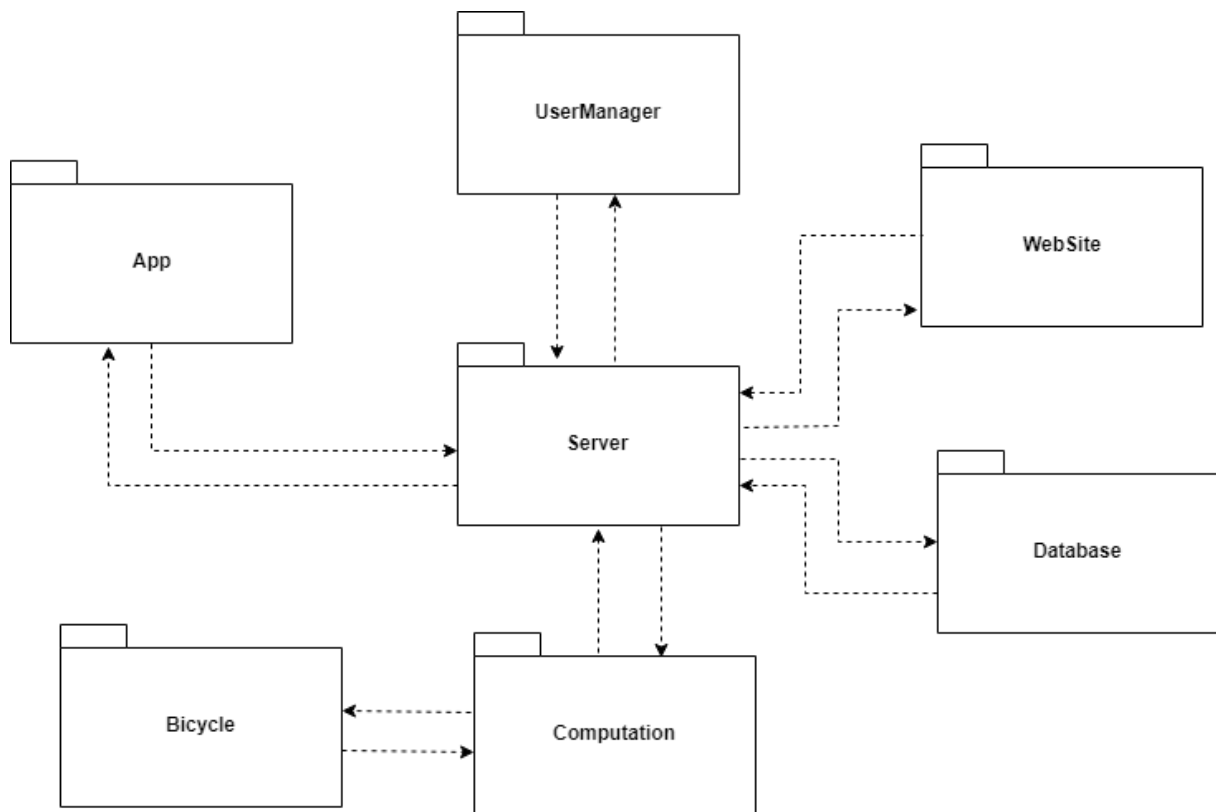
- Софтуерна архитектура - съвкупност от структури, които служат за документиране на основните решения (за дизайна) на софтуерната система и улесняват дискусията между заинтересованите лица. Показват се също така поведение на системата и вътрешни и външни взаимодействия
- Софтуерни структури : множество от софтуерни елементи които имат някаква връзка помежду си. Те биват модули, компоненти и конектори, структури на разположението
- Модули (modules) - единици, които имат конкретни функционални отговорности, те са „статични“ в смисъл, че описват как функционалността на системата е разделена на отделни и по-малки части, но не описва как тези части си взаимодействат в по време на изпълнение
- Компоненти и Връзки (component and connector structures) (още структури на процесите) - това са „динамични“ структури в смисъл, че показват системата като съвкупност от елементи, които се характеризират с поведение по време на изпълнение (компоненти) и взаимодействията (връзки) между тях, чрез които по време на работа на системата се осъществява нейната функционалност
- Структури на разположението (allocation structures) - показват какво е отношението на системата към организационната, разработващата и изпълняващата ѝ среда, напр. за конкретни елементи какво е тяхното

разпределение върху хардуера, файлово разпределение, връзки по интернет мрежата

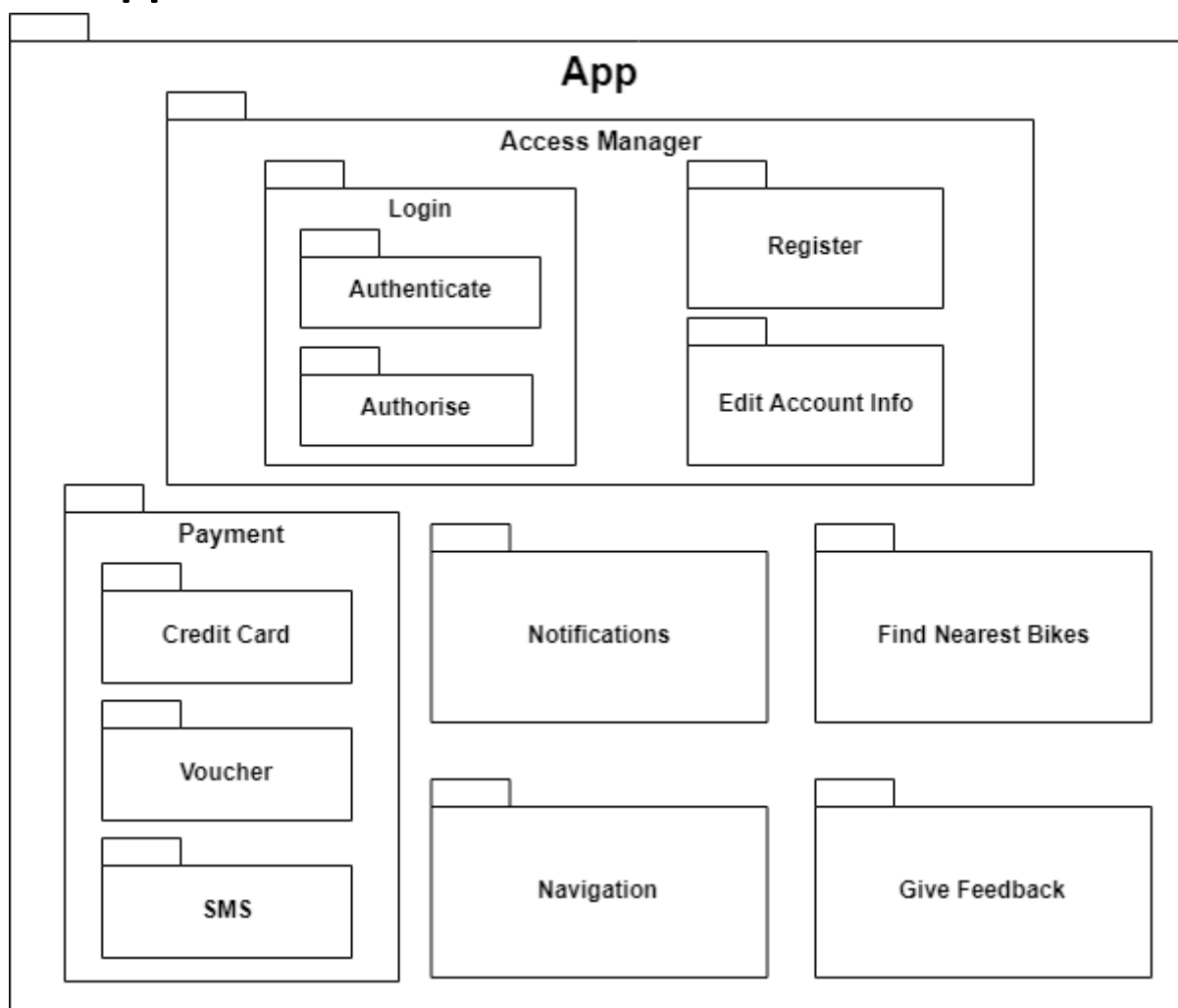
- Мобилно приложение - компютърна програма, която е предназначена да работи на мобилни устройства като смартфони и планшети
- Сървър - компютърна програма, която предоставя услуги на други програми, наречени клиенти
- Клиент – компютърна програма, която осъществява функционалността си чрез достъпване на услуга, предоставена от сървър
- Автентикация (Authentication) - процес на установяване и потвърждаване на самоличността на потребителя, т.е. дали такъв фигурира в системната база от данни
- Авторизация (Authorisation) – процес, чрез който се уточняват правата на достъп на потребителя, тоест до предоставяне на достъп само до необходимата информация
- База от данни - представлява колекция от логически свързани данни, които са структурирани по определен начин така, че да се осъществява ефективното им модифициране. Също така се осигурява надеждно и дълготрайно съхранение данните.
- Външната система – система, която е различна от разглежданата и до която се комуникира чрез определен интерфейс.
- Интерфейс – набор от правила, чрез които даден компонент или система предоставя определени услуги на друг, бива два типа : входен и изходен
- Наблюдатели на системата – следят за настъпване на нарушения по време на използване на системата и извършват съответни действия при нужда
- Администратор – човек, който е отговорен за поддръжката и обновяването на софтуера и има достъп до повечето подсистеми

2.Декомпозиция на модулите

2.1 Общ вид на декомпозицията на модули за системата



2.2 Application Module



2.2.1 Предназначение на модула

Това е мобилно приложение предназначено за използване от наемателите на велосипеди (обикновените потребители).

2.2.2 Отговорности на модула

Осъществява login на лицата, които използват системата, като им предоставя определени функционалности. Реализират се различните начини за заплащането на услугата. Предоставя информация за най-близките свободни велосипеди на потребителите. Сигнализира потребителите при превишаване на времето за използване на велосипеда. Позволява интеграция с онлайн услуги за географски карти. Има възможност за изпращане на обратна връзка.

2.2.3 Елементи на модула

2.2.3.1 Access Manager

Този подмодул отговаря за регистриране и влизане в системата от страна на обикновените потребители чрез мобилното приложение.

2.2.3.1.1 Login

Модулът отговаря за вписване на обикновените потребители в приложението.

2.2.3.1.1.1 Authenticate

Този подмодул отговаря за автентикация на потребителя спрямо системата , т.е. идентифициране, че този потребител съществува в системата.

```
public bool IsAuthenticated(string username, string password)
```

2.2.3.1.1.2 Authorise

Този подмодул отговаря за авторизиране на потребителя спрямо системата, т.е. дават му се само права за определени функционалности на приложението.

```
public bool CheckUserRights(string user_id)
```

2.2.3.1.2 Register

Отговорност на този подмодул е да се извършва регистрация на потребители в системата през него.

```
public UserProfile GetUserInput()
```

```
private bool CheckIfUsernameAlreadyExists(string username)
```

```
public bool SendEmailConfimartion(string user_email)
```

2.2.3.1.3 Edit Account Info

Този модул позволява на наемателите на велосипеди да преглеждат и редактират личните си данни.

```
public void ChangeUsername(string new_username)
```

```
public void ChangeEmail(string new_email)
```

```
private void SendEmailConformation(string user_mail)
```

2.2.3.2 Payment

В този подмодул се осъществяват плащанията от страна на клиентите за използваната услуга.

2.2.3.2.1 Credit Card

Този модул осъществява функционалността по плащане чрез кредитна карта.

```
public void InvalidCardError(string err_messange)
```

```
public void PayViaCard()
```

2.2.3.2.2 Voucher

Този модул осъществява функционалността по плащане чрез ваучер.

```
public QRcode ScanCode()
```

```
public void PayViaVoucher(QRcode)
```

2.2.3.2.3 SMS

Този модул осъществява функционалността по плащане чрез СМС.

```
public void InvalidPhoneNumber(string err_message)
```

```
public void PayViaSMS()
```

2.2.3.3 Notifications

Модулът осъществява функционалността по известяване на клиента през уеб приложението в определени ситуации – например пресрочено време на използване или неправомерно използване на велосипеда.

```
public void UseTimeExceeded(string message)
```

```
public void DangerousUse(string message)
```

2.2.3.4 Navigation

Този модул служи за интеграция с външни системи за предоставяне на услуги за географски карти.

```
public void ImportSettings(MapsConfig maps_config)
```

```
public MapsConfig ExportSettings()
```

2.2.3.5 Find Nearest Bikes

Модулът служи за изпращане на заявка за намиране на най-близкия велосипед, готов за използване на потребителя.

```
public void FindBike(GpsPos pos)
```

2.2.3.6 Give Feedback

Модулът дава възможност на потребителите на системата да дават обратна връзка на администраторите.

```
public void ProvideFeedback()
```

2.2.4 Други

2.2.4.1 Ограничения при употреба

Приложението е трябва да бъде инсталирано, за да могат потребителите да го използват. Също така е нужна връзка към интернет и наличие на GPS сензор за осъществяване на основните функционалности като намиране на велосипеди и използване на услуги за географски карти.

За заплащането чрез SMS е нужно изпращане на съобщение през СИМ картата на устройството, върху което е инсталирано приложението.

2.2.4.2 Грешки и изключения

При възникване на изключение по време на работата на приложението, потребителят бива осведомен, чрез подходящо съобщение за съответната грешка.

2.2.4.3 Зависимости от други елементи

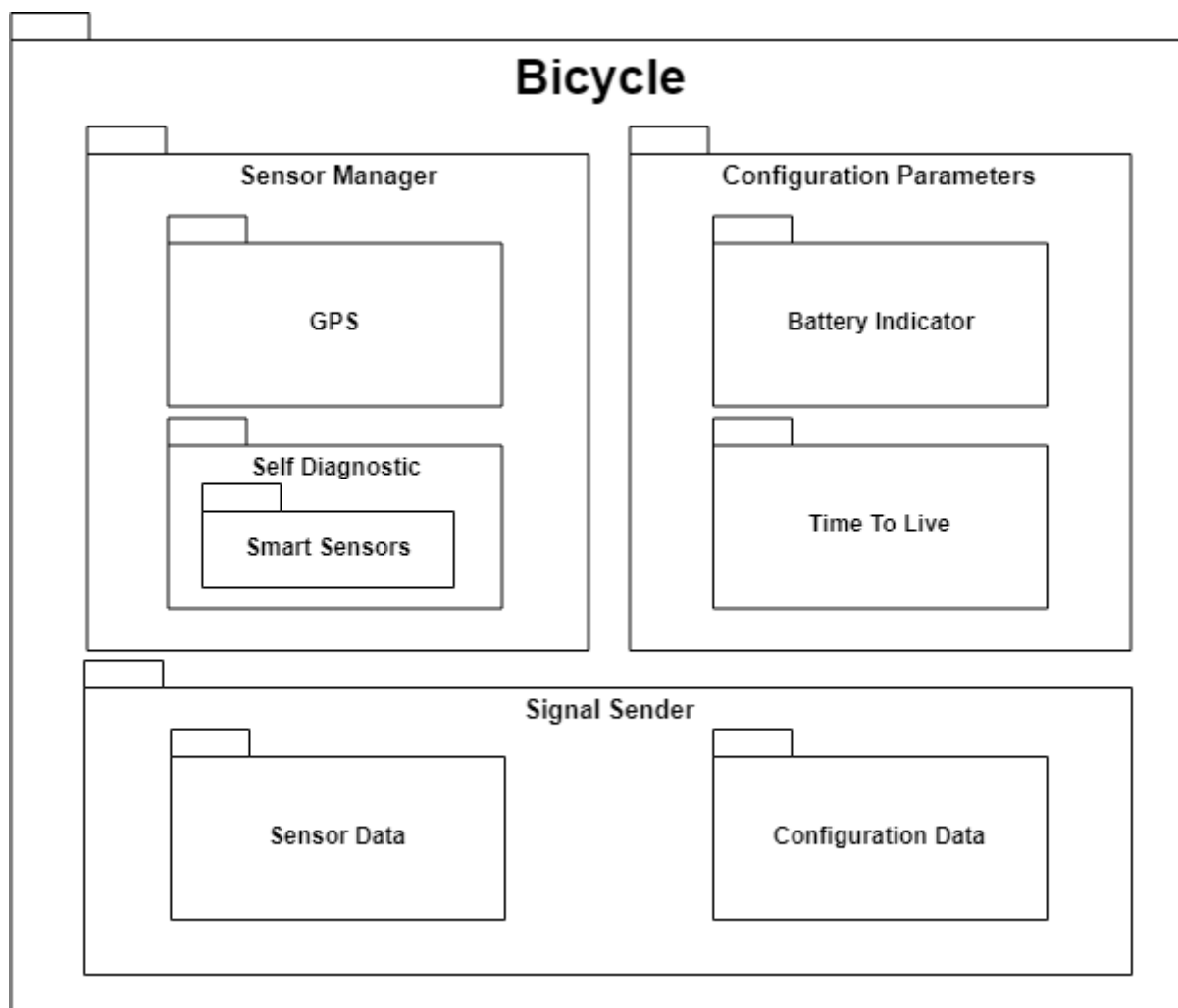
Модулът има зависимости към външни системи за осъществяване на плащането чрез кредитна карта, както и за предоставянето на услугите за географски карти. В системата, модулът App зависи от модулът Server, към който отправя заявки за намиране на велосипеди и получава отговор, а също така получава и нотификации за неправомерно използване.

Тъй като модулът е отделно приложение, което се инсталира върху някакво мобилно устройство, приложението ще е зависимо от операционната система и как тя разпределя правата за използване на хардуера.

2.2.4.4 Описание на възможните вариации

Възможно е добавянето на нови начини на заплащане, както и свързване с нови услуги за географски карти. Също така може да се добави възможност за регистрация, чрез използване на съществуващ акаунт от компании, които имат вече развита инфраструктура, както за географски карти, така и за заплащане, например Google Maps и Google Pay.

2.3 Bicycle Module



2.3.1 Предназначение на модула

Получаване на информация за велосипедите чрез сензори за тяхното техническо състояние и данни за местоположението им.

2.3.2 Отговорности на модула

Диагностициране на техническото състояние на велосипеда, чрез смарт сензори. Събиране на информацията от смарт сензорите и GPS системата и изпращане към модула за изчисления (Computation). Съхранение на информация за конфигурационните параметри на системата.

2.3.3 Елементи на модула

2.3.3.1 Sensor Manager

Този модул е предназначен за извличане на информацията от сензорите, които са закрепени на велосипедите.

2.3.3.1.1 GPS

Този модул е предназначен за генериране на информация за текущото физическо местоположение на велосипеда.

```
public GpsPos GetGps()
```

2.3.3.1.2 Self Diagnostic

Модулът е предназначен за диагностициране на физическото състояние на велосипеда.

2.3.3.1.2.1 Smart Sensors

В този модул се съдържат Смарт сензори, които генерират информация, нужна за диагностициране на състоянието на компонентите на велосипеда.

```
public bool GpsAlive()
```

```
public bool ElectricMotorAlive()
```

```
public bool HasFlatType()
```

```
public SensorInfo CheckSensorState()
```

2.3.3.2 Configuration Parameters

В този модул се съдържат конфигурационните параметри на велосипеда.

2.3.3.2.1 Battery Indicator

В този модул се съхранява информация за състоянието на батерията на велосипеда, нейния моментен заряд, както и конфигурационния

параметър на системата за процент на заредена батерия при наемане на велосипед.

```
public unsigned int GetBatteryLevel()
```

```
public unsigned int GetBatteryProcent()
```

```
public void SetMinBatteryProcent(unsigned int new_min_procent)
```

2.3.3.2.2 Time To Live

В този модул се съхранява информация за оставащото време, в което велосипедът може да бъде използван от клиента.

```
public int GetTimeToLive(string BikeID)
```

```
public void AdjustTimeToLive(string BikeID, int new_ttl)
```

2.3.3.3 Signal Sender

Този модул отговаря за изпращане на събраната информация от сензорите и конфигурационните параметри на велосипеда към модула за изчисления.

2.3.3.3.1 Sensor Data

В този модул се съдържа информацията, генерирана от сензорите на велосипеда. При настъпване на техническа повреда тук ще се събират данни от сензорите, които ще послужат за по-ефективно откриване и отстраняване на повредите.

```
public SensorInfo GetSensorData(string BikeID)
```

```
public unsigned int GetBatteryProcent(string BikeID)
```

2.3.3.3.2 Configuration Data

В този модул се съдържа информацията за текущото състояние на конфигурационните параметри на велосипеда.

```
public void AdjustConfig(ConfigParam newcfg)
```

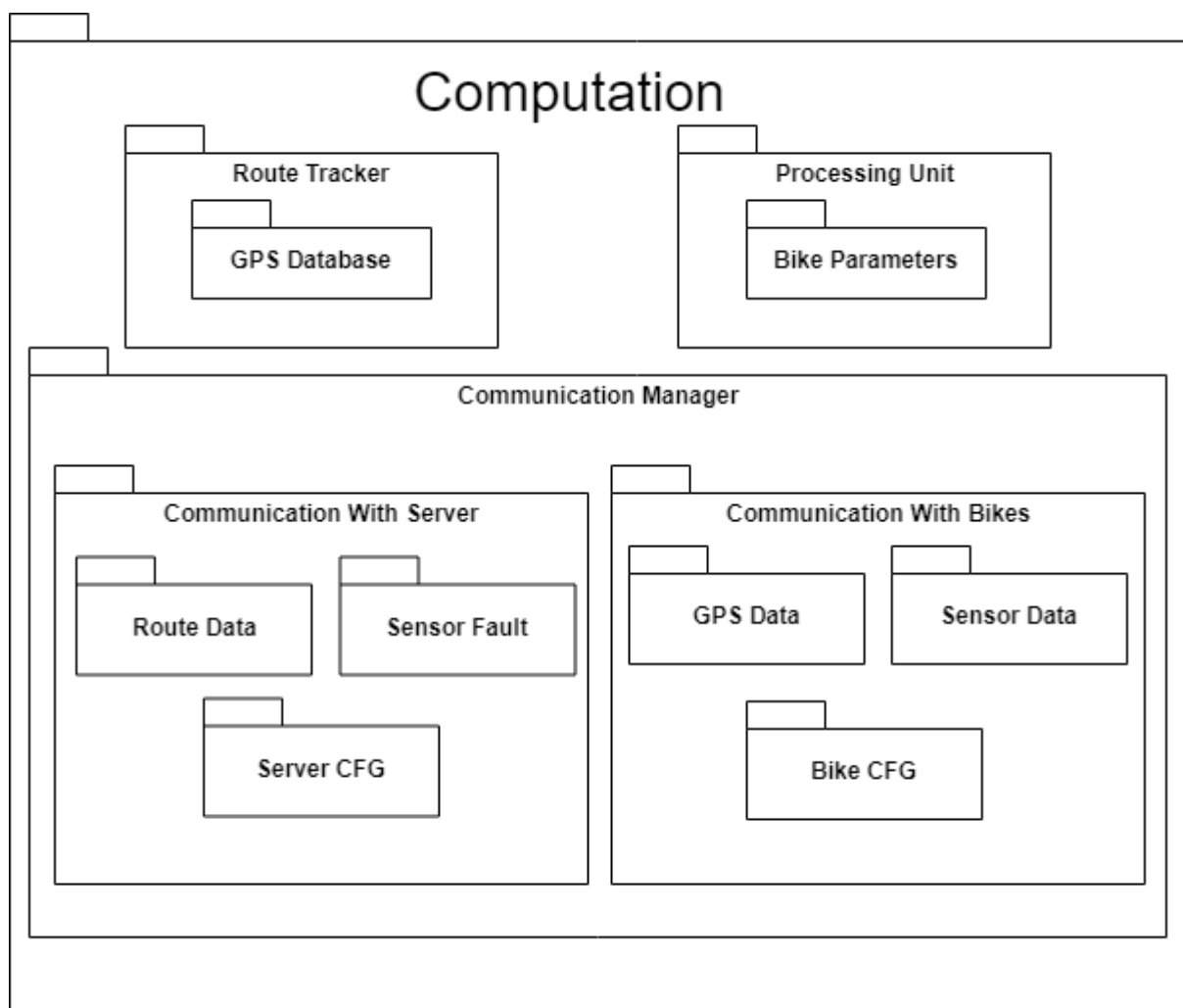
```
public ConfigParam GetConfig(string BikeID)
```

2.3.4 Други

2.3.4.1 Зависимости от други елементи

Модулът получава информация за текущите параметри на велосипедите от Bicycle, както и данни за настъпили повреди. Съответно Computation трябва да предприеме действия да изпрати тези данни към Server. От Server се получават нови конфигурационни параметри.

2.4 Computation Module



2.4.1 Предназначение на модула

Обработка на данните, получени от велосипедите и сървъра.

2.4.2 Отговорности на модула

Получава информация от велосипедите за тяхното местоположение и системните им показатели в реално време. В подмодулът Route Tracker, чрез AI се прави обработка на данните за GPS координатите и се изчислява най-вероятното местоположение на велосипед, при загуба на връзка с него. За всеки велосипед се изчисляват неговите показатели спрямо конфигурационните параметри, зададени от сървъра.

2.4.3 Елементи на модула

2.4.3.1 Route Tracker

В този модул се осъществяват изчисленията за маршрута на велосипеда в последните 30 минути.

2.4.3.1.1 GPS Database

В този модул се съдържа база данни, в която се съхраняват маршрутите на велосипедите в последните 30 минути.

```
public void AddPosition(string BikeID, GpsPos curr_pos)
```

```
public void DeleteOlderPositions(string BikeID)
```

```
public ClearPositions(string BikeID)
```

2.4.3.2 Processing Unit

Този модул изчислява текущите стойности на конфигурационните параметри на велосипеда и ги сравнява със зададените системни параметри.

2.4.3.2.1 Bike Parameters

В този модул се съдържат текущо зададените конфигурационни параметри на системата, с които се извършват изчисления, за да се осигури тяхното спазване.

```
private void ComputeTimeToLive(string Bike, ConfigParam curr_config)
private unsigned int GetBatteryLevel(string Bike)
public bool TimeToLiveExpired(string Bike)
public bool HasEnoughBattery(unsigned int batt_proc, ConfigParam
config)
```

2.4.3.3 Communication Manager

Този модул осъществява двустранна комуникацията между велосипед и сървър.

2.4.3.3.1 Communication With Server

Този модул отговаря за осъществяване на комуникация с модула за велосипед и модула сървър.

2.4.3.3.1.1 Route Data

В този модул се съдържат данни за маршрута на даден велосипед през последните 30 минути и неговото вероятно местоположение при загуба на връзка.

```
public void RouteData(GpsPos most_likely_position)
```

2.4.3.3.1.2 Sensor Fault

Този модул съдържа подробна информация от смарт сензорите на велосипед, при за настъпването на техническа повреда.

```
public void SendSensorData(SensorInfo Data)
```

2.4.3.3.1.3 Server CFG

В този модул се получават стойностите за текущите конфигурационни параметри на системата, зададени от модула Server.

```
public ConfigParam ReceiveConfig()  
public void ApplyConfiguration(ConfigParam new_cfg)
```

2.4.3.3.2 Communication with Bikes

Този модул отговаря за комуникация между сървър и велосипеди.

2.4.3.3.2.1 GPS Data

Този модул служи за получаване на данни от GPS сензорите във всеки велосипед и последващото им предаване на модула за намиране на маршрута.

```
public GpsPos GetBikePosition(string BikeID)
```

2.4.3.3.2.1 Sensor Data

В този модул от велосипедите се получава информация за техническото им състояние и евентуално при повреда ще се съдържат детайли за неизправността.

```
public SensorInfo GetBikeState(string BikeID)
```

2.4.3.3.2.2 Bike CFG

В този модул се получава информация за текущите показатели на конфигурационните параметри на системата във всеки велосипед.

```
public ConfigParam GetBikeCfg(string BikeID)
```

2.4.4 Други

2.4.4.1 Ограничения при употреба

Изчисленията се извършват през определен период от време, защото системата има краен изчислителен ресурс. От това следва, че скоростта за отговор на заявките е зависима от наличното количеството изчислителен ресурс, както и от натовареността на системата.

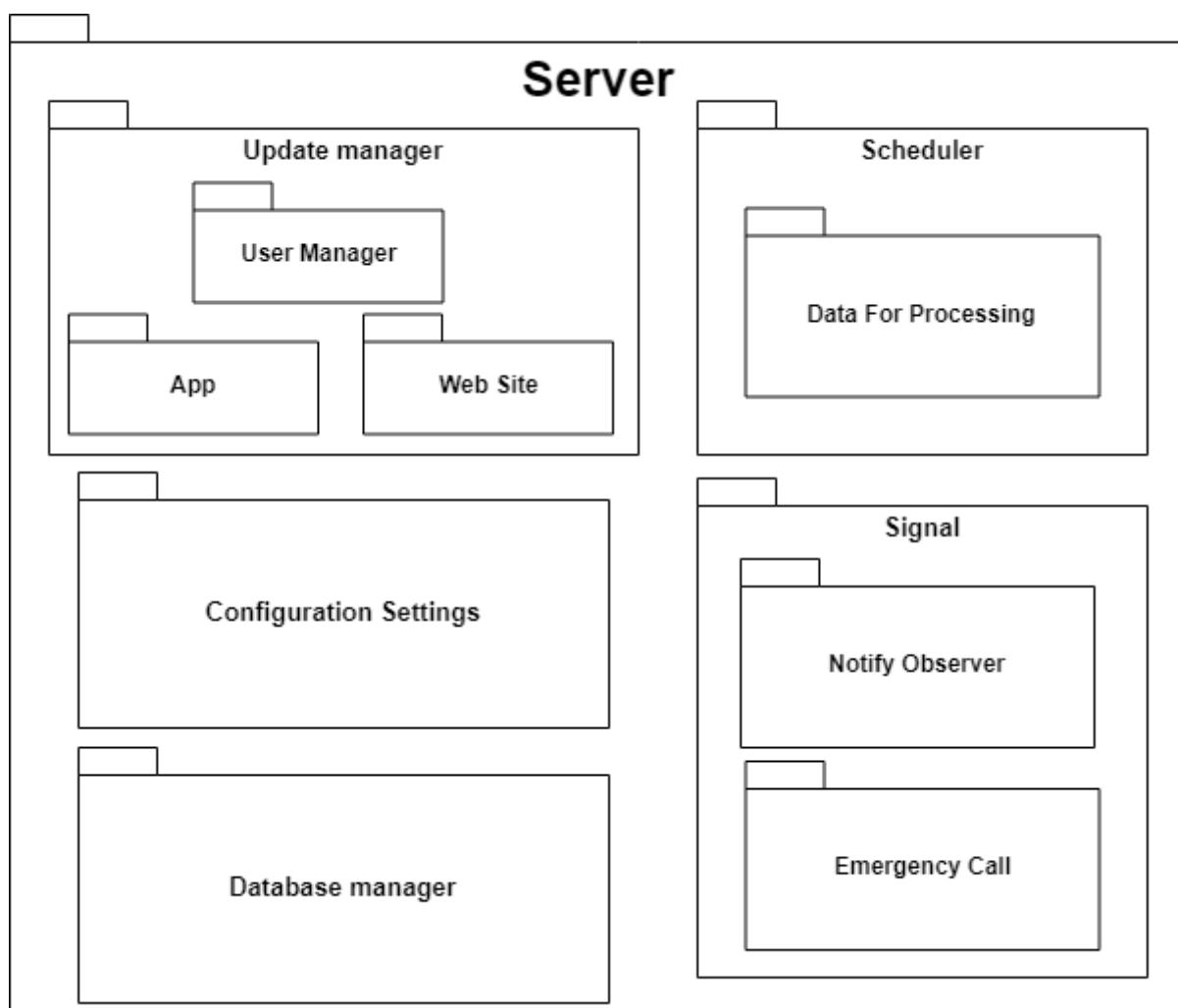
2.4.4.2 Грешки и изключения

При настъпване на отказ или срыв, модулът изпраща съобщение за настъпилото събитие към Server, откъдето тя трябва да се препрати към наблюдателите на системата.

2.4.4.3 Зависимости от други елементи

Модулът има зависимости към външни системи за осъществяване на плащането чрез кредитна карта, както и за предоставянето на услугите за географски карти. В системата, модулът App зависи от модулът Server, към който отправя заявки за намиране на велосипеди и получава отговор, а също така получава и нотификации за неправомерно използване.

2.5 Server Module



2.5.1 Предназначение на модула

Този модул осъществява комуникацията и обмена на информация между всички модули.

2.5.2 Отговорности на модула

В него се задават конфигурационните параметри на системата. Извършва сигнализация на наблюдателите на системата и на групата по техническа поддръжка. Чрез Scheduler се извършва приоритизиране за обработка на сигналите при аварии. Изпълнява основните операции с базата от данни Извършва обновления на уеб сайта, мобилното приложение и мениджъра на акаунти.

2.5.3 Елементи на модула

2.5.3.1 Scheduler

Този модул отговаря за разпределение и изпълнение на най-приоритетните задачи.

2.5.3.1.1 Data For Processing

В този модул се съдържат данните, които трябва да бъдат обработени от главния модул.

```
public void EventListener(Event new_event)
public Event GetTopEvent()
public Event[] GetAllEvents()
public bool GrantTopPriority(Event top_event)
public void AlterPriorities()
```

2.5.3.2 Update Manager

Този модул отговаря за съхранение на версиите за обновяване на компонентите на системата.

2.5.3.2.1 Web Site

Този модул отговаря за обновяване на уеб сайта на системата.

```
public void ReceiveNewVersion(SystemUpdate new_version)
```

```
public bool QueryScheduler(Event web_update)
```

```
public void PushUpdate(SystemUpdate new_version)
```

2.5.3.2.2 App

Този модул отговаря за обновяване на мобилното приложение на системата.

```
public void ReceiveNewVersion(SystemUpdate new_version)
```

```
public bool QueryScheduler(Event app_update)
```

```
public void PushUpdate(SystemUpdate new_version)
```

2.5.3.2.3 User Manager

Този модул отговаря за обновяване на мениджъра за управление на потребителите на системата.

```
public void ReceiveNewVersion(SystemUpdate new_version)
```

```
public bool QueryScheduler(Event user_manager_update)
```

```
public void PushUpdate(SystemUpdate new_version)
```

2.5.3.3 Configuration Settings

В този модул се задават и съхраняват конфигурационните параметри на системата като позволено време за използване на велосипедите и необходим процент заредена батерия за намиране на велосипеди.

```
public void ReceiveNewCofig(ConfigParam new_cfg)
```

```
public bool QueryScheduler(Event cfg_change)
```

```
public void StoreNewConfig(ConfigParam new_cfg)
```

```
public void SendConfig(ConfigParam new_cfg)
```

2.5.3.4 Signal

Този модул отговаря за сигнализиране на определени лица при настъпване на извънредни събития и процесите, които се случват в него се изпълняват с най-висок приоритет в сравнение с останалите процеси в модула Server.

2.5.3.4.1 Notify Observer

Този модул отговаря за сигнализиране на наблюдателите на системата при загуба на връзка с велосипед или сигнализиране на техническа поддръжки при диагностициране на някаква физическа повреда.

```
public void LostConnection(GpsData position)
```

```
public void TechnicalFailureOccured(SensorInfo sensor_data)
```

```
public void QueryScheduler(Event notify_observer)
```

```
public void SendPosition(GpsData position)
```

```
public void SendFailureDetails(SensorInfo sensor_data)
```

2.5.3.4.2 Emergency Call

Този модул отговаря за уведомяване на спешна медицинска помощ при настъпване на пътнотранспортно произшествие.

```
public void ReceiveIncident(Incident inc_details)
```

```
public void QueryScheduler(Event emergency)
```

```
public void MakeEmergencyCall(Incident incident)
```

2.5.3.5 Database manager

Този модул отговаря за осъществяване на връзката на сървъра с базата данни, извършване на заявки към базата и периодично записване на състоянието на системата в базата.

```
public void BackUpSystemState()
```

```
public void RestorePreviousState()
```

2.5.4 Други

2.5.4.1 Ограничения при употреба

Сървърът не може да обработи безкраен брой заявки и за да не бъде претоварен, информацията се обновява на определени периоди от време. Ако настъпи срив, системата може да се върне към предишно състояние, записано в базата от данни. За осъществяване на комуникация с останалите части на системата, модулът трябва да има връзка към интернет.

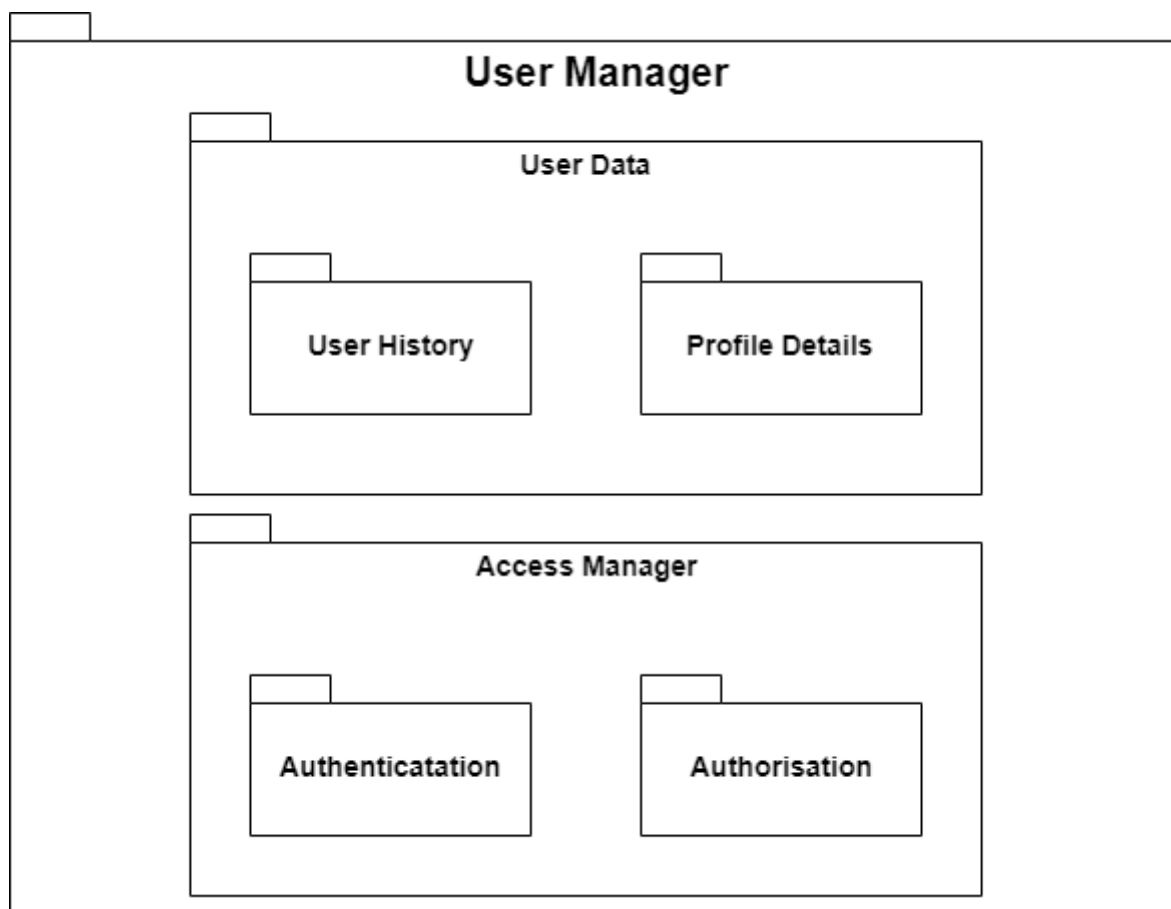
2.5.4.2 Грешки и изключения

При настъпване на грешка модулът връща на съответния инициатор на действието съобщение за грешка, която той трябва да обработи, а Server продължава работата си. Към събитията, които отправят заявка към Scheduler-а, но настъпва или се обработва събитие с по-висок приоритет, се изпраща подходящо съобщение, че действието е възпрепятствано и е нужно да се инициира отново.

2.5.4.3 Зависимости от други елементи

Сървърът се свързва с услуга за уведомяване на служби за спешна медицинска помощ при настъпване на инцидент. Също така модулът е зависим от скоростта на обмяна на данни с базата от данни.

2.6 User Manager



2.6.1 Предназначение на модула

Този модул е отговорен за съхранение и защита на личните данни на наемателите на велосипеди.

2.6.2 Отговорности на модула

Данните в този модул могат да се достъпят само от наблюдателите за правомерното използване на системата, които следят за потенциални злоупотреби с велосипедите, като имат достъп до историята на използване на потребителите.

2.6.3 Елементи на модула

2.6.3.1 User Data

В този модул се съдържа личната информация за всеки един от наемателите на велосипеди.

2.6.3.1.1 User History

В този модул се съхранява историята за дейностите на всеки потребител, дата и продължителност на използването на системата.

```
private Date GetLastTripDate(string user_id)
private int GetLastTripDuration(string user_id)
public HistoryRecord GetUserInfo(string user_id)
```

2.6.3.1.2 Profile Details

В този модул се съхраняват личните данни на потребителите, неговите имена и контакти за връзка.

```
public UserProfile GetUserDetails(string user_id)
```

2.6.3.2 Access Manager

Този модул ограничава достъпа до личните данни на потребителите на системата.

2.6.3.2.1 Authentication

Този модул отговаря за автентикация на потребителите, които искат да достъпят модула, необходимо за осигуряване на максимална защитеност на данните.

```
public bool IsAuthenticated(string username, string password)
```

2.6.3.2.2 Authorisation

Този модул отговаря за авторизация на потребителя, определя правата на потребителите, които достъпват модула, единствено наблюдателите на системата и самите потребители могат да преглеждат и модифицират данните в него.

```
public bool CheckUserRights(string user_id)
```

2.6.4 Други

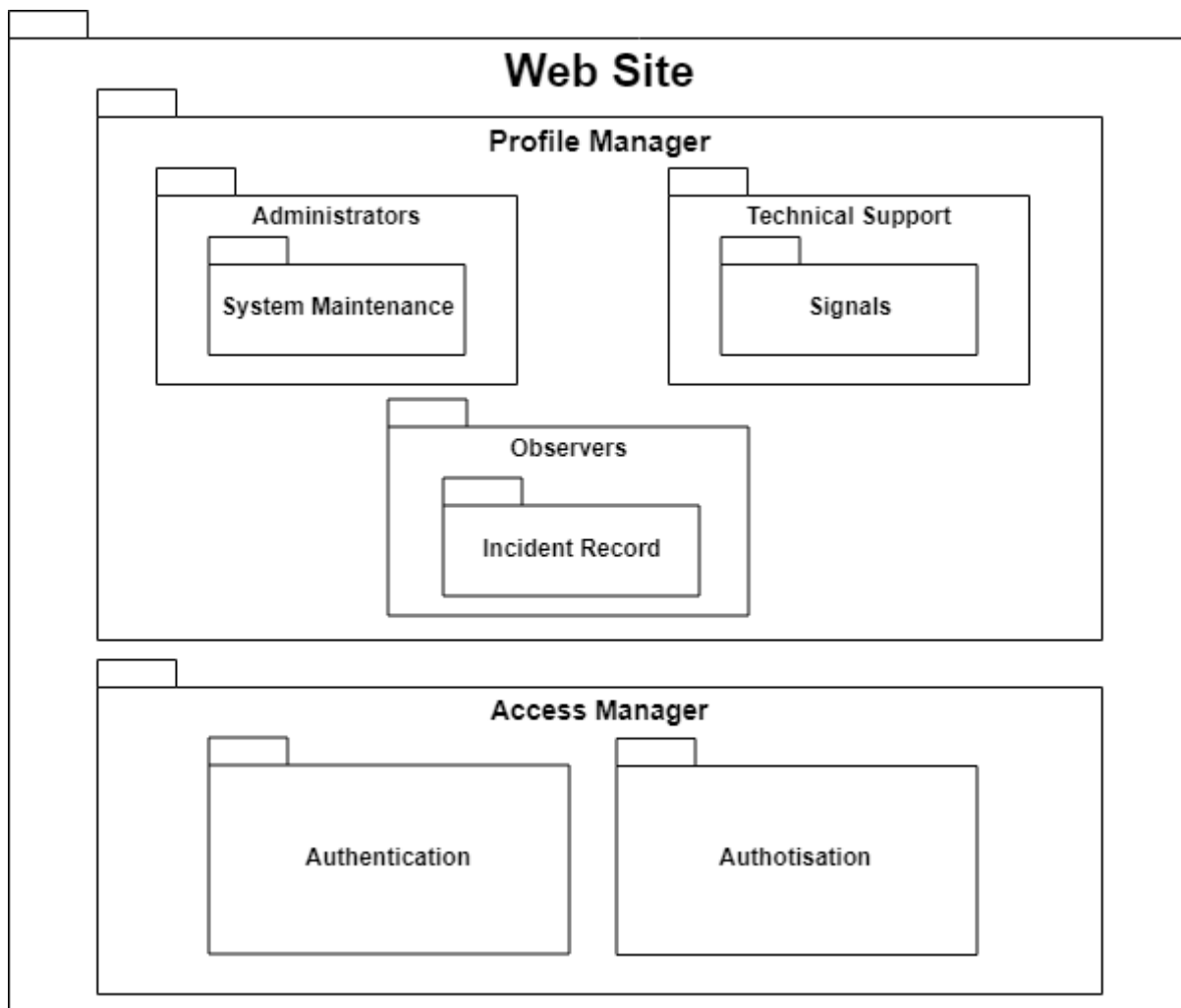
2.6.4.1 Ограничения при употреба

Модулът получава заявки единствено от App и от Web Site. При всеки опит за достъп се извършва ново процес по автентикация и авторизация на потребителя, желаещ да достъпи данните.

2.6.4.2 Грешки и изключения

При възникване на опит за достъпване на данните от неоторизиран източник, системата трябва да регистрира събитието, да блокира източника и да се сигнализира на наблюдателите на системата.

2.7 Web Site



2.7.1 Предназначение на модула

Предназначен за използване на системата от служебните потребители.

2.7.2 Отговорности на модула

Извършва се автентикация и авторизация на лицата, които искат да влязат в уебсайта. Предоставя нужната информация за всеки тип от „служебните“ потребители – версия на системата и нейни параметри на системните администратори, сигналите за повреди на техническата поддръжка и данни за настъпил инцидент на наблюдателите.

2.7.3 Елементи на модула

2.7.3.1 Profile Manager

Този модул отговаря за различните групи служебни лица на системата и техните функционалности.

2.7.3.1.1 Administrators

Този модул дава специфични функционалности на администраторите.

2.7.3.1.1.1 System Maintenance

Този модул отговаря за преглеждане на текущото състояние и промяна на конфигурацията на системата.

```
public SystemInfo GetSystemVersion()  
  
public ServerInfo GetServerState()  
  
public void ShutDownSystem()  
  
public void PowerUpSystem()  
  
public void PushWebSiteUpdate()  
  
public void PushAppUpdate()  
  
public void PushUserManagerUpdate()
```

2.7.3.1.2 Observers

Този модул дава специфични функционалности на наблюдателите.

2.7.3.1.2.1 Incident Record

В този модул се съдържат записите за инциденти, които наблюдателите трябва да разгледат.

```
public Incident GetLastIncident()  
  
public Incident[] GetIncidents()
```

```
public UserProfile ViewUserDetails(string user_id)
```

```
public HistoryRecord ViewUserInfo(string user_id)
```

2.7.3.1.3 Technical Support

Този модул дава специфични функционалности на лицата по техническа поддръжка.

2.7.3.1.3.1 Signals

В този модул се съдържат сигнали за технически повреди, които лицата по техническа поддръжка трябва да поправят.

```
public SensorInfo[] ListFailures()
```

2.7.3.2 Access Manager

Този модул отговаря за влизане в системата на нейните служебни лица.

2.7.3.2.1 Authentication

Този модул отговаря за автентикация на потребителя.

```
public bool IsAuthenticated(string username, string password)
```

2.7.3.2.2 Authorisation

Този модул отговаря за авторизация на потребителя.

```
public bool CheckUserRights(string user_id)
```

2.7.4 Други

2.7.4.1 Ограничения при употреба

Достъпът през уеб сайта става чрез браузер, за което е нужна и връзка с интернет. Ако потребителят желае да получава нотификации за настъпили събития, е нужна да се даде разрешение в браузера на сайта да изпраща такива нотификации.

2.7.4.2 Грешки и изключения

При възникване на опит за достъпване на данните от неоторизиран източник, системата трябва да регистрира събитието, да блокира източника и да се сигнализира на наблюдателите на системата.

2.7.4.3 Зависимости от други елементи

Модулът получава информация за текущото състояние на системата от Server и към него се отправят заявки за спиране за поддръжка, пускане и обновяване на системата.

2.8 Database

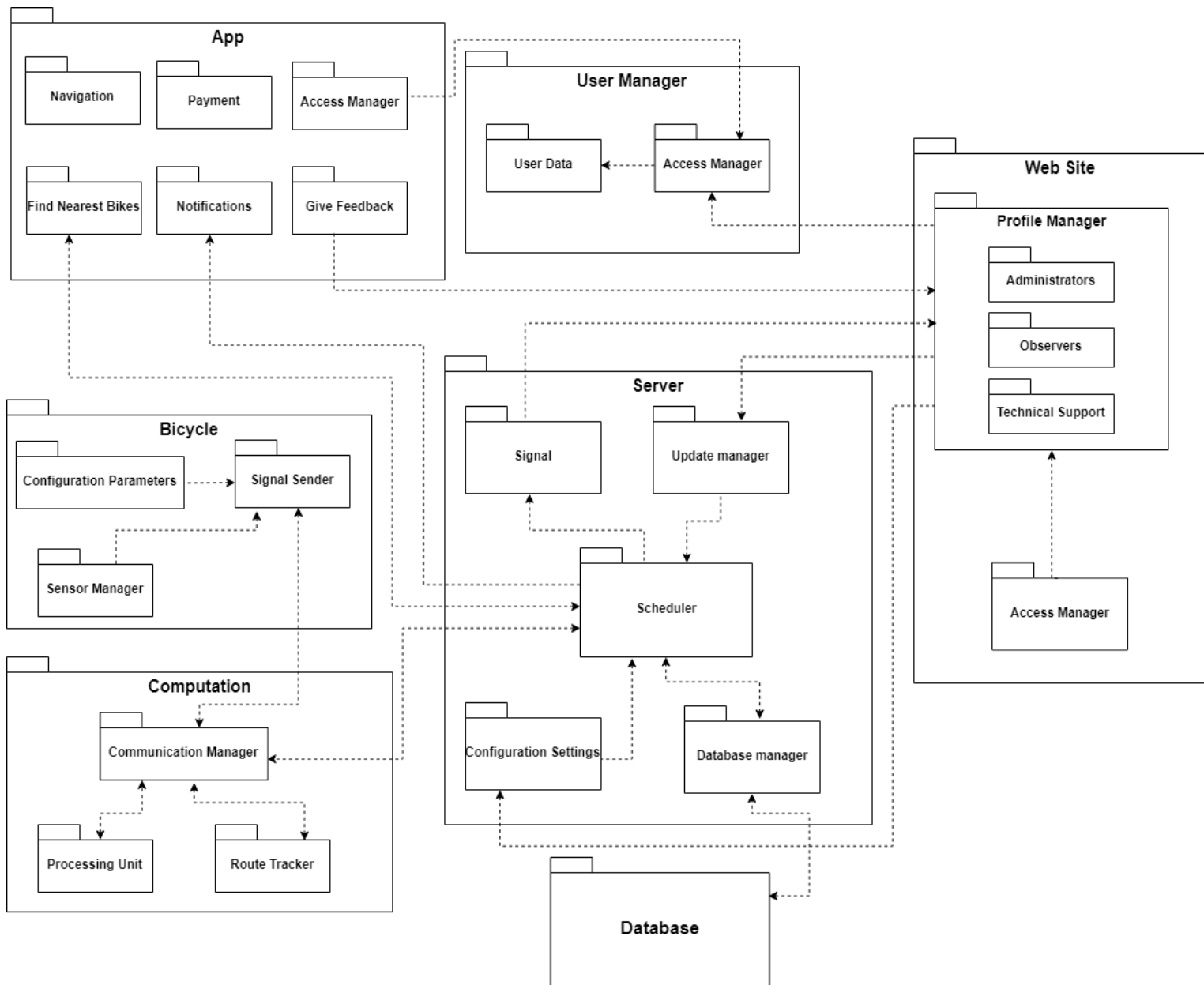
2.8.1 Предназначение на модула

База данни, която се съхранява информация за системата като например подробен отчет за всеки инцидент, данни за натовареността на системата, резервно копие на системата при срив, финансови отчети и др.

3.Описание на допълнителни структури

3.1 Употреба на модулите

3.1.1 Първично представяне



3.1.2 Мотивация за избор

Нашият мотив да изберем структурата „употреба на модулите“ като допълнителна структура за нашия проект е, че тя представя зависимостите между отделните модули в системата. Така може лесно да се определи при промяна в даден модул какво въздействие ще има в системата като цяло. В тази структура са пропуснати някои детайли, за да се представи по-общ и прост изглед над цялата система.

3.1.3 Описание на елементи и връзки

Елементите на структурата са основните модули на системата и техните подмодули, а връзките между тях описват комуникацията и обмяна на информация между тях.

3.1.3.1 Server

3.1.3.1.1 Scheduler

Този модул отговаря за разпределение и изпълнение на най-приоритетните задачи. Получава информация за извънредните събития от Computation и ги предава на Signal. Обработва заявки от потребителите за намиране на велосипеди от модул App и ги предава на Computation, който връща отговор за най-близките велосипеди, и посредством Scheduler се връща отговор на App. При обновяване на системата Update manager изисква от Scheduler време за прилагане на обновлението. През определен период от време предава сигнал на Database manager да направи резервно копие на състоянието и данните на системата. Той предава информация за обновление на конфигурационните параметри на системата на Computation или изпраща сигнал на App,

когато има пресрочване на допустимото време за използване на велосипеда.

3.1.3.1.2 Signal

При настъпване на извънредни събития той получава сигнал от Computation и отправя заявка към Scheduler за уведомяване на наблюдателите на системата, известяване на технически екип при повреда към модула Web Site или изпраща на сигнал към службите за спешна медицинска помощ.

3.1.3.1.3 Update manager

От модула Web Site получава заявки за обновление на системата и подава сигнал към Scheduler за извършване на обновление към останалите модули.

3.1.3.1.4 Configuration settings

От модула Web Site получава пакет с нови конфигурационни предмети и подава сигнал към Scheduler за прилагане на промените в системата.

3.1.3.1.5 Database manager

Получава заявки от Scheduler за промяна, добавяне и изискване на информация в базата данни на системата.

3.1.3.2 Computation

3.1.3.2.1 Communication Manager

Този модул извършва комуникация със Server и Bicycle, изпращайки информация от сензорите на велосипедите и информация за конфигурационните параметри на системата.

3.1.3.2.2 Processing Unit

Този модул служи за пресмятане на текущите показатели на велосипедите и сравнение с конфигурационните параметри на системата и при нарушение на някой от показателите изпраща сигнал чрез Communication Manager на Server, който да предприеме съответните действия.

3.1.3.2.3 Route Tracker

Този модул служи за изчисление на текущото положение на велосипедите чрез данните получени от Bicycle чрез Communication Manager.

3.1.3.3 Bicycle

3.1.3.3.1 Configuration Parameters

Този модул служи за съхранение на конфигурационните параметри и текущите показатели на велосипеда и чрез Signal Sender се изпращат и получават нови стойности.

3.1.3.3.2 Signal Sender

Този модул служи за осъществяването на комуникация между Bicycle и Computation.

3.1.3.3.3 Sensor Manager

Този модул служи за извличане на данни от смарт сензорите, свързани с велосипедите и ги изпраща чрез Signal Sender на Communication Manager-а в Computation.

3.1.3.4 Web Site

3.1.3.4.1 Profile Manager

Този модул осъществява функционалността на основните групи потребители. Лицата по техническа поддръжка

получават нотификации за необходими ремонти от Server. Наблюдателите на системата се известяват за нарушения по време на експлоатация на велосипедите от Server. Администраторите задават нови конфигурационни параметри и нови версии за обновление към Server. Също така администраторите могат да получат достъп до личната информация на клиентите чрез User Manager.

3.1.3.4.2 Access Manager

Този модул осъществява влизане в системата и даване на съответните правомощия на администраторите, лицата по техническа поддръжка и наблюдателите на системата.

3.1.3.5 Database

В този модул се съхранява информация за системата и резервни копия на данните. Комуникацията се осъществява чрез Database manager, който се намира в Server.

3.1.3.6. User Manager

3.1.3.6.1 User Data

Този модул служи за съхранение на личните данни на клиентите на системата както и тяхната история на използване на велосипеди. Тези данни могат да се достъпят единствено от администраторите чрез Web Site.

3.1.3.6.2 Access Manager

Във връзка с осигуряване на защита на личните данни на клиентите този модул служи за определяне на правата на достъпващите потребители от Web Site и App.

3.1.3.7 App

3.1.3.7.1 Navigation

Този модул служи за интеграция с външни системи за предоставяне на услуги за географски карти.

3.1.3.7.2 Payment

Този модул е връзка с външна система за осъществяване на плащания. Плащането не се осъществява в нашата система с цел по-висока сигурност на данните на потребителите.

3.1.3.7.3 Find Nearest Bikes

Чрез този модул потребителят може да изпраща заявки за намиране на най-близък велосипед към Server, който я обработва и връща намерени резултати.

3.1.3.7.4 Notifications

Този модул получава уведомления от Server при пресрочено време на използване на велосипеда.

3.1.3.7.5 Give Feedback

Този модул предоставя възможност за изпращане на обратна връзка.

3.1.3.7.6 Access Manager

Този модул служи за регистрация, влизане в системата и достъпване и редактиране на личните данни в модула User Manager, като това става след преминаване през мениджъра за достъп.

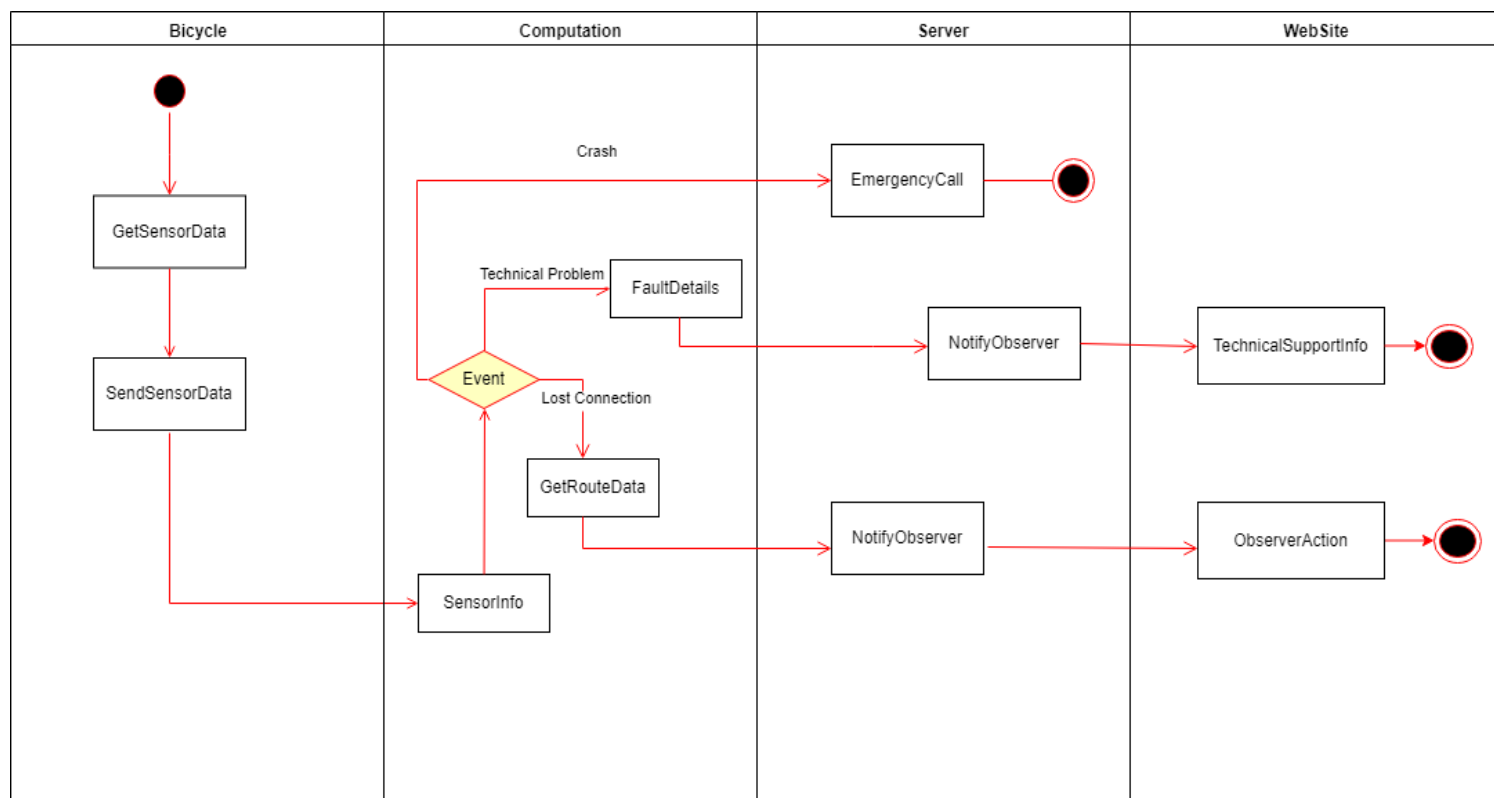
3.2 Структура на процесите

3.2.1 Мотивация за избор

Нашият мотив да изберем „структура на процесите“ като допълнителна структура за нашия проект е, че тя представя по лесен и разбираем начин протичането на основните процеси в системата. Избрали сме два от важните процеси, които протичат в системата: диагностициране на информацията, предавана от сензорите, при техническа повреда на велосипедите, при изгубване на сигнал или при настъпване на аварийна ситуация. Другият процес е промяна на конфигурационните параметри на системата, която се извършва от системните администратори. За всеки от двата процеса са представени диаграми, които описват основните действия на системата при определени условия.

3.2.2 Процес на диагностициране на информация от сензорите

3.2.2.1 Първично представяне



3.2.2.2 Описание на елементите и връзките

През определен период от време сензорите на велосипеда, които се намират в модула Bicycle, събират информация за техническото състояние на велосипеда и я изпращат към модула за извършване на изчисления Computation. Спрямо получената информация се определя типа на събитието като има три възможни случая.

- 1) В първия случай на загуба на връзка се взема информацията за маршрута на велосипеда в последните 30 минути и чрез модула Route Tracker, които използва AI се определя вероятното текущо

местоположение на велосипеда. След това се праща сигнал с висок приоритет към разпределителя на задачите(Scheduler) в модула Server, в който се извършва нотификация към наблюдателите на системата, които взимат съответните мерки за реагиране.

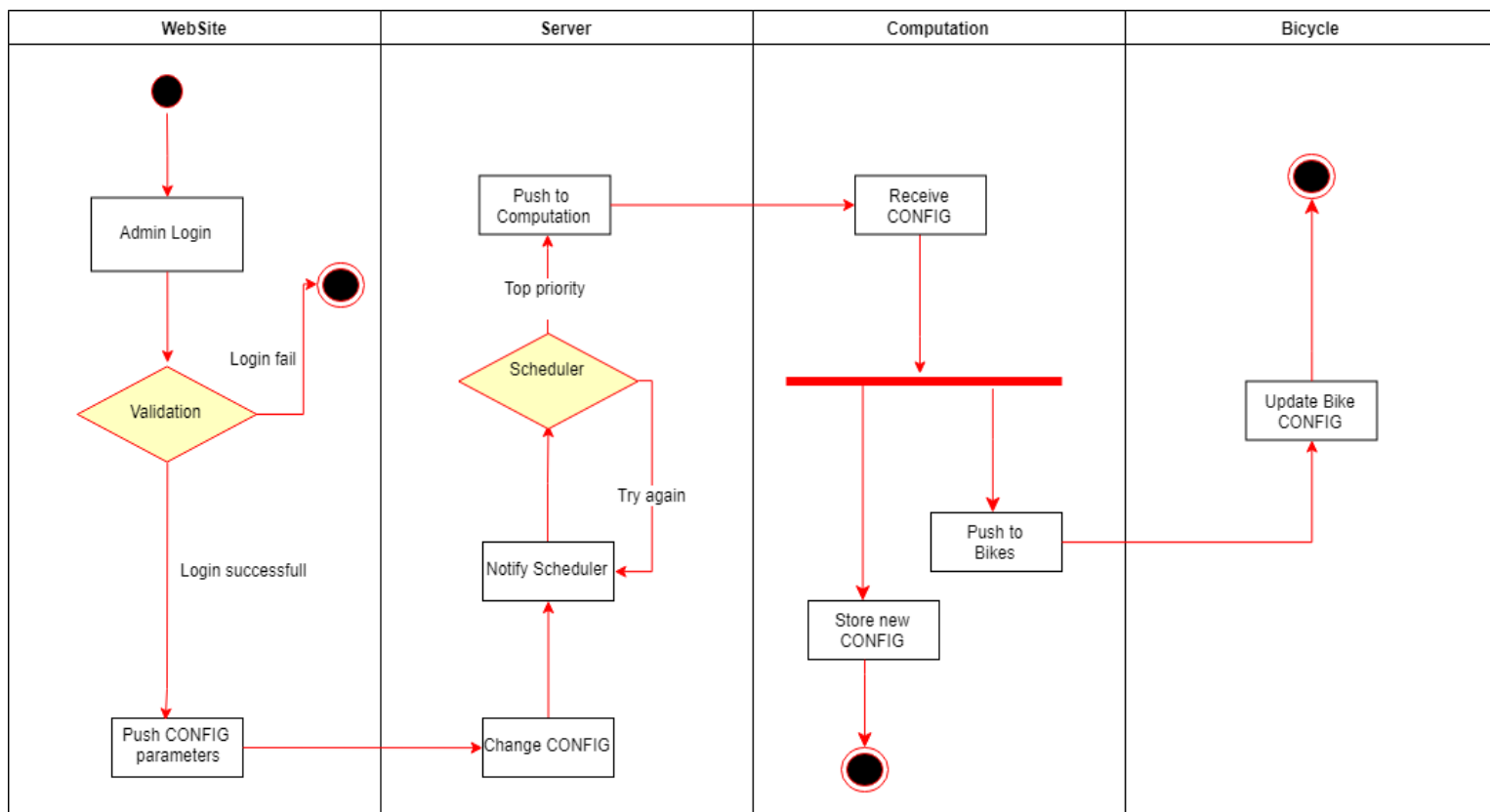
- 2) Във втория случай при наличие на техническа повреда се предоставя информация от смарт сензорите на екипите по техническа поддръжка чрез същата процедура.
- 3) В третия случай при настъпване на животозастрашаващо събитие се изпраща сигнал с най-висок приоритет към разпределителя на задачи (Scheduler) на модула Server, който уведомява модула Emergency Call, чрез който се известяват службите за спешна медицинска помощ.

3.2.2.3 Описание на обкръжението

При постъпване на информация за катастрофа системата се свързва с външна услуга за уведомяване на служби за спешна медицинска помощ, като се предават детайли за настъпилото произшествие, като време на загубване на връзка и приблизителна локация.

3.2.3 Процес на промяна на конфигурационни параметри

3.2.3.1 Първично представяне



3.2.3.2 Описание на елементите и връзките

Процесът започва след успешно влизане в администраторски акаунт през модула Web Site. Администраторът заявява новите конфигурационни параметри на системата и ги изпраща към модула Configuration Settings, който се намира в Server, и той от своя страна изпраща нотификация за промяната към разпределителя на задачи (Scheduler), който ако няма събитие с по-висок приоритет(като например катастрофа, техническа повреда, загуба на сигнал), предава конфигурационните настройки към модула за пресмятане Computation. В Computation се записват новите стойности на

параметрите в самия модул и също така се обновяват показателите в самия модул за велосипедите.

3.2.3.3 Описание на обкръжението

Обновяването на конфигурационните параметри е процес с висок приоритет в системата, но ако настъпят събития като загубване на връзка с велосипед или катастрофа, те ще бъдат събитията, които ще се пуснат да се обработват първи от Scheduler-а в Server и към Update Manager-а ще бъде пуснато съобщение да опита отново заявката си след определен период от време.

4.Архитектурни драйвери

4.1 Функционални изисквания

- **Системата трябва да 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)**
 - д. Наблюдател/отговорник по използването на велосипедите**Това изискване е важно, защото показва всички основни категории потребители, които ще използват системата. То обособени са основните роли и правата на всяка група при взаимодействие с приложението.
- **Максималното време T 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 най-близката стоянка и да го замени с друг ако му е необходимо.**

Изискването е важно, защото то гарантира, че ще няма да настъпи прекалено дълго заемане на ресурсите.

- 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 изискване е важно за използваемостта на приложението, тъй като използването на онлайн услуги за географски карти значително улеснява планирането на маршрута и ориентирането по време на пътуването. По този начин потребители, които използват тези услуги, когато шофират кола, ще могат да интегрират своите настройки и най-чести маршрути.

4.2 Качествени изисквания

- **защитеност (security)**

- i. Личните данни на потребителите трябва да са абсолютно защитени от външна намеса. Достъпни са единствено до наблюдателя на правомерното използване на велосипедите.

Това е изискване е ключово, тъй като то гарантира спазване на регулациите за защита на личните данни на потребителите.

- **използваемост (usability)**

- i. 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).

Чрез реализиране на това изискване се предоставя гъвкавост по отношение на начините на заплащане и така увеличава използваемостта.

- **безопасност (safety)**

- i. 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.

Изискването е важно, тъй като чрез него се гарантира, че само изправни и надеждни велосипеди ще могат да бъдат наемани и така се осигурява сигурност на наемателите на велосипеди.

- ii. 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.**

Това изискване е важно, защото то може да предотврати евентуални злополуки и неправомерно използване на велосипедите.

- **производителност (performance)**

- i. 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.**

Изпълняването на изискване е изключително важно, защото осигурява по-бърза реакция при злополука и по този осигурява висока сигурност на наемателите на велосипеди.

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

Системата трябва да изпълнява това изискване, защото то пряко засяга бързодействието на системата и е предпоставка за по-добро потребителско изживяване

- **наличност (availability)**

- i. 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.

Изискването е важно, защото осигуряването на висока наличност на системата е необходимо условие за висока надеждност.

- **преносимост (portability)**

- i. Системните администратори и наблюдателите използват системата през Уеб приложение.

Осигуряването на това изискване предоставя платформена независимост, което ще улесни достъпа на системните администратори и наблюдателите.