

# GoSmart

Изготвили: Леонид Шубин 1MI0600120, Димитър Илиев 7MI0600001

## 1. Въведение

### а) Обща информация за текущия документ

- i. **Предназначение на документа:** Този документ демонстрира архитектурата на приложение за наемане на електромобили “GoSmart”.
- ii. **Описание на използваните структури на архитектурата.**

*Декомпозиция на модулите* – представя логическо разделение на модули и подмодули в приложението. Основните модули са Presentation Layer, Service Layer, Security Layer и Database Layer.

*Изглед на внедряването* - показва конфигурацията на компонентите за обработка по време на изпълнение и компонентите, които живеят върху тях. Моделира физическите аспекти на обектно-ориентирана система. Показва постигането на изискванията: системата да е 99,999% налична, системата да е устойчива към пикови натоварвания в най-натоварените в денонощието, часове за придвижване.

*Изглед на процесите* – визуализира ключовите процеси на системата. Показва как и в какъв ред групи от обекти работят заедно. Демонстрира функционалности “Наем на електромобил”, “Регистрация”, “Активиране на Премиум профил” и “Обслужване на електромобили”.

### iii. Структура на документа

- 1. Въведение
  - a. Обща информация за текущия документ
    - i. Предназначение на документа
    - ii. Описание на използваните структура на архитектурата
    - iii. Структура на документа
  - b. Общи сведения за системата
  - c. Терминологичен речник
- 2. Декомпозиция на модулите
  - a. Общ вид
  - b. Контекстна диаграма
  - c. Описание на модули
- 3. Описание на допълнителните структури
- 4. Архитектурна обосновка

## **b) Общи сведения за системата**

GoSmart представлява софтуерна система, предназначена за локализиране и наемане на пътнически и товарни електромобили в рамките на град София, България.

Основната цел на системата е потребителите през десктоп или мобилно приложение да се информират за най-близките свободни електромобили и да ги наемат с цел да се придвижват само в рамките на града или да превозват товари със специално, предназначени за тази цел тежкотоварни електромобили. Електромобилите са снабдени с информационни системи, даващи пълна информация за състояние, местоположение, статус и т.н.

## **с) Терминологичен речник**

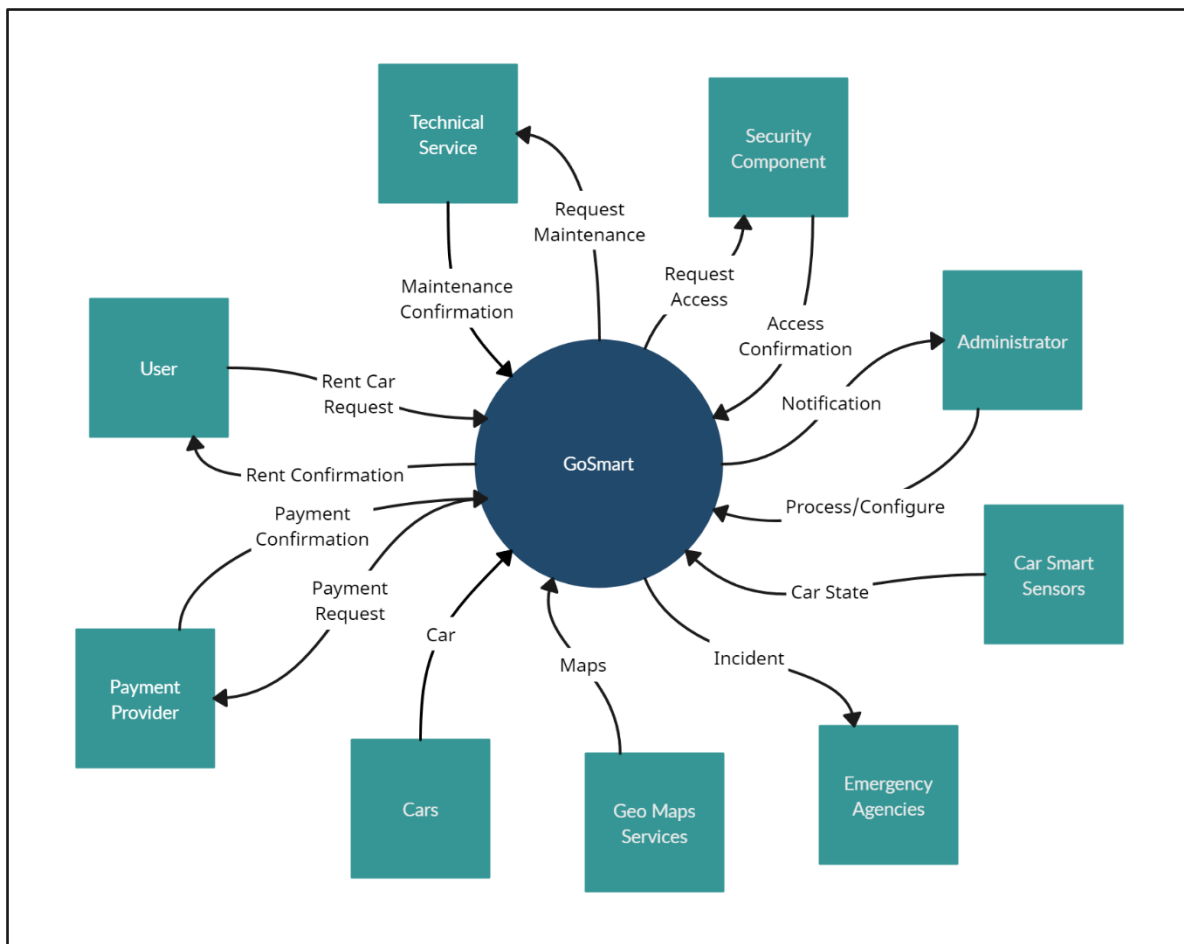
- API – Application Programming Interface. Представлява механизъм позволяващ на два или повече софтуерни компонента да комуникират помежду си.
- Клиенти - потребители на приложението.
- Обикновен потребител – потребител, който не е заплатил за премиум функционалностите.
- Премиум потребител – потребител, който е платил месечната такса и може да използва премиум функционалностите.
- Спешните Служба – включва Спешна Медицинска Помощ, Полиция, Пожарна безопасност и други.
- Технически Екип – съвкупност от различни служби за обслужване, зареждане и поддръжка на електромобили.
- Системен администратор - конфигурира системата и следи за правилната ѝ работа. Пренасочва мобилни екипи на съответната позиция за зареждане на батериите на изтощените електромобили.
- Оператор обслужване на клиенти – осъществяват връзки с потребители на приложението. Достъпват личните данни на потребителите.
- Bottleneck – точка на задръстване в производствена система, забавяща или спираща прогреса.
- И{0-26} – изискванията за проектираната система.

## **2. Декомпозиция на модулите**

### **а) Общ вид на декомпозицията на модули за системата**



## b) Контекстна диаграма



## c) Описание на модули

**Presentation layer:** отговорен за управлението на потребителския интерфейс (UI). Това е модулет, с който потребителите взаимодействат директно и е отговорен за представянето на информацията на потребителя. Презентационният модул действа като посредник между потребителя и другите слоеве на приложението. Той получава вход от потребителя и го превежда във формат, който другите модули могат да разберат. Подобно на това, той получава информация от другите модули и я представя на потребителя.

- Mobile Client UI: определя потребителския интерфейс на мобилната версия на приложението. Предоставя потребителския интерфейс за Android и IOS, в зависимост от операционната система на потребителя.
- Web Browser UI: определя потребителския интерфейс на уеб версията на приложението.

**Service layer:** съдържа бизнес логиката на приложението. Предоставя набор от услуги или API-та (Application Programming Interface), които могат да бъдат използвани от презентационния модул или други модули на приложението, за да достъпват и манипулират данни. Обслужващият модул действа като посредник между презентационния модул и модула за достъп до данни. Той получава заявки от презентационния модул, обработва ги с помощта на бизнес логика и връща резултатите на презентационния модул.

- Authentication: грижи се да разпознае кой е потребителят и дали този потребител има определени права да извършва дадена операция.
  - Access Manager
    - Permissions: управлява информацията за правата на потребители в приложението.
    - Groups: управлява информацията за ролята на потребител в приложението.
    - Premium Subscription Manager: следи месечните абонаменти на потребителите. Паузира достъпа до Премиум функциите, ако не се плаща месечната такса.
  - External Personal Info Provider: грижи се за предоставяне на лични данни (например ЕГН) от външни системи (например МВР).
  - User Info Manager: управлява потребителите, проверява паролите и хешира данните.
- Emergency Agencies Manager: грижи се за връзка със Спешните Служби в случаи на необходимост.
  - Emergency Report Manager: свързва се със органите на реда, изпраща нужната за тях информация.
  - External Emergency Agencies Provider: предоставя връзки с различни Спешни Служби.
- Car Rent: грижи се за избор и наем на електромобили
  - Search Manager: управлява търсенето на електромобили.
    - Search Engine: намира и предоставя списък с подходящи варианти, според зададените филтри и личните данни на потребителя.  
Interface ISearchEngine:  
public List<Car> FindCars(uint uid, Filter? filter)
    - Geo Maps Bindings: осъществява връзка с модул External Geo Maps Provider. Търсенето се осъществява според предоставени от модула онлайн услуги за географски карти.
- Payment Manager: Грижи се за правилното протичане по заплащането на наем за електромобил по трите възможни начина- чрез кредитна карта, смс или предварително закупени талони.
  - Payment Subscriber: абонира се към payment provider и получава информация за плащането на потребителя.
  - Payment Provider: предоставя сервис за извършване на онлайн плащане.
- Car Maintenance: обслужване на електромобили

- Technicians Contact Manager: управлява Техническите Екипи за ремонт и поддръжка на електромоделите според данните, събрани от модула Surveillance.
- Surveillance Bindings: връзки с модула Surveillance.
- Surveillance: грижи се за цялостното следене на електромодела.
  - Position Tracking Manager: следи текущото местоположение на електромодела. При загуба на връзка, възможното местоположение се определя чрез специални алгоритъм.
  - Car State Manager: предоставя актуална информация от смарт-сензори, с които е снабден електромоделът(ниво на батерия).  
Interface ICarState:  
public CarStats GetCarStats(uint carId)
  - Visual Supervision Manager: достъпва интериорното и екстериорното видеонаблюдение на електромодела.
- External Geo Maps Provider: онлайн услуги за географски карти.
- In-App Notifier: модул за вътрешните съобщения в приложението, улесняващи потребителя.  
Interface: public void Notify(string message, uint uid)

**Security layer:** междинен слой, разделящ Service layer и Database layer. Данните, минаващи през него се абстрахират и криптират, за да се избегнат загуби на лични данни.

- Database Encryption and Security Protocols: грижи се по какъв начин се криптират данните, как се декриптират, как се предават нагоре или надолу. Самото криптиране се извършва от същия модул.
- Database Bindings: връзки с бази данни от Database layer.
- Database Abstractors: предоставя слой на абстракция между приложението и физическата база данни, позволявайки на приложението да взаимодейства с данните, без да знае детайлите за това как се съхраняват.

**Database layer:** тук се пазят всички дълготрайни данни в системата.

- Accounts DB: съдържа информация за текущи потребители на системата и техните данни.

Application DB: съдържа останалата информация за правилно функциониране на системата. Например списък с налични електромодели или станции за зареждане.

### 3. Описание на допълнителните структури

**а) Структура на внедряването:** Ще демонстрираме архитектурата за изпълнение на система, включително взли като среди за изпълнение на хардуер или софтуер и междинния софтуер, който ги свързва.

Deployment структура визуализира начина за изпълнение на изискванията:

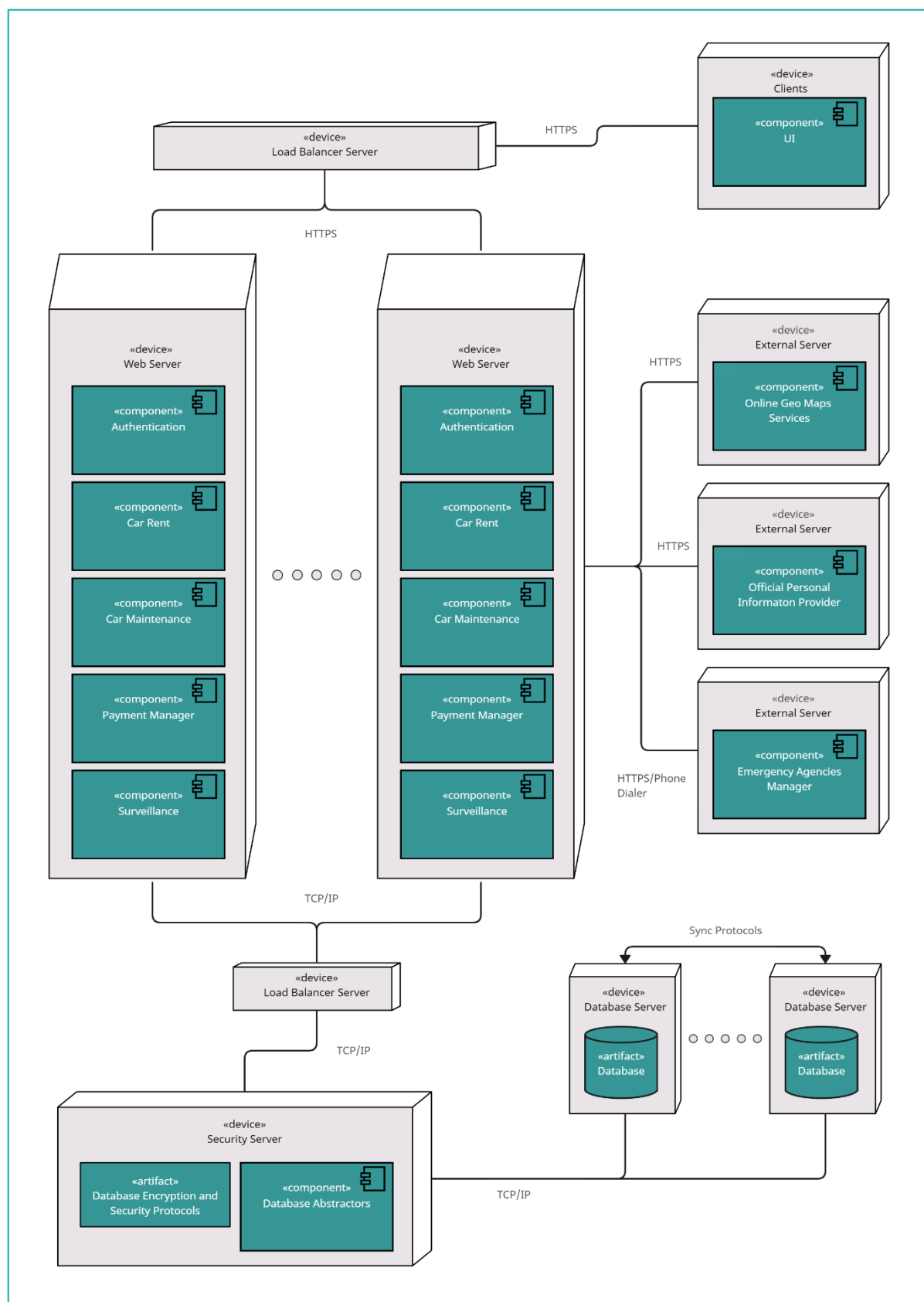
- Допуска се ремонт и профилактика в интервала от 3:00 до 4:00 ч. В останалата част на деня, системата трябва да е 99,999% налична.
- Системата трябва да е устойчива към пикови натоварвания в най-натоварените в денонощието, часове за придвижване.

Имаме 4 основни компонента, условно кръстени “Клиенти”, “Уеб сървър”, “Сървър на базата данни”, “Сървър за сигурност”, както и три допълнителни - “Географски карти”, “Лична информация”, “Спешни Служби”.

Компонентът “Клиенти” съдържа потребителския интерфейс на системата. Той комуникира с Уеб сървъра чрез HTTP заявки и получава отговори във формата на HTML страници. Използва се т.нар. Load Balancer - компонент, който разпределя входящия мрежови трафик към множество уеб сървъри с цел оптимизиране на производителността и подобряване на наличността. Той действа като мениджър на трафика, разпределяйки заявките от клиентите към група от задни сървъри.

На уеб сървъра се намират основните функционалности на системата. Осъществява се връзка чрез HTTP заявки с външни за системата услуги (Географски карти, Лична информация, Спешни служби). Не предвиждаме конкретен брой уеб сървъри, тъй като това зависи от конкретната реализация и брой потребители на системата. По този начин можем лесно да присъединим още сървъри, ако текущия брой не е достатъчен.

Базата данни се достъпва от уеб сървъра чрез TCP/IP протоколи. Първо заявките преминават през Security Server, който разрешава достъп към сървър с база данни. Както и при връзка Клиент-Уеб Сървър, имаме Load Balancer, който разпределя трафика към сървъри с бази данни. Отново не определяме конкретен брой сървъри, причините са същите както при Уеб Сървъри. Ще използваме вградени стратегии в системите за управление на бази данни, за да синхронизираме множество от сървъри с бази данни и осигурим backup на данните върху тях.

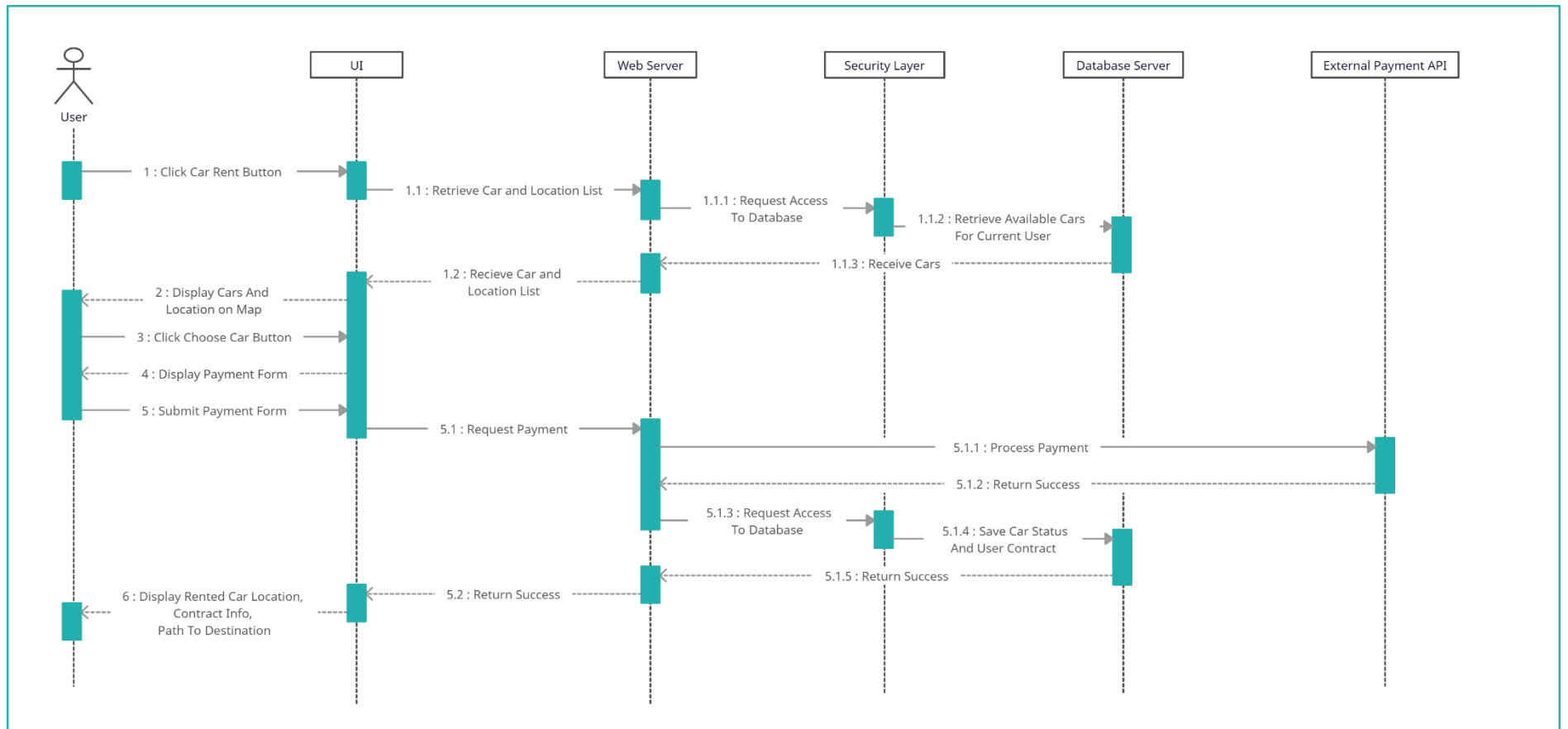




**b) Структура на процесите:** визуализира ключовите процеси на системата и показва как и в какъв ред групи от обекти работят заедно.

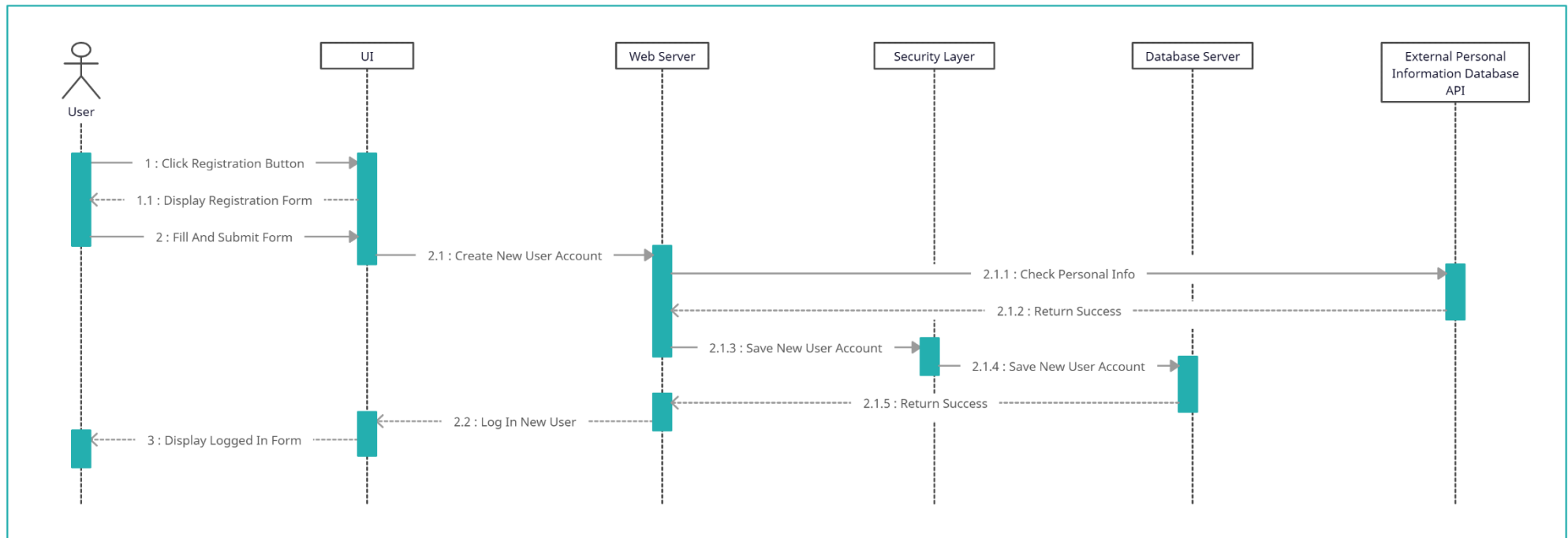
*Наемане на електромобил* - основна функционалност на системата без която приложението не може да функционира. Чрез нея ще се генерира по-голямата част от печалбите.

1. Потребителят натиска бутона “Наемане на електромобил”.
2. След изпращането на заявка към сървъра за получаване на списък с електромобили се случват следните действия.
  - a. След изпращането на заявка към сървъра за получаване на списък с електромобили се случват следните действия.
  - b. Компонента разрешава достъпа и предава заявка нататък към база данни.
  - c. База данни връща отговора директно към уеб сървъра.
  - d. Уеб сървъра препраща форматираните данни за коли и тяхното местоположение на потребителския интерфейс.
3. Потребителят вижда достъпни електромобили и информация за тях.
4. Потребителят разглежда опциите и натиска бутона “Избери електромобил”.
5. Интерфейсът изобразява формата за плащане.
6. След попълването на данни за плащане и изпращане на заявка към уеб сървъра се случват следните действия.
  - a. Уеб сървър изпраща заявка към payment provider-a.
  - b. В случай на успешната транзакция уеб сървъра поисква достъп до база данни.
  - c. Променя статуса на електромобил на “Зает” и записва информация за договора, сключен с потребителя.
  - d. Връща заявка за успешно плащане и запазване в база данни към уеб сървъра.
7. Показва на потребителя местоположението на електромобила, информация за сключения договор и пътя към избраната дестинация в избрани от потребителя онлайн услуги за географски карти.



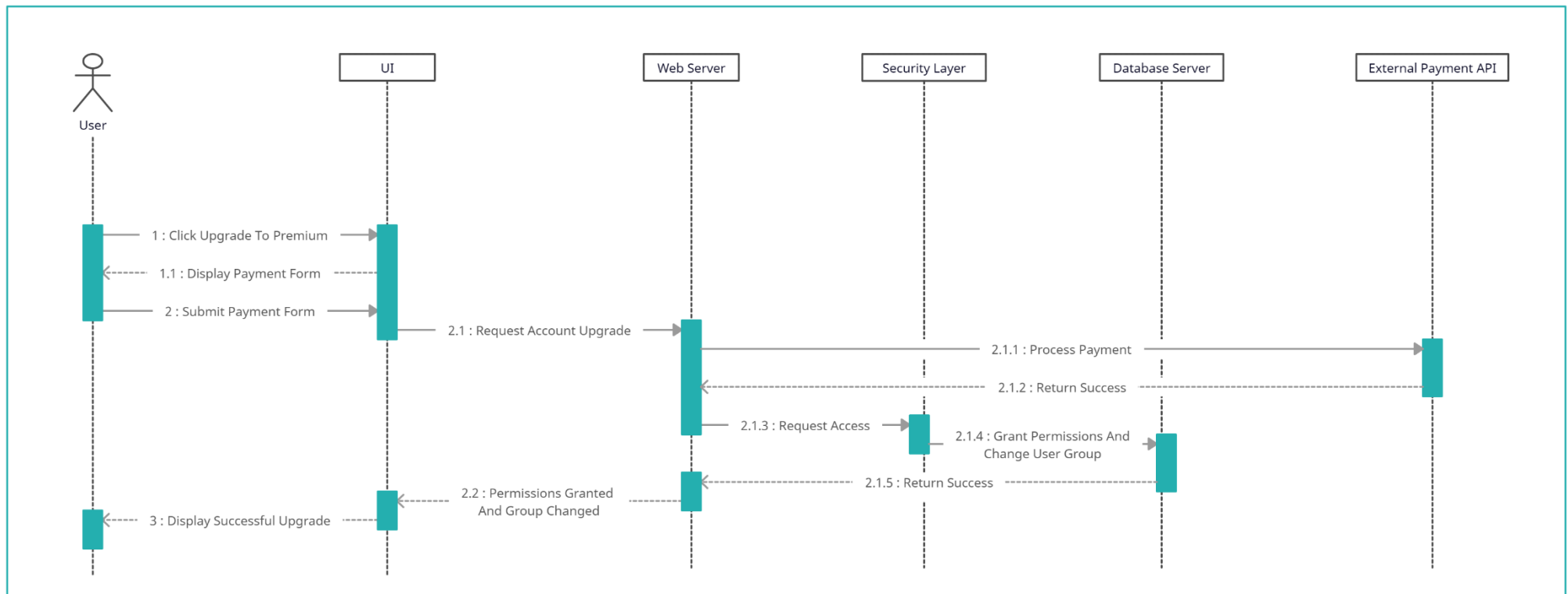
*Регистрация* - наред с “Наемане на електромобил”, е важна част от приложението, тъй като при нея се разпознават и проверяват лични данни на потребителя, както и се определя спектъра на достъпни електромобили.

1. Потребителят натиска бутона “Регистрация”.
2. Показва се форма за регистрация където потребителят вкарва стандартни данни(имейл, потребителско име, парола), така и допълнителни лични данни(имена, ЕГН, данни за връзка, данни и телефон за връзка на второ лице за контакт).
3. След като потребителят изпрати формата за регистрация, подава се заявка към уеб сървъра за създаване на нов профил и се случват следните действия.
  - a. Уеб сървър изпраща заявка за проверка на лични данни към външна база данни(например MBP).
  - b. Получава положителна обратна връзка, ако информацията удовлетворява изискванията.
  - c. В зависимост от въведената от потребителя информация създава обектен модел на профила му.
  - d. Уеб сървърът изисква достъп до базата данни от защитния компонент, за да запази новия профил.
  - e. Ако се случи успешно, се съобщава на потребителския интерфейс да отрази влизане в системата.
4. Интерфейсът показва на потребителя формата за вписан потребител.



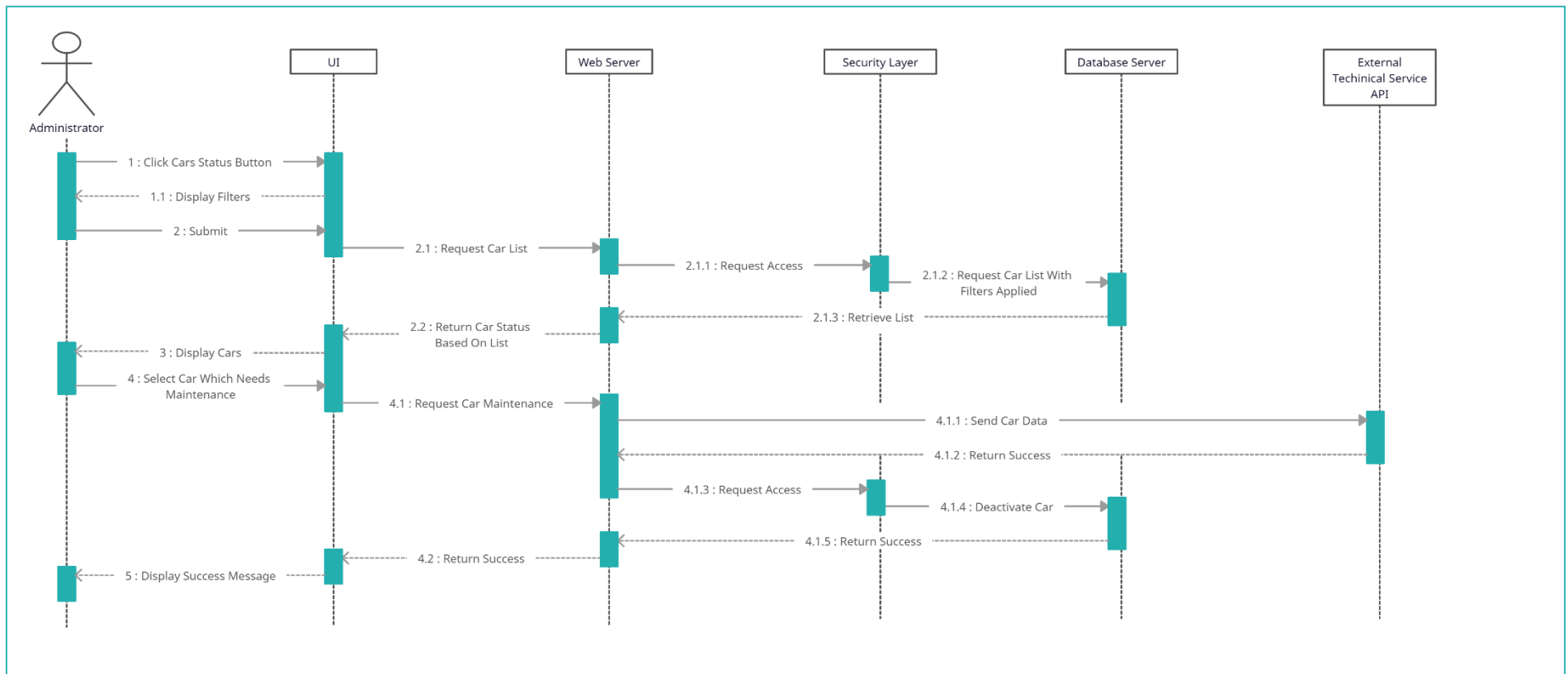
*Активиране на Премиум профил* - една от основните функционалности за генериране на печалба(привлича премиум клиенти)

1. Потребителят натиска бутона “Премини към Премиум”.
2. Показва се формата за плащане.
3. След попълването и изпращането на формата, се случват следните действия.
  - a. Уеб сървър изпраща заявка към payment provider-a.
  - b. В случай на успешната транзакция уеб сървъра поисква достъп до база данни.
  - c. Уеб сървърът изисква достъп до базата данни от защитния компонент, за да промени правата и да смени роля на потребителя.
  - d. При успешен завършек, уеб сървърът е уведомен.
4. Интерфейсът изобразява съобщение за успешна промяна на абонамента.



*Обслужване на електромобили* - показва как се извършва поддръжка на електромобили. При изтощаване на батерия, електромобил не може да се използва и се намалят опции за избор на клиентската база. Затова Системния Администратор следи състоянието на всеки електромобил и насочва технически екипи за зареждане.

1. Администраторът натиска бутона “Статус на колите”.
2. След изпращане на заявка за статуса на електромобили, се случват следните действия.
  - a. Уеб сървърът изисква достъп до базата данни от защитния компонент, за да получи данни за електромобили според зададени филтри.
  - b. Списъка се изпраща до уеб сървър, а в последствие и на потребителския интерфейс.
  - c. Администраторът решава дали има електромобил нуждаещи се от обслужване.
  - d. Изпраща се заявка до уеб сървъра за извикване на функционалността “Поддръжка на автомобили”.
  - e. В последствие уеб сървърът изпраща на външната система за сервиз информация за електромобил.
  - f. Сървърът изисква достъп до базата данни от защитния слой, за да промени статусът на “неактивен”.
  - g. При успешна промяна на статуса се уведомява уеб сървъра и потребителския интерфейс.
3. Потребителският интерфейс показва съобщение за успешно извършена операция.



## 4. Архитектурна обосновка

*И1* : Достъпът до системата трябва да може да се осъществява през браузър или чрез мобилен клиент за iOS и Android. Приложението трябва да може да работи в уеб среда и на мобилни устройства. При такъв вариант имаме различен вид клиенти, затова разделяме логиката на представяне от бизнес логиката. Таково разделение демонстрират Presentation Layer и Service Layer. Presentation Layer се декомпозира до Mobile Client UI и Web Browser UI, които представляват потребителския интерфейс за мобилни устройства и уеб среда. Service Layer съдържа цялата бизнес логика на приложението и е независим от Presentation Layer. По този начин действията, които извършва приложение са едни и същи, а потребителския интерфейс е различен в зависимост от средата. Ако в бъдещето ще има нужда приложението да работи в друга среда, например десктоп, добавянето ще бъде лесно заради това разделение.

*И2* : Системата поддържа различни роли потребители, всеки от които използват специфични нейни функционалности. Те са: Системен администратор (софтуерна поддръжка), Оператор обслужване на клиенти, Обикновен потребител, Премиум потребител. Трябва да се разделят функционалности им и се имплементират правила за сигурност, които ограничават един вид потребители да не достъпва функционалности на друг потребител. Във връзка с това е направен Authentication модула, а по конкретно Access Manager. Той управлява правата на потребител, както и следи subscription-и на Премиум потребители. Подмодули Permissions, Groups съдържат, както следва, правата на всеки вид потребител и видове профили в приложението.

*И3/И4* : За да използва системата, всеки потребител, задължително, трябва да се регистрира в системата чрез имейл, потребителско име и парола. В регистрацията задължително се попълват и следните лични данни: имена, ЕГН, данни за връзка, както и данни и телефон за връзка на второ лице за контакт при инцидент. Системата трябва да верифицира правоспособност за управление на МПС(електромобил) – наличие на шофьорска книжка и категория за съответното МПС за наемане. Authentication модул управлява потребителски профили. Той регистрира нови потребители и определя техните права и роли, по конкретно подмодула Access Manager. Подмодул User Info Manager се грижи за локалните профили и съхраняването на данните на потребителите. Проверка и верификация на личните данни на потребителя се извършва от External Personal Info Provider, който се грижи за интеграция с външна система за проверка на лични данни, както и права за управление на МПС, категория и други данни. Трите модула са относително независими, което ще улесни реализация, промени, тестване и поддръжка.

*И5* : В системата потребителят с профил „Премиум потребител“ достъпва всички функционалности на системата и има достъп до всички превозни средства с

приоритет над потребители с профили „Обикновен потребител“. Премиум достъп се управлява от подмодула Premium Subscription Manager. Той следи абонаментите, спира или дава права на потребители платили месечната такса. Върху този модул трябва да се обърне по-голямо внимание по време на проектирането и реализацията, тъй като това е един от начините на монетизиране на системата(ако има проблем, ще струва доста на бизнеса).

*И6/И7/И10/И18* : Системата трябва да поддържа информация за местоположение, тип, статус, възможен преход и т.н на всички електромобили в системата за наемане. Отговорността поема Surveillance модула. Подмодула Position Tracking Manager получава данни за местоположение на електромобил и ги визуализира. Получава данни директно от система за навигиране на електромобил, съответно има информация за избрана дестинация и възможен преход. Car State Manager управлява данни свързани с текущ технически статус на електромобил. Извършва се посредством смарт-сензори, с които е снабден всеки електромобил. Visual Supervision Manager предоставя възможност за визуално следене на електромобил, понеже всички електромобили са снабдени с интериорно и екстериорно видеонаблюдение.

*И8/И9* : При изтощаване на батерията на електромобила, се уведомява системния администратор и автомобилът преминава в статус на неактивен и не може да бъде наеман за превоз от потребителите. Подаването на сигнал се осъществява от Surveillance модула, а по конкретно Car State Manager подмодула. Смарт-сензорите пращат сигнал към него, а той активира In-App Notifier за да уведоми Системния администратор. Също така се праща сигнал до уеб сървър, а той праща заявка за промяна на статуса на електромобила. Подмодула Technicians Contact Manager праща данни на електромобил заедно със заявка за обслужване/зареждане до достъпни технически екипи.

*И11/И17* : Потребителите използват системата през десктоп или мобилно приложение да се информират за най-близките свободни електромобили и да ги наемат. Това се осъществява чрез Car Rent модула. Описано е в диаграма на процесите.

*И12* : В зависимост от правоспособността за управление на МПС, потребителите могат да наемат различни категории електромобили. Спектъра на достъпните електромобили се определя по време на регистрацията(И3/И4), като данните се верифицират от външен сървър с лична информация.

*И13* : Обикновените потребители могат да запазват пътнически и тежкотоварни електромобили до 3 дневен срок. При избиране на електромобил за наем се посочва срока за наем(до 3 дни), като информация за договора с потребителя се запазва. Потребителя се уведомява чрез In-App Notifier за изтичащ срок за наем.

*И14* : „Премиум потребителите“ заплащат месечен абонамент за използване на GoSmart системата, с приоритет за запазване на пътнически и тежкотоварни електромобили до 30 дневен срок. Тези потребители имат достъп до ексклузивна категория автомобили, достъпни само за тях, нар. серия „Лукс премиум“. Това са електромобили, характерни с висока себестойност и луксозен интериор и екстериор. Access Manager определя роля на потребителя, като информация се пази в Groups и Permissions. Съответно тези два модула определят дали потребителя има право и приоритет да ползва „Лукс премиум“ (дали му се показват когато търси електромобили).

*И15* : 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). Payment Provider в Payment Manager модула управлява плащанията. Със структура Subscriber/Provider постигаме абстракция върху различни начина за плащане, като това ще ни позволи лесно да добавим нови.

*И16* : Личните данни на потребителите трябва да са абсолютно защитени от външна намеса. Достъпни са единствено до потребител „Оператор обслужване на клиенти“. За да постигнем това използваме Security Layer компонент, който разрешава достъп до база данни чрез HTTPS протоколи, които от своя страна също криптират своите данни. Без защитния компонент сървърът с база данни няма да даде достъп до информация паеща се в него. С тези решение би могли да се предпазим евентуални атаки и загуба на лични данни, които се пазят в системата за правилното ѝ функциониране.

*И19* : При излизане на електромобил от рамките на града, трябва да се сигнализира „Оператор обслужване на клиенти“ и „Системен администратор“ в рамките на 1 мин, като се изпратят данни за движението на електромобил в последните 30 мин. Системата трябва да сигнализира наемателя за неправомерното напускане на позволения периметър за предвижване. Position Tracking Manager в Surveillance извършва сигнализарането, както и In-App Notifier съобщава на потребителя да неправомерни действия.

*И20* : При или загуба на връзка с даден електромобил трябва да се сигнализира „Оператор обслужване на клиенти“ и „Системен администратор“ в рамките на 10 сек като се изпратят данни за движението на електромобил в последните 30 мин, заедно с найвероятната му позиция на която се е намирал в момента на изгубване на връзката. Position Tracking Manager в Surveillance изчислява примерното местоположение чрез специален алгоритъм при загуба на връзка. In-App Notifier съобщава на „Оператор обслужване на клиенти“ и „Системен администратор“ за загуба на връзка.



*И21* : Системата трябва да поддържа връзка с органите на реда и спешни служби като МВР, Пожарна, Спешна помощ и т.н., при възникване на извънредни ситуации. За да постигнем това, ще се създаде Emergency Agencies Manager модул, който се декомпозира до Emergency Report Manager и External Emergency Agencies Provider. Emergency Report Manager позволява бързо да се подадат нужните данни(лични данни) на спешни служби при извънредни ситуации. External Emergency Agencies Provider интегрира външни системи за комуникация с спешни служби и органи на реда, които ще могат да се използват от Emergency Report Manager.

*И22* : При засичане на пътен или друг инцидент с електромобил, се изпраща автоматично сигнал до спешна помощ (112), в рамките на 1 сек след засичане на инцидента като системата изпраща точното местоположение на инцидента и данните на наемателя на МПС. В рамките на 5 сек се известява и „Оператор обслужване на клиенти“ и „Системен администратор“. Смарт-сензори изпращат сигнал за пътен инцидент до Car State Manager. Той активира модула Emergency Agencies Manager, а конкретно подмодула Emergency Report Manager за да изпрати сигнал до спешна помощ(112) с данни на потребителя. In-App Notifier ще уведоми „Оператор обслужване на клиенти“ и „Системен администратор“.

*И23* : 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. Модула External Geo Maps Providers интегрира външни онлайн услуги за географски карти. Предоставя избор на вече интегриран списък от услуги, с възможност да се добавят нови.

*И24/И25/И26* : Допуска се ремонт и профилактика в интервала от 3:00 до 4:00 ч. В останалата част на деня, системата трябва да е 99,999% налична. Прилагаме модифицирана three-tier архитектура. При нея трябва да имаме 3 отделни модула: Presentation Layer – отговаря за потребителския интерфейс, Service Layer – бизнес логиката, Database Layer – дълготрайни данни. За бази данни в Database Layer използваме вградени тактики за синхронизация и backup на данните. За да избегнем пренатоварването на отделни слоеве, между тях поставяме Load Balancer-и, които разпределят потока на заявки към различни сървъри. Броя на сървърите при Service Layer и Database Layer може да се увеличи или да се намали при необходимост(ако има bottleneck някъде). Всичко това е описано в deployment диаграма.