



# Софтуерна архитектура на уеб приложението „CityVox”



**CityVox**

Изготвили: Група № 8 (Алекс Хутев, Аркан Ахмедов, Дуйгу Дуран, Ивелин Панчев, Пенко Пенев, Преслав Гатев)

Преподаватели: Александър Петков и Ангел Нинов

## Въведение

Документация за софтуерната архитектура на „CityVox“ - уеб приложение, целящо да преодолее пропастта между местните власти и гражданите по отношение на инфраструктурните въпроси в техните населени места. Тъй като светът става все по-урбанизиран, подобна платформа е наложителна, за да се гарантира гладкото функциониране на градовете, да се информират жителите и да се отчитат местните власти. Този подробен документ има за цел да очертае монолитната архитектура на софтуера, като подробно описва различните му слоеве, техните функционалности и взаимодействието между тях.

## Актьори

### Крайни потребители/граждани

- **Роля:** Това са основните потребители на платформата „CityVox“. Те съобщават за проблеми с инфраструктурата, взаимодействат в сегмента на социалната мрежа
- **Необходимост:** Надеждна и удобна за ползване платформа за докладване на проблеми, взаимодействие с други граждани и получаване на актуална информация от представители на общината.

### Администратори

- **Роля:** Наблюдава валидирането на проблеми, докладвани от потребители, управлява потребителски акаунти и поддържа целостта на съдържанието в платформата.
- **Необходимост:** Надежден администраторски интерфейс за управление на заявки, преглед на проблеми и комуникация с потребителите.

### Представители на общината

- **Роля:** Ангажират се с потребителите, като предоставят актуална информация за дадени проблеми, отговарят на важни въпроси и правят официални съобщения, свързани с квартални проекти.
- **Необходимост:** Специален сегмент в платформата на социалните медии за комуникация с гражданите, който гарантира прозрачност.

### **Разработчици и технически екип**

- **Роля:** Отговарят за разработването, внедряването, поддръжката и мащабирането на „CityVox“. Те гарантират, че приложението е функционално, сигурно и отговаря на нуждите на потребителите.
- **Необходимост:** Ясна архитектура, насоки за разработване и механизми за обратна връзка, които да гарантират безпроблемното функциониране и непрекъснатото усъвършенстване на платформата.

### **Заинтересовани страни/инвеститори/възложители**

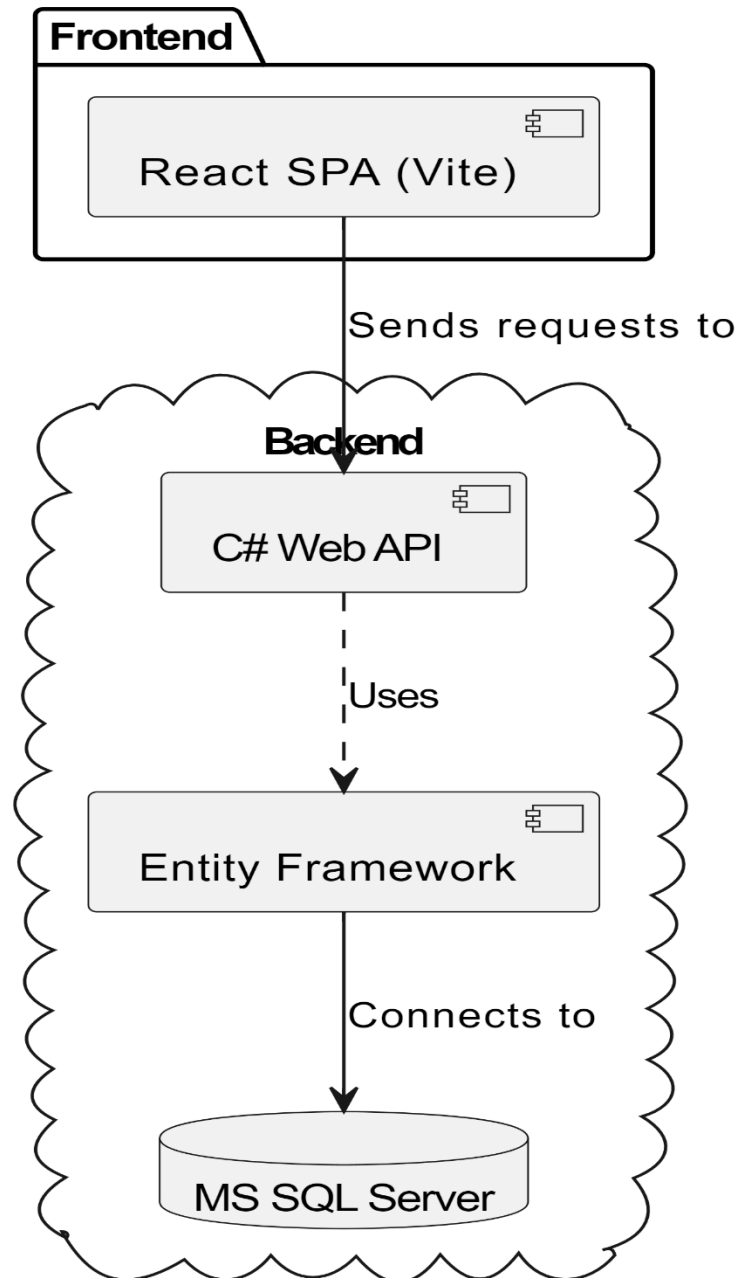
- **Роля:** Физически или юридически лица, които са инвестирали средства в проекта „CityVox“ и очакват той да бъде успешен и въздействащ.
- **Необходимост:** Редовно информиране за напредъка на проекта, показателите за ангажираност на потребителите и цялостната посока на развитие на платформата.

*Това въведение подготвя почвата за по-дълбоко навлизане в архитектурата, съображенията за проектиране и техническите особености на платформата „CityVox“. В следващите раздели ще бъдат описани подробно компонентите, техните взаимодействия и обосновката на архитектурните решения.*

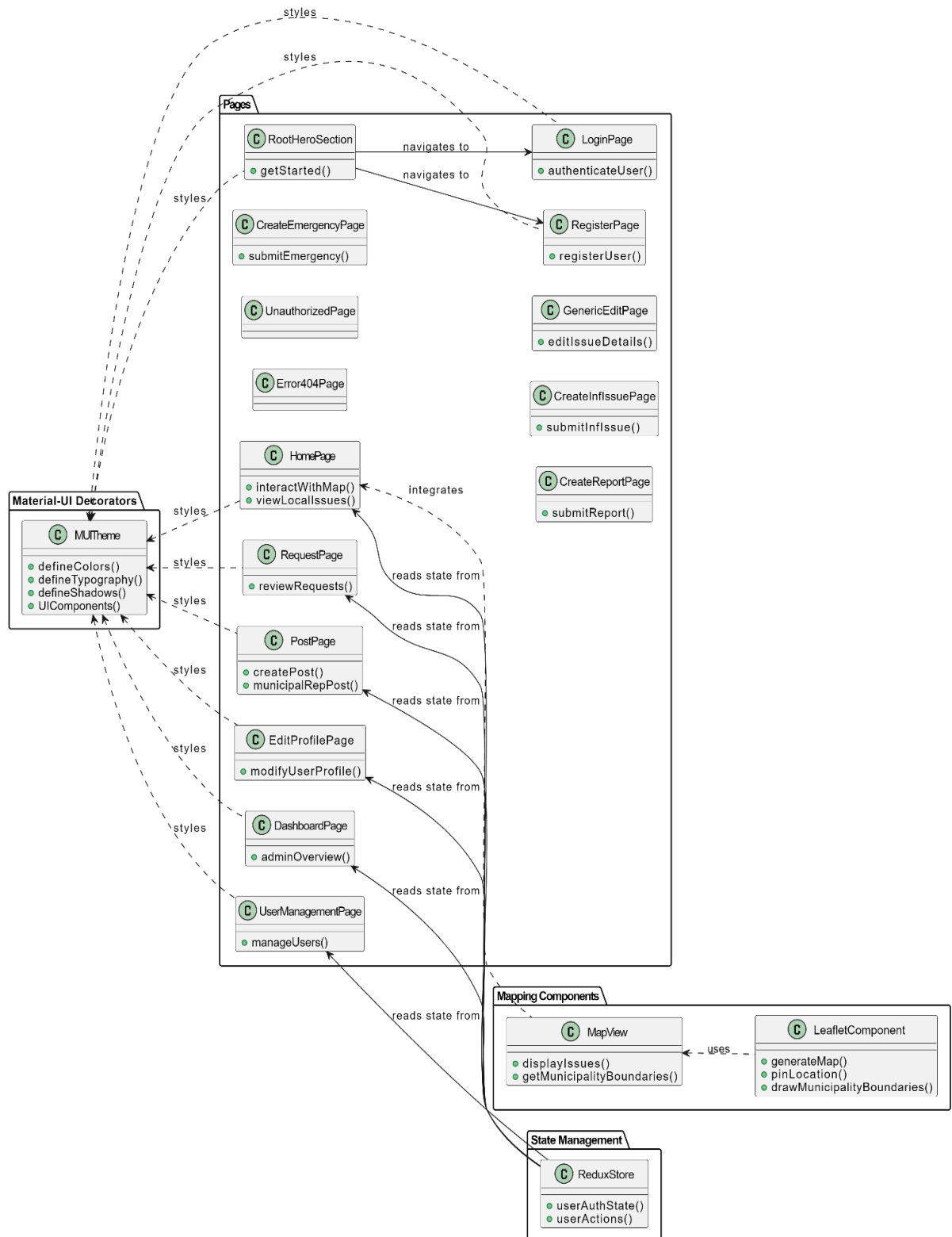
## Архитектурен обзор

### Логически изглед

Диаграмата предоставя архитектурен преглед от високо ниво на нашето уеб приложение, като описва основните компоненти и техните взаимоотношения.



- **Логически изглед на потребителския интерфейс**  
Диаграма:



## Описание:

Потребителският интерфейс ще бъде изграден с помощта на „ReactJS“ и ще включва различни компоненти и библиотеки, за да се осигури мащабируемост, повторна използваемост и лесна поддръжка.

### 1. Страници

#### 1) Целева/Първична страница (Root Hero Section Page)

- Това е компонентът, който приветства потребителите в приложението.
- Предлага на потребителите навигационни бутони за регистрация и вход в приложението

#### 2) Страница за регистрация (Registration Page)

- Позволява на нови потребители да се регистрират в приложението.
- Събира информация за потребителя и го регистрира в системата.

#### 3) Страница за вход (Login Page)

- Удостоверява съществуващи потребители.
- Приема потребителските идентификационни данни и ги валидира спрямо „бекенда“.

#### 4) Страница за несъществуващи пътеки (404 Error Page)

- Показва съобщение за грешка при несъществуващи маршрути или страници.

#### 5) Страница за неоторизирани пътеки (Unauthorized Page)

- Уведомява потребителите, когато се опитват да получат достъп до ограничена област без необходимите правомощия.

#### 6) Начална страница (Home Page)

- Основна точка на взаимодействие за потребителите за разглеждане на местни въпроси.
- Потребителите могат да взаимодействат с компонент на картата и да видят докладваните проблеми в тяхното населено място.

#### 7) Страници за докладване на проблеми (Create [Emergency/Report/InfIssue] Pages)

- Потребителите могат да докладват за конкретни видове проблеми.

8) Страница за редактиране на профила (Edit profile Page)

- Позволява на потребителите да променят данните на профила си.

9) Страница за постове (Post Page)

- Интерфейс, подобен на този в социалните мрежи, в който потребителите могат да обсъждат въпроси, а представителите на общината да публикуват актуализации.

10) Страници с ограничен достъп (Администратори)

- Специални интерфейси за администраторите за управление на приложението.

2. Карта (Mapping Component – Leaflet)

1) Изглед за карта (MapView)

- Интегрира библиотеката "Leaflet" за представяне на проблемите и границите на общините.

3. Управление на състоянието

1) Redux Store

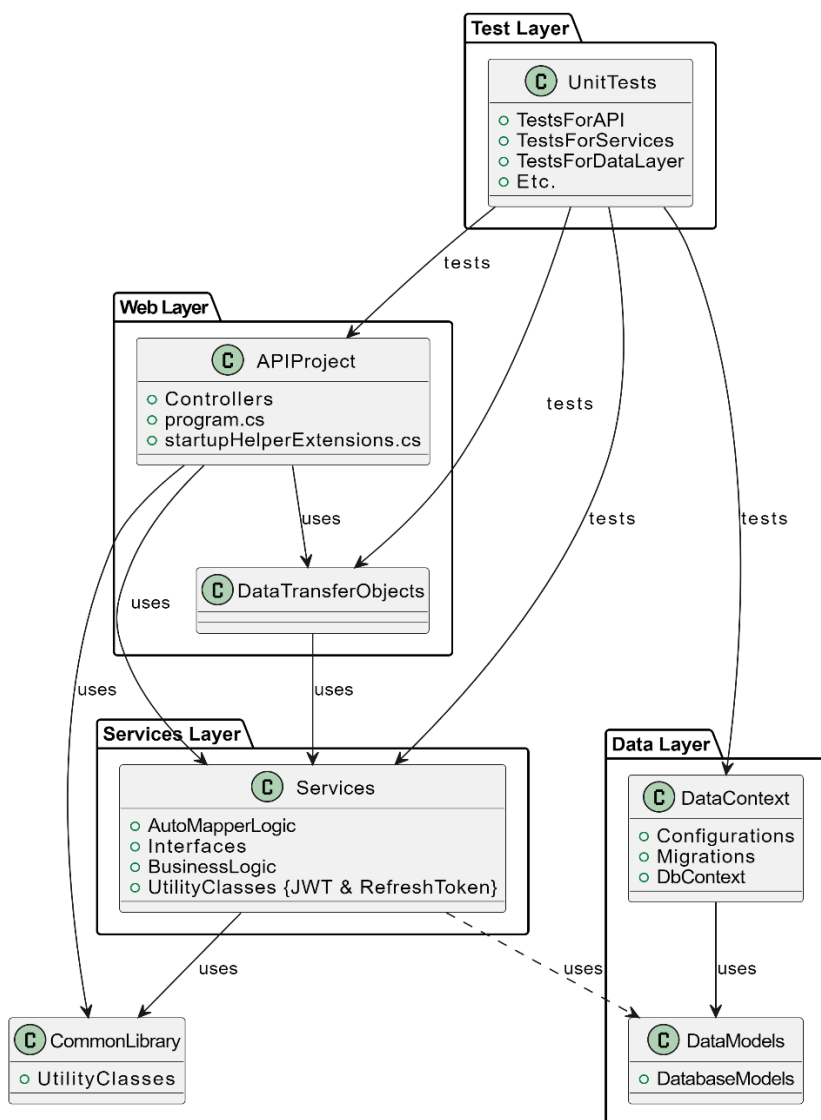
- Централизирано управление на състоянието на приложението.

4. Декорация и Material UI

1) Тема

- Задава цветове, шрифтове, сенки и предоставя предварително дефинирани компоненти на потребителския интерфейс, за да поддържа последователен вид и усещане в цялото приложение.
- Използва се библиотеката „Material UI”

➤ Логически изглед на Web API



Описание:

Диаграмата описва основните компоненти, техните отговорности и връзките между тях.

1. Уеб слой (Web layer)

- Арі Project: Това е главният интерфейс на приложението.
  - Контролери – Те обработват входящи уеб заявки и връщат отговори.
  - Program.cs – Основна входна точка на приложението



- **DataTransferObjects (DTOs)** : Това са обекти, които пренасят данни между процесите. Те се използват за прехвърляне на данни от APIProject към слоя на услугите и обратно.

## 2. Бизнес Слой/ Слой на услугите

- **Services**: Управлява основната бизнес логика и операции.

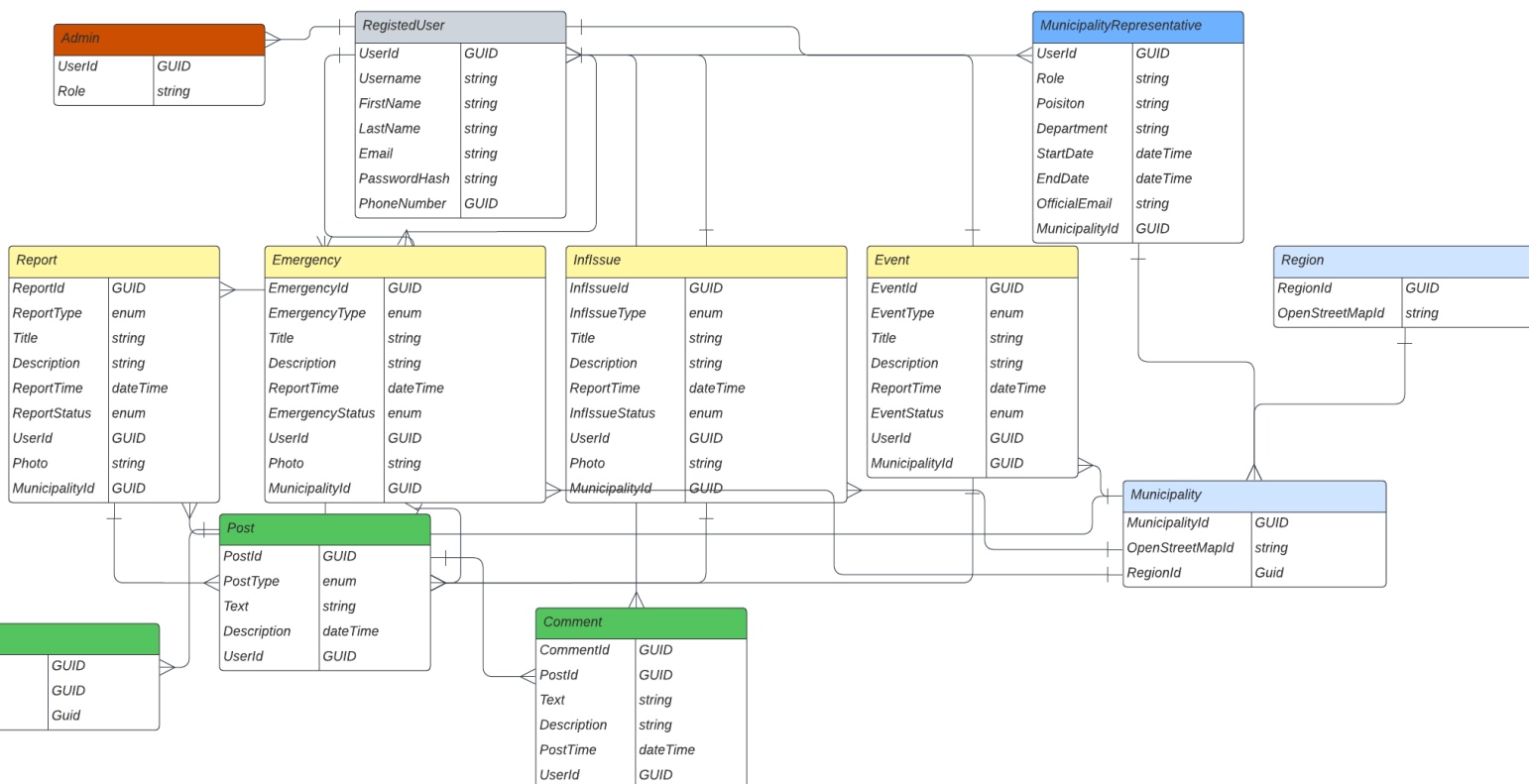
## 3. Слой на данните – отговаря за комуникацията с базата данни

## 4. Слой на тестовете – отговаря за тестването на различни компоненти от системата

# Изглед на данните

## Database Logic View Model

| October 9, 2023



## **Определяне на субекти:**

**Потребители:** Редовни участници в платформата, които съобщават за проблеми и участват в общи дискусии.

**Администратори:** Лица, отговорни за потвърждаването на докладваните проблеми и управлението на дейността на платформата.

**Представители на общините:** Упълномощени служители от различни общини, които се занимават с проблеми и предоставят официални актуализации на платформата.

**Issue(Проблем):** Общ термин, обхващащ различни видове проблеми, докладвани от потребителите.

**Report(Казус):** Стандартни въпроси или притеснения, докладвани от потребителите.

**Emergency(Спешни случаи):** Проблеми, които изискват незабавно внимание и разрешаване.

**Infissue (Инфраструктурен проблем):** Проблеми, свързани с физически инфраструктурни аспекти.

**Event (Събитие):** конкретни събития или случки в даден район.

**Община:** Административни структури, отговорни за разглеждането и решаването на проблеми в рамките на определена географска граница.

**Регион:** Географска област, която обхваща различни общини.

**Публикации:** Постове, направени от потребители или представители на общината с цел информиране, актуализиране или започване на дискусии в общността.

**Коментари:** отзиви или отговори, направени от потребители по публикации или въпроси.

**Гласове:** одобрения от страна на потребителите, свързани с определени въпроси или публикации, които помагат да се прецени колективната гледна точка на общността.

## Use-case изглед

Системата има три роли: Администратор, Представител на общината и Потребител.

### Случаи на използване от страна на Потребителя:

- Регистрация: Потребителите могат да се регистрират в приложението, като предоставят необходимата информация.
- Докладване на проблеми: Регистрираните потребители могат да подават доклади за различни видове проблеми (спешни случаи, инфраструктурни проблеми) с подробно описание и снимки.
- Редактиране на профил: Потребителите могат да управляват информацията в профила си, включително да променят профилните си снимки и да актуализират личните си данни.
- Преглед на одобрени доклади: Потребителите могат да разглеждат, коментират и харесват одобрени доклади чрез платформа, подобна на тази в социалните мрежи.
- Избор на регион: Потребителите могат да изберат своя регион, за да филтрират и разглеждат доклади в своето населено място.
- Взаимодействие с картата: Потребителите могат да определят местоположението на проблема на Leaflet картата, когато подават доклад.
- Известяване: Потребителите получават известие, когато докладът им е одобрен от администраторите.

### Случаи на използване от страна на Администратора:

- Регистрация: Администраторите могат да се регистрират в приложението, като предоставят необходимата информация.
- Докладване на проблеми: Регистрираните администратори могат да подават доклади за различни видове проблеми (спешни случаи, инфраструктурни проблеми) с подробно описание и снимки.
- Редактиране на профил: Администраторите могат да управляват информацията в профила си, включително да променят профилните си снимки и да актуализират личните си данни.
- Преглед на одобрени доклади: Администраторите могат да разглеждат, коментират и харесват одобрени доклади чрез платформа, подобна на тази в социалните мрежи.
- Избор на регион: Администраторите могат да изберат своя регион, за да филтрират и разглеждат доклади в своето населено място.

- Взаимодействие с картата: Администраторите могат да определят местоположението на проблема на Leaflet картата, когато подават доклад.
- Одобрение на докладите: Администраторите преглеждат подадените доклади и одобряват валидните, които след това, стават публично видими.
- Управление на потребителите: Администраторите имат право да управляват ролите на потребителите. Те могат да ги повишават в Администратори или Общински представители.
- Администраторско табло: Администраторите имат достъп до администраторско табло, чрез който извършват мониторинг на новопостъпилите и разрешените доклади.

#### **Случаи на използване от страна на Общинските представители:**

- Регистрация: Общинските представители могат да се регистрират в приложението, като предоставят необходимата информация.
- Докладване на проблеми: Регистрираните общински представители могат да подават доклади за различни видове проблеми (спешни случаи, инфраструктурни проблеми) с подробно описание и снимки.
- Редактиране на профил: Общинските представители могат да управляват информацията в профила си, включително да променят профилните си снимки и да актуализират личните си данни.
- Преглед на одобрени доклади: Общинските представители могат да разглеждат, коментират и харесват одобрени доклади чрез платформа, подобна на тази в социалните мрежи.
- Избор на регион: Общинските представители могат да изберат своя регион, за да филтрират и разглеждат доклади в своето населено място.
- Взаимодействие с картата: Общинските представители могат да определят местоположението на проблема на Leaflet картата, когато подават доклад.
- Известяване: Общинските представители получават известие, когато докладът им е одобрен от администраторите.
- Публикуване в общинската секция: Общинските представители разполагат със специален раздел за публикации, в който съобщават на местните жители за актуалните новини за текущите, предстоящите и успешно приключилите проекти за района.

#### **Случаи на използване от страна на сървиса за управление на автентикация:**

- CRUD операции: Сървисът за управление на автентикацията на потребителите може да създава, връща, изтрива нови потребителски роли, както и да обновява правата им.
- Управление на потребителски изгледи: Сървисът за управление на автентикацията на потребителите може да регулира достъпа до различните потребителски изгледи. Ако

потребителят не принадлежи към определената за изгледа роля, той не може да достъпи конкретния изглед (Връща се 401 (Unauthorized) грешка).

**Случаи на използване от страна на сървиса за управление на доклади:**

- CRUD операции: Сървисът за управление на доклади може да създава, връща, изтрива потребителските доклади.

**Случаи на използване от страна на сървиса за управление на картата:**

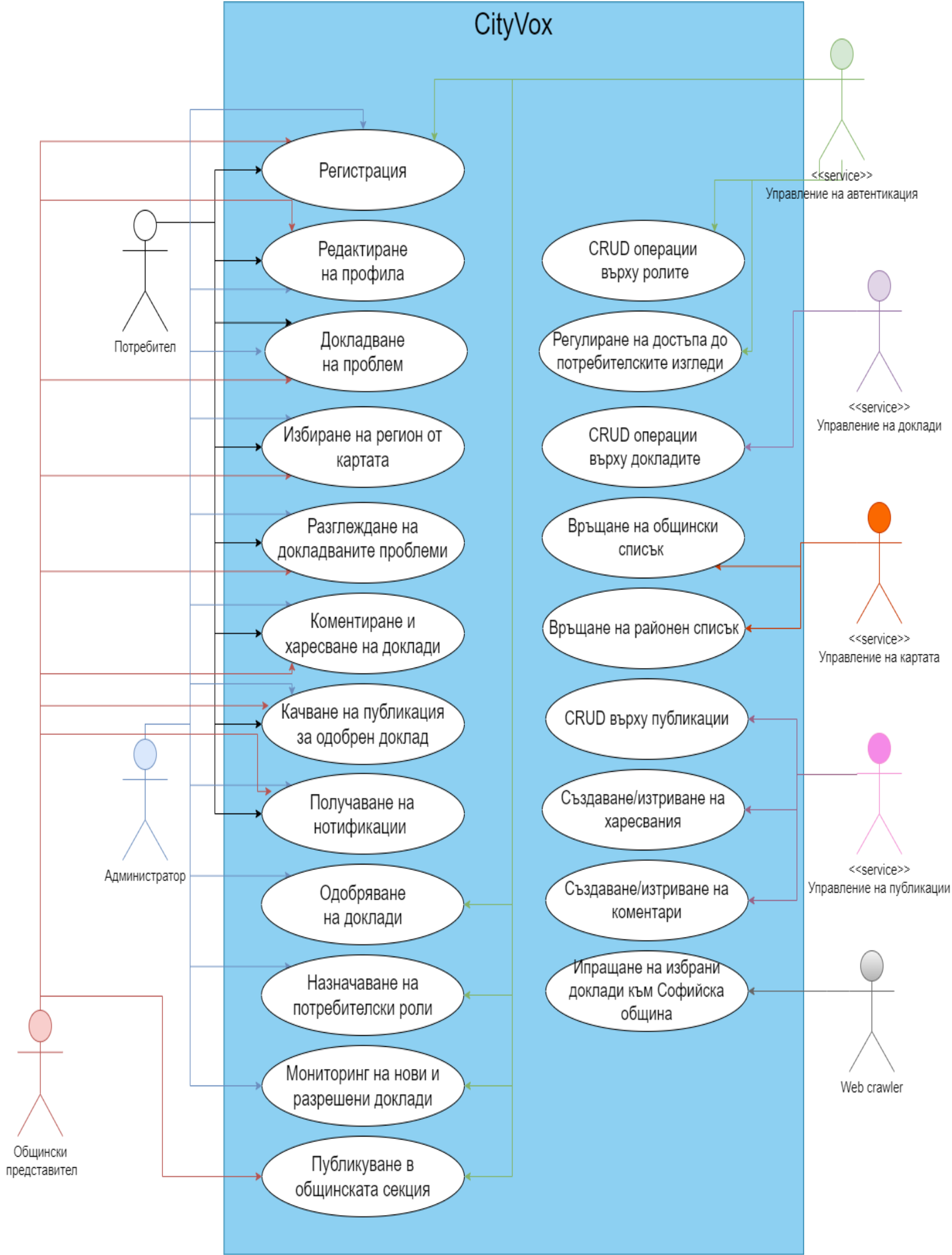
- Връщане на общински списък: Сървисът за управление на картата връща списък с всички райони, начертани върху картата.
- Връщане на районен списък: Сървисът за управление на картата връща списък с всички общински райони с еднакъв уникален номер за района.

**Случаи на използване от страна на сървиса за управление на публикации:**

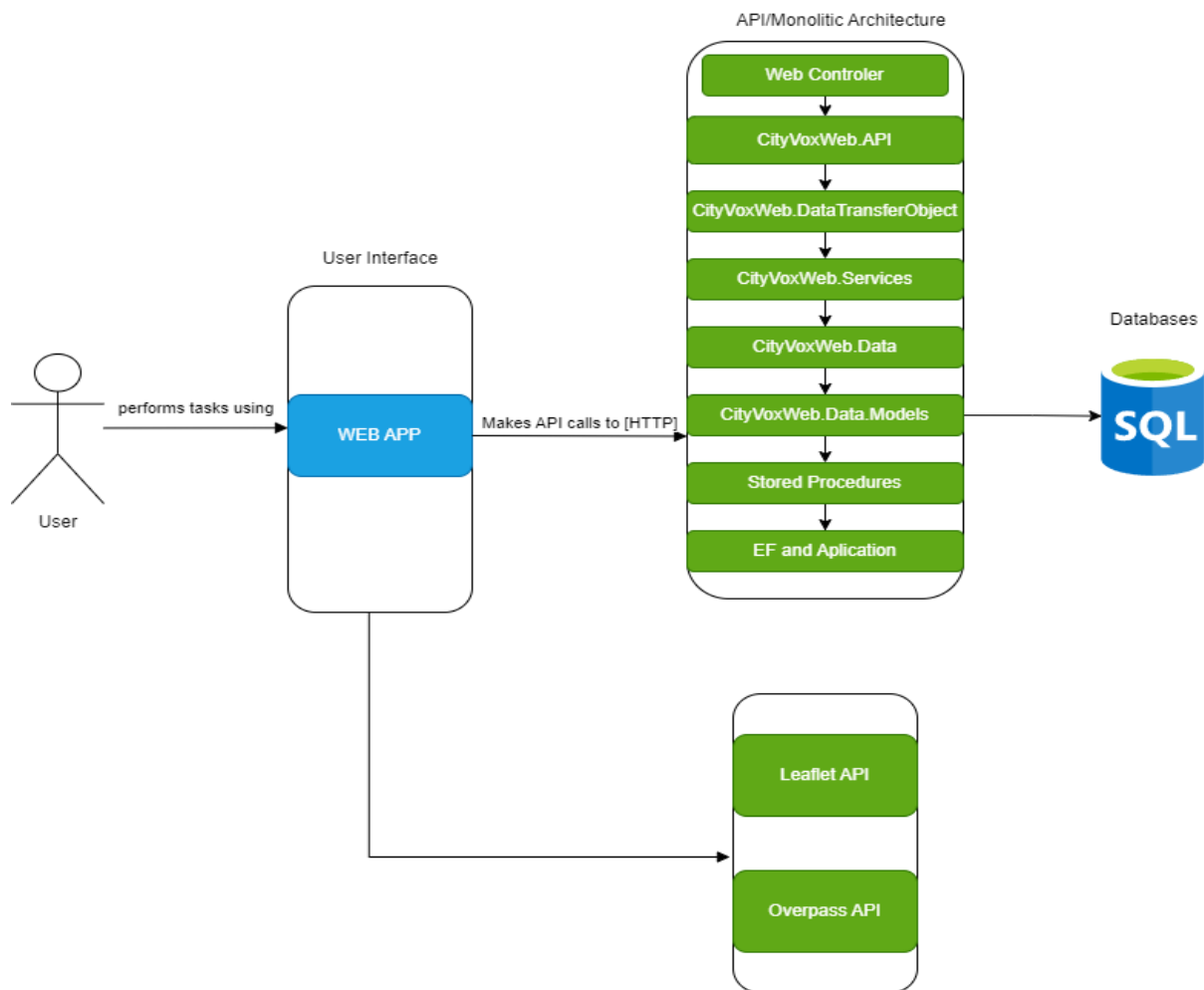
- CRUD операции: Сървисът за управление на публикации може да създава, връща, изтрива потребителските публикации.
- Създаване/изтриване на харесвания: Сървисът за управление на публикации може да създава и изтрива харесвания за дадена публикация при потребителска интеракция.
- Създаване/изтриване на коментари: Сървисът за управление на публикации може да създава и изтрива коментари за дадена публикация при потребителска интеракция.

**Случаи на използване от страна на Web crawler:**

- Ипращане на избрани доклади към Софийска община: Web crawler-ът може да изпраща, подбраните от администраторите за по-важни, доклади към Софийска община



## Изглед на Имплементацията



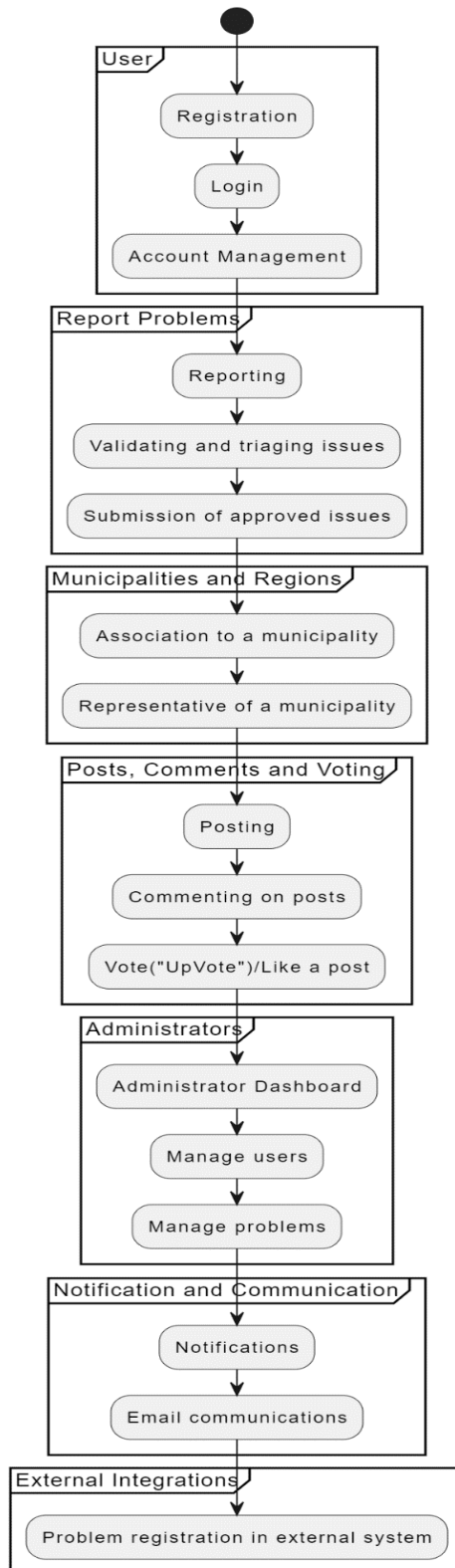
Обобщение - Тази диаграма представя структурата на имплементацията на CityVox. Системата се състои от три основни слоя: Интерфейсен слой, Бизнес логикаAPI и Достъп до данни. Всеки слой има определена роля и функция в системата. Интерфейсният слой обслужва потребителския интерфейс, бизнес логиката регулира потребителите и докладите, а слой за достъп до данни осигурява връзка с базата данни.

Описание на компонентите и слоевете

1. **Интерфейсен слой** - Този слой представлява уеб приложението и потребителския интерфейс на системата. Той позволява на потребителите да взаимодействат с приложението чрез браузъра. Интерфейсният слой комуникира с Бизнес логиката чрез REST API заявки.
2. **Бизнес логикаAPI** -Този слой съдържа бизнес логиката на системата и работи с моделите на базите данни.
3. **Достъп до данни**- Този слой представлява хранилището на базите данни.

## Процесен Изглед

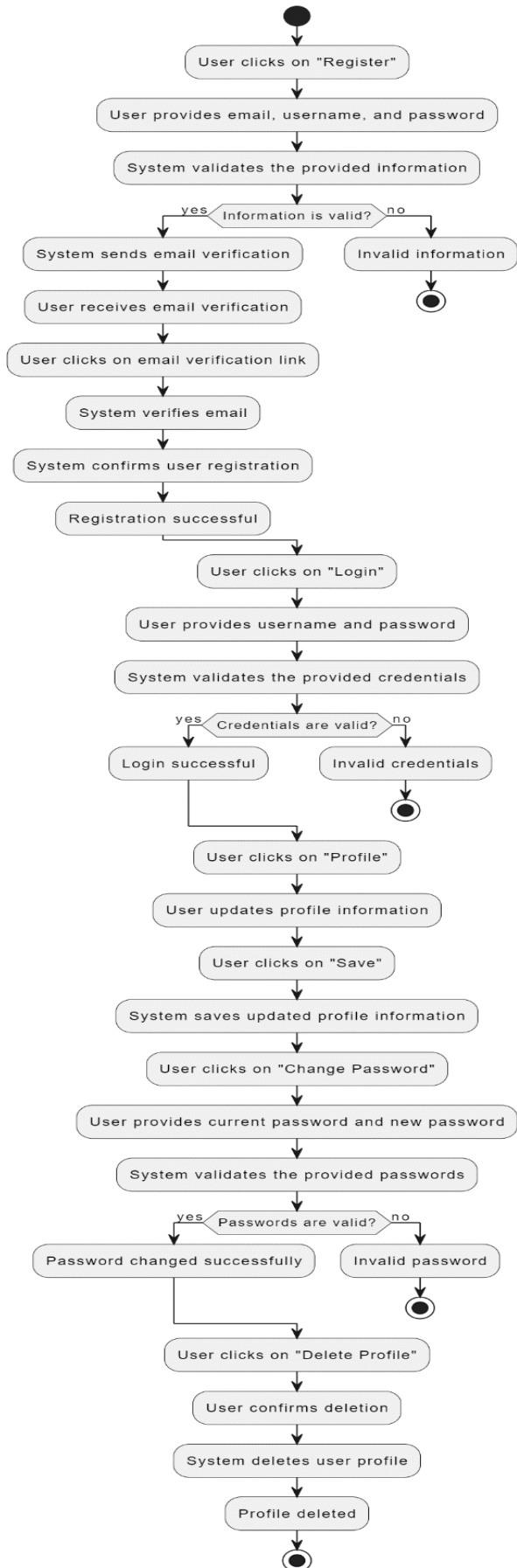
### ❖ Обща диаграма на активността





## 1. Потребител:

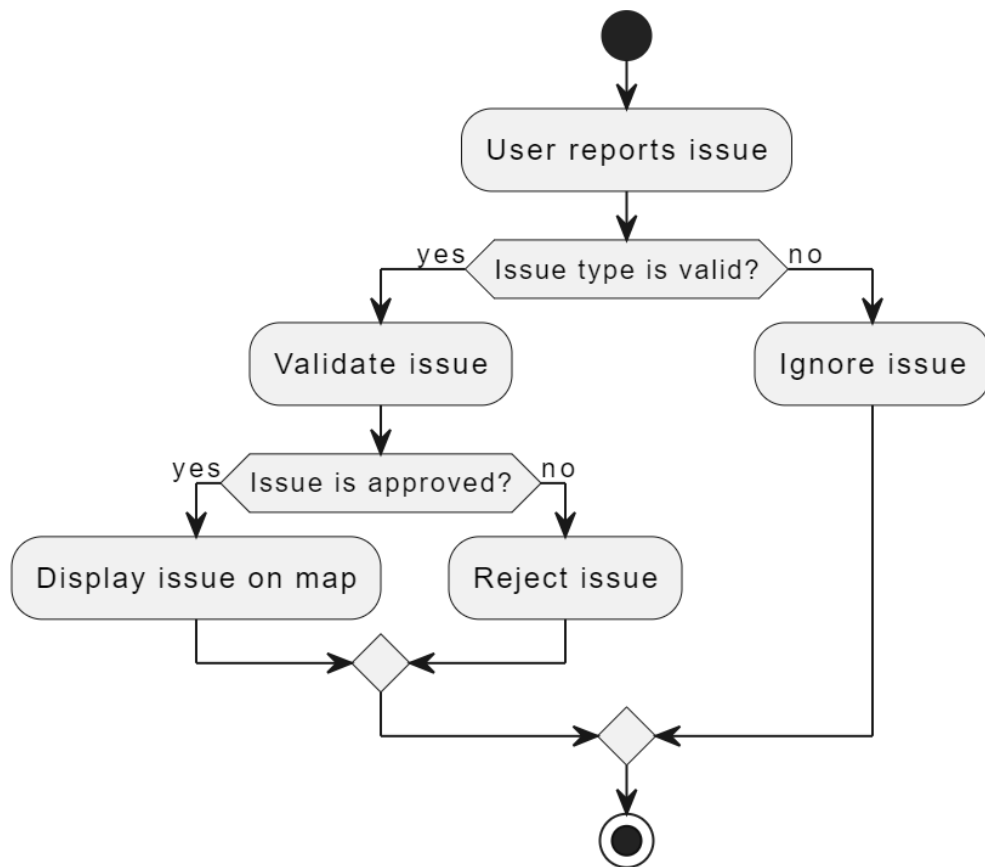
- ❖ Регистрация: Потребителите могат да създадат акаунт, като предоставят своя имейл, потребителско име и парола. Процесът на проверка по имейл потвърждава регистрацията на потребителя.
- ❖ Вход: Регистрираните потребители могат да получат достъп до системата, като използват своите идентификационни данни, като е налична и функция за възстановяване на паролата.
- ❖ Управление на профила: Потребителите могат да актуализират профилите си, включително да променят профилните си снимки и пароли. Те също така имат възможност да изтриват своите профили.



## 2. Докладване на проблеми:

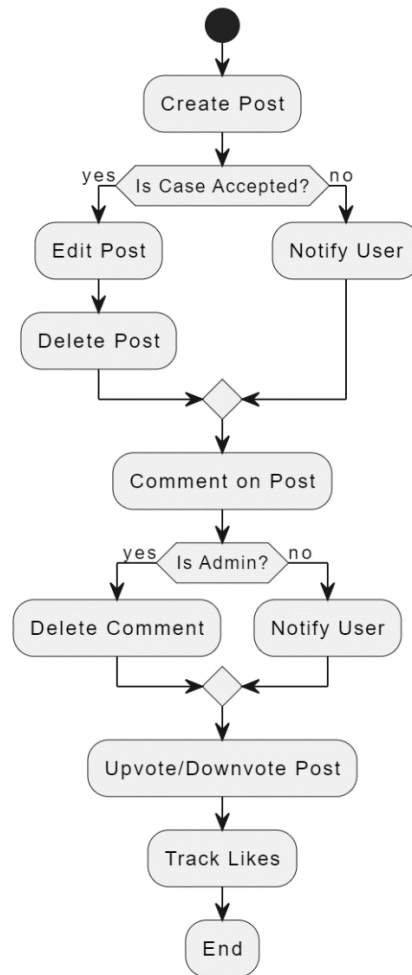
- ❖ Докладване: Потребителите могат да докладват за проблеми, като посочват типа (например "Доклад", "Авария", "Проблем с инфраструктурата", "Събитие"), предоставят заглавие, описание и местоположение чрез интерактивна карта.
- ❖ Утвърждаване и решаване на проблеми: Администраторите преглеждат, потвърждават и одобряват/отхвърлят подадените от потребителите проблеми.
- ❖ Подаване на одобрени проблеми: Одобрените проблеми се показват на картата и стават видими за всички потребители, стига те да са избрали съответната област и община.

Потребителят започва с докладване на проблем, като посочва типа, заглавието, описанието и местоположението му. След това системата проверява дали типът на проблема е валиден. Ако е валиден, проблемът се валидира и се преглежда от администраторите. Ако проблемът е одобрен, той се показва на картата. Ако въпросът не е одобрен, той се отхвърля. Ако типът на проблема не е валиден, системата игнорира проблема.



### 3. Публикации, коментари и гласуване:

- ❖ Публикуване: Потребителите и представителите на общините могат да създават, редактират и изтриват публикации, свързани с проблеми, актуализации или предстоящи проекти. Потребителите се нуждаят от одобрението на администраторите, за да създадат публикация.
- ❖ Коментиране на публикации: Потребителите могат да коментират публикации с цел обсъждане или споделяне на допълнителна информация. Администраторите имат право да изтриват коментари.
- ❖ Гласуване ("UpVote") / харесване на публикация: Потребителите могат да увеличават или намаляват гласовете за публикации. Системата проследява общия брой харесвания на даден пост, за да помогне на потребителите да идентифицират популярни въпроси.

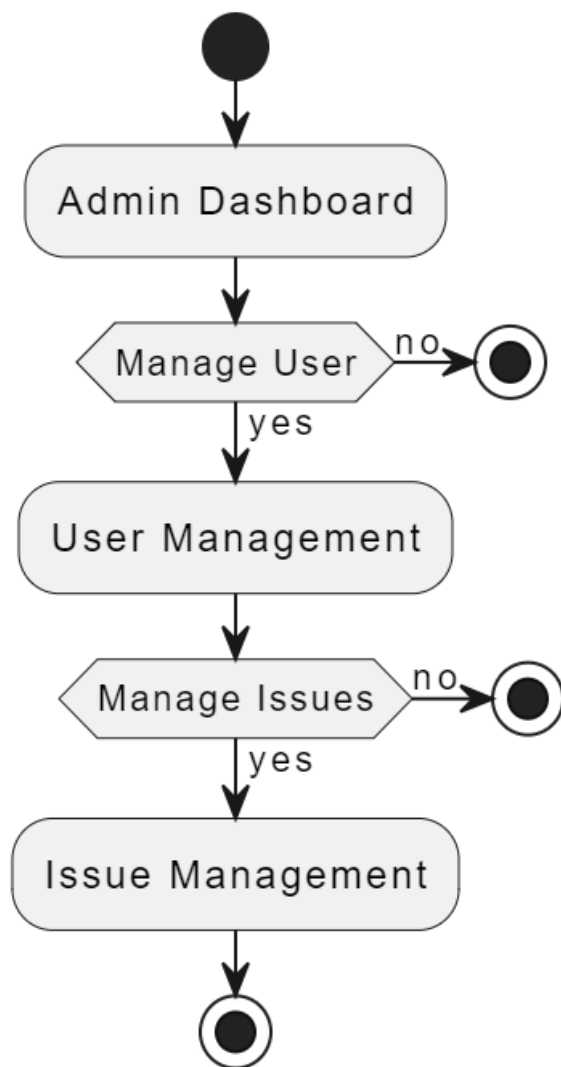


#### 4. Администратори:

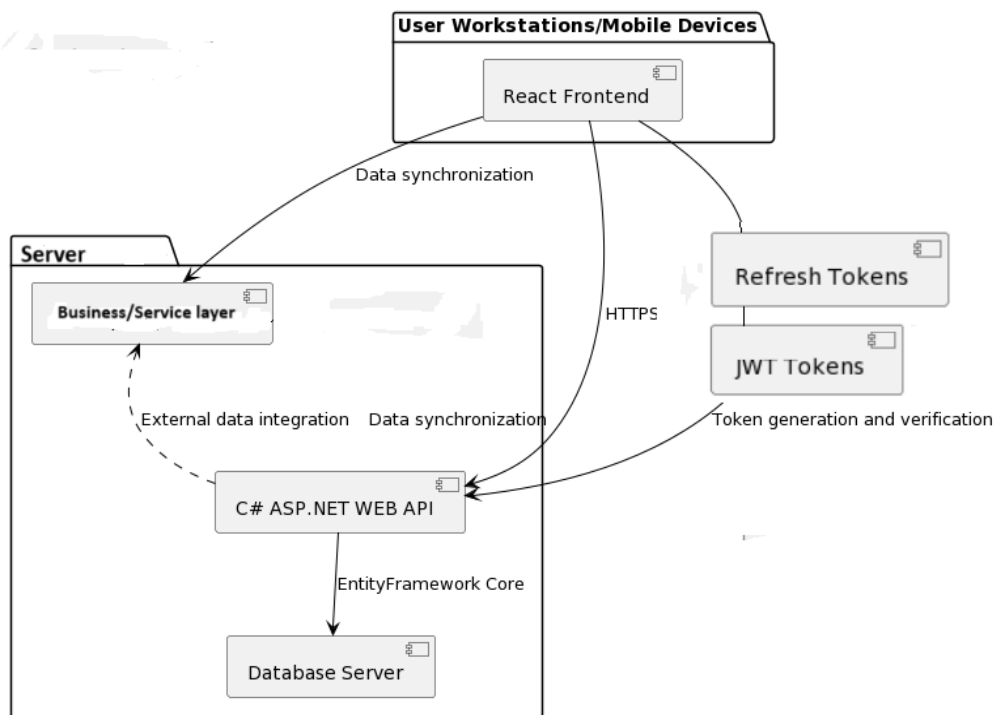
- ❖ Табло за управление на администраторите: Администраторите имат достъп до табло за управление на дейностите на потребителите и различни статистически данни.
- ❖ Управление на потребители: Потребителите могат да се управляват чрез панел за управление: Администраторите могат да управляват ролите на потребителите, включително повишаване, спиране или изтриване за нарушаване на правилата.

- ❖ Управление на проблеми: Администраторите имат право да управляват (одобряват/отхвърлят/редактират) всички подадени доклади за проблеми.

Тази диаграма на дейностите представя движението от дейности за администраторите. Те започват с достъп до таблото за управление. След това, в зависимост от това дали имат права да управляват потребители или проблеми, те могат да преминат съответно към дейностите Управление на потребители или Управление на проблеми.



## Изглед на внедряването



### 1. Сървърна инфраструктура:

Backend сървъри, хостващи C# ASP.NET Web API с EntityFramework Core.

Сървъри за бази данни за съхраняване на потребителски данни, отчети за издаване и друга информация за приложението.

### 2. Работни станции и мобилни устройства на потребителите:

Работни станции и преносими компютри, използвани от потребителите на приложението за достъп до приложението CityVox.

Мобилни устройства (смартфони и планшети), използвани от потребителите за достъп до уеб-приложението CityVox.

Тези устройства използват уеб браузъри, за да взаимодействат с фронтенда на CityVox.

### 3. Местна мрежа и интернет комуникация:

Комуникацията между работните станции на потребителите, мобилните устройства и бекенд сървърите се осъществява по интернет.

Комуникацията между frontend и backend се осъществява чрез HTTPS протоколи, за да се осигури сигурно прехвърляне на данни.

### 4. Комуникация с бази данни:

C# ASP.NET Web API комуникира със сървърите на базата данни, като използва EntityFramework Core, който осигурява обектно-релационно съпоставяне (ORM) за взаимодействие с основната база данни.

API обработва заявки за удостоверяване на потребителя, доклади за издаване, данни от карти, публикации в социалните мрежи и други дейности, свързани с потребителя.

### 5. Прехвърляне и съхранение на данни:

Прехвърлянето на данни между frontend и backend се улеснява чрез API повиквания с помощта на axios.

Данните се съхраняват и извличат от базата данни с помощта на EntityFramework Core, която управлява взаимодействието с ентитетите на базата данни.

### 6. Мерки за сигурност:

Удостоверяването и оторизацията на потребителите се обработват с помощта на JWT токени, съхранявани в локално хранилище, и бисквитки, съдържащи само HTTPS RefreshTokens.

Протоколът HTTPS осигурява криптирана комуникация между фронтенда и бекенда, като предпазва чувствителните данни по време на предаването им.



В бекенда са приложени допълнителни мерки за сигурност, като например валидиране на данните, за да се предотврати неоторизиран достъп и нарушаване на сигурността на данните.

#### 7. Балансиране на натоварването и мащабируемост:

Могат да бъдат приложени техники за балансиране на натоварването, за да се разпредели входящият трафик на приложенията между няколко бекенд сървъра, като се гарантира оптимална производителност и наличност.

Архитектурата на бекенда на приложението е проектирана да се мащабира хоризонтално, което позволява на системата да се справя ефективно с увеличения трафик на потребителите и изискванията за обработка на данни.

## Производителност

Създаването и поддържането на уебсайт, на който потребителите могат да споделят своите проблеми, изисква задълбочен анализ на ресурсите и оптимизация на производителността. Изчисляването на дадените ресурси обаче е доста относително. То се определя от фактори като трафик на данни, брой посетители и брой на оплакванията. Въпреки това целта на сайта остава ясна. Бърз и лесен за използване интерфейс, който позволява използването му от голям брой потребители.

От какво се определя производителността:

- Основните ресурси за един сървър

Този уебсайт ще изисква мощни сървъри, тъй като надеждната и бърза доставка на съдържание до потребителите е от съществено значение. Необходимо е да се определят ресурсите като процесор, оперативна памет, съхранение и широколентова връзка, като се вземе предвид очаквания поток от потребители и големината на съхраняваните данни.

На този етап не могат да бъдат посочени ясни числа, но примерните изисквания са следните:

- Процесор: Допустимо използване на 4-6 ядрен процесор с тактова честота от поне 2.5 GHz.
- Оперативна Памет: Минимум 16 GB RAM за старт, с възможност за разширение до 32 GB в зависимост от трафика.

- Пространство за съхранение : 500 GB SSD за начало, за съхранение на постоянни данни и медийни файлове.

- Ресурси на базата данни

Поради обема на данните, свързани с докладването на проблеми и новините, базата данни трябва да бъде добре оптимизирана. Това включва проектиране на ефективни структури за данни, индексиране и оптимизация на заявките.

Уебсайта ще съдържа:

- Мощен SQL базиран систем - MySQL.
- Индексиране и оптимизация на заявките.
- Дисково пространство: 1 TB SSD за съхранение на данни с възможност за автоматично разширение.

- Network/Мрежови ресурси

За постигане на бързо и безпроблемно зареждане на съдържание, трябва да се гарантира бързо мрежово свързване. Този аспект е от особено значение при качване или изтегляне на медийни файлове. Мрежовите ресурси включват:

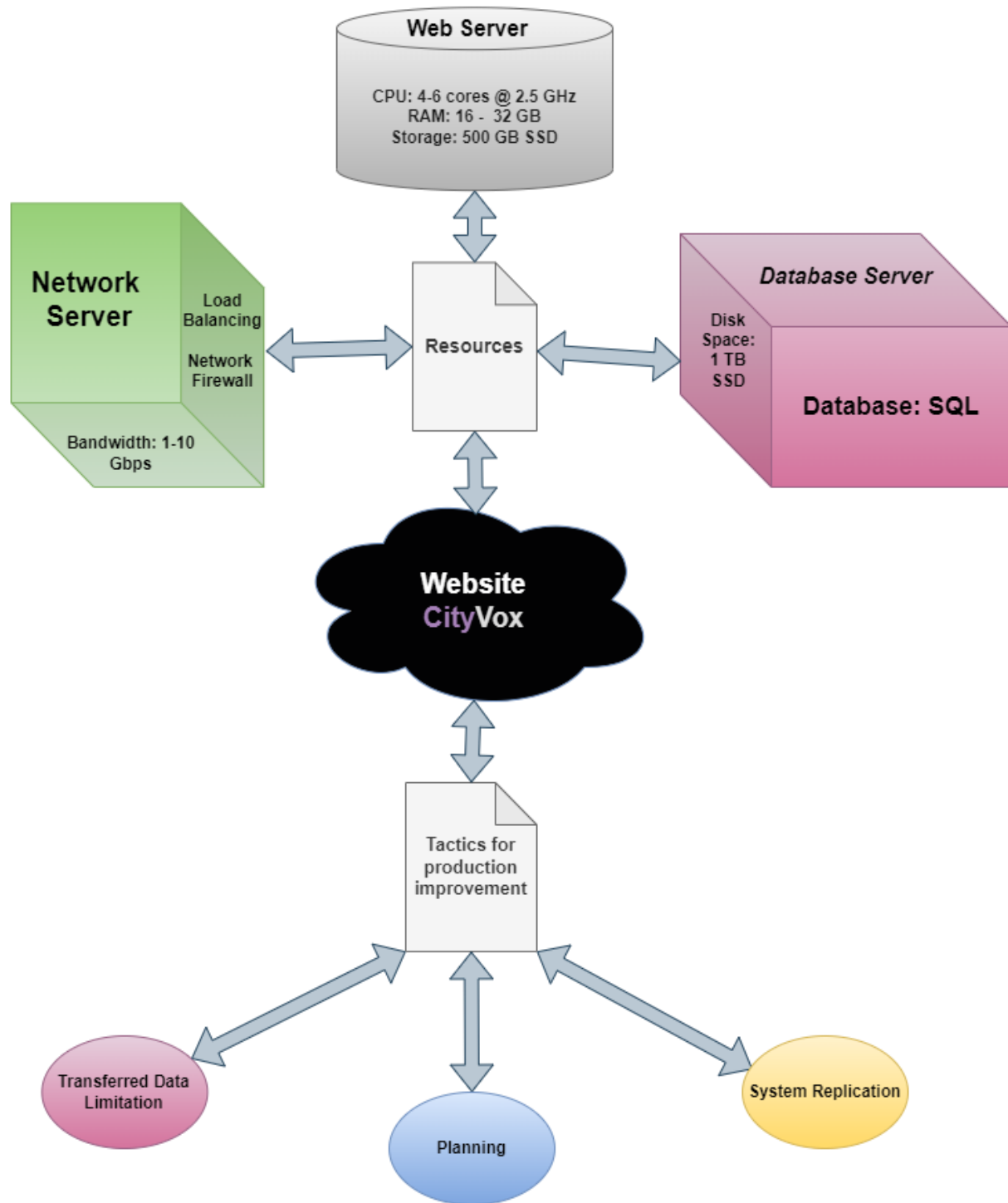
- Широколентова Връзка (Bandwidth): Минимум 1 Gbps, с възможност за скорост до 10 Gbps в зависимост от обема на трафика. Уебсайтът очаква голям обем на потребителско генерирано съдържание, включва мултимедийни елементи, затова може да бъде необходима значителна широколентова връзка, за да се гарантира гладко доставяне на съдържание.
- Балансиране на товара (Load Balancing) за оптимизиране на мрежовата производителност. Помага за разпределение на трафика към множество сървъри, за да се гарантира отзивчиво потребителско изживяване.
- Мрежова Защитна Стена (Network Firewall) е от съществено значение за защитата на данните на потребителите и предпазване от потенциални заплахи за сигурността. Тя помага за контролиране и филтриране на входния и изходния трафик, за да се защити информацията на потребителите и цялостната интегритет на платформата.
- Мрежова Доставка на Съдържание (CDN) Внедряването на CDN може да стане необходимо по мярка на растежа на сайта. То може да ускори доставката на съдържание и да подобри общото потребителско изживяване чрез намаляване на времето за зареждане.

Друг фактор определящ производителността на един уебсайт са тактиките за подобряването ѝ.

Това са:

- **Ограничаване на Товара:** Това включва определяне на максимални лимити за броя на заявките, които потребител може да направи за определен период. Например, ограничение на броя на заявките, които един потребител може да направи на секунда, като се определя максимален лимит, например 100 заявки на секунда.
- **Планиране:** Фоновите задачи, като одобрение на сигнали и публикации, се планират така, че да не засягат реалните потребителски взаимодействия. Системата трябва да бъде проектирана, така че поддръжка и резервни копия да се извършват по време на периоди с нисък трафик.  
Планиране на фоновите задачи и резервните копия да се изпълняват в часове с по-нисък трафик, например извън работните часове.
- **Репликация на Системата:** Репликацията на системата може да подобри производителността и устойчивостта. Чрез репликация на сървърите и балансиране на товара може да се справи с повишения поток от потребители и да се гарантира, че системата работи непрекъснато дори при евентуални инциденти.  
Репликация на сървърите и балансиране на товара, например използване на поне два или три сървъра със споделен трафик.

След като системата бъде пусната в работа ще се продължи наблюдението и оптимизацията. Ефективното следене и използване на инструменти за анализ на производителността ще бъдат важни за откриването на евентуални проблеми и оптимизиране на ресурсите.



## Използваемост

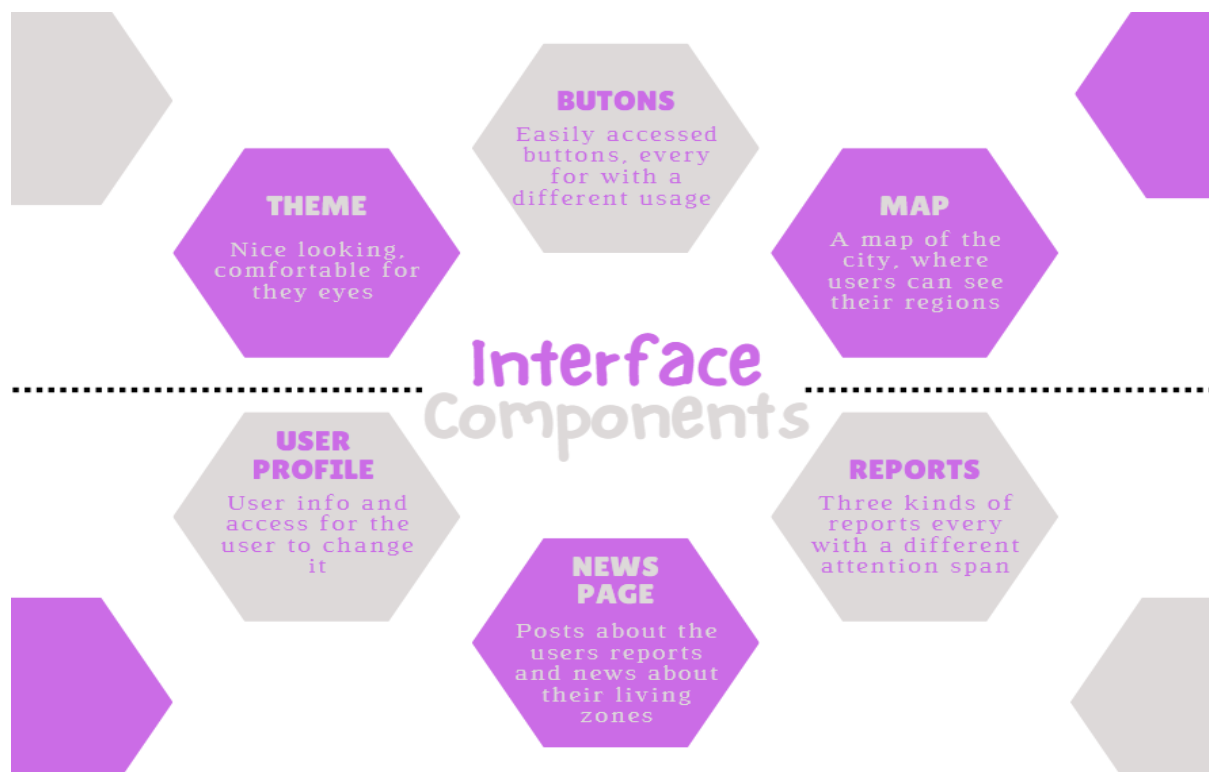
Анализът на използваемостта е критичен процес на оценка, който се фокусира върху ефективността, ефикасността и удовлетворението на потребителите, взаимодействащи с интерфейса на уебсайта. В контекста на нашия уебсайт, който се грижи за потребителите, които споделят своите проблеми, анализът на използваемостта играе решаваща роля за осигуряване на удобно за потребителя използване на интерфейса, както и удовлетворяващо прекарване в сайта.

Има няколко критерия за добър интерфейс, които нашият сайт трябва да покрие. Те са:

- Оценка на потребителския интерфейс:

За създаването на един достъпен, но в същото време, приятен за използване интерфейс, ще бъдат включени следните компоненти:

1. Обагрена тема, която прави сайта по приятен за гледане.
2. Достъпни бутони, които могат да отведат потребителите до желаното от тях място.
3. Карта на районите в съответния град, с помощта на която потребителя може да се запознае с проблемите в неговата зона.
4. Три вида докладвания, като всяко едно от тях е ясно изразено и има отделна страница, обособена специално за типа оплаквания.
5. Място за преглеждане и споделяне на докладвания, както и новини от председателите на даден район, споделяне на снимки отнасящи се за съответните проблеми, както и способността на за коментар под даден пост.
6. Място за потребителския профил, където потребителят може да види своите данни и евентуално да ги променя по свое желание.



- Навигация и Архитектура на Информацията

При представените компоненти, съдържащи се в нашият сайт, можем и да направим едно ясно представяне, как работи интерфейса на сайта и кои са участниците в него. Екипа за Frontend-а се състои от 3 човека, всеки отговарящ за различен компонент. Начинът, по който работи е следния. След регистрация, потребителя излиза на основната страница (main page), където има няколко бутона и една карта, взета от ... . Потребителя има достъп до три вида бутони, които го отправят съответно до страница за нормални проблеми, сериозни проблеми и проблеми обхващащи инфраструктурата. Там потребителят споделя за проблема си, в кой район е и я подава за одобрение от администратор. В частни случаи проблемът, по желание на потребителя, може да бъде докладван на упълномощените институции. На страницата за новините, човек може да сподели за проблема си и да разгледа други подобни или да се запознае с новини спрямо неговия район, споделени от представителите на жилищната зона.

- Достъпност и Инклузивност

Достъп до сайта ще имат всички, от деца до възрастни, без изключения. Това разбира се не означава, че хората ще могат да правят какво решат в сайта. Активността ще бъде строго следена и всякакви негативни коментари и неподходящи постове ще бъдат изтривани.

- Итеративно Подобрение

Съобразно резултатите от анализа на употребимостта и тестовете с потребители, ще бъде започнат итеративен процес на подобрение. Проблеми в употребимостта ще бъдат адресирани и ще последва поетапно подобрение на интерфейса, за да усъвършенстване цялостното потребителско изживяване.