

SafeStreets

Mandatory project 2019 – 2020

Software Engineering II

Maldini - Paone

Requirements

Goals

USER:

- G₁ Notify authorities about traffic violations
 - G₁₋₁ Send picture of violation
 - G₁₋₂ Send Position of the violation
- G₂ Authorities must be able to take an available assignment
- G₃ Allow authorities to report a finished assignment
- G₄ Allow all actors to visualize update statistics
- G₅ Allow the system manager to register Municipality to the service

SafeStreets:

- G6 Allow a Visitor to join the system registering him/herself to ensure reliability of the information provided by him/her
- G7 Store information about violations provided by users:
 - G7-1 Complete it with metadata
 - G7-2 Mine information
- G8 Identify potentially unsafe areas:
 - G8-1 Suggest possible interventions
- G9 Allow municipality to register Authorities to the service
- G10 Help the Municipality to make decision

Security Goals:

- S_1 Offer different levels of visibility to different type of users
- S_2 Personal data of users are stored respecting current security standards

Assumptions

- D₁) For each notification data and metadata provided by the system of the mobile phone are correct.
- D₂) Authorities always intervene in case of a notified violation
- D₃) GPS of authorities devices works correctly and gives the correct position every time.
- D₄) Authorities if available correctly informs the system about their availability.
- D₅) When citizen takes a photo the mobile application completes it with the correct metadata.
- D₆) System Manager, Municipality and authorities respects their duty of care
- D₇) When an authority is sent an email to register this will surely be received
- D₈) Information provided by authority are correct and no false report is ignored (always reported by authority as false).
- D₉) The Agent and municipality must be able to communicate
- D₁₀) Information of authority which is being registered are known by municipality
- D₁₁) The data of external database are always available

Requirements

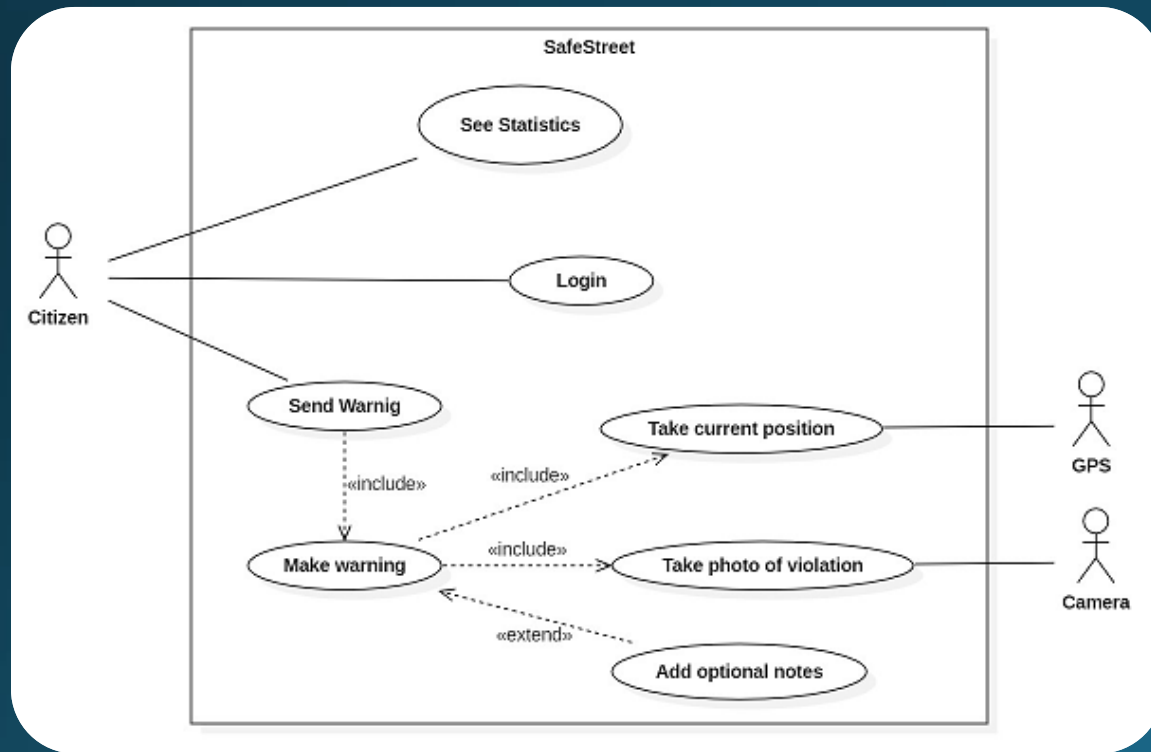
- R₁) Authorities' location must be known by the system when they are in service.
- R₂) When a Citizen makes a report the position is correctly added with the GPS when is available.
- R₃) The right authorities are notified about violations.
- R₄) Authority must be able to provide the system how the assignment finished: resolved and the type of violation, no intervention needed when arrived, false report.
- R₅) The system must make Statistics available when asked.
- R₆) Statistics are always updated when an event happens.
- R₇) For registering a Municipality his/her data must be provided to a System manager who will add those data to the service to sign up him/her.
- R₈) A visitor must be able to begin sign up process in the SafeStreets App filling a form with his data.

Requirements

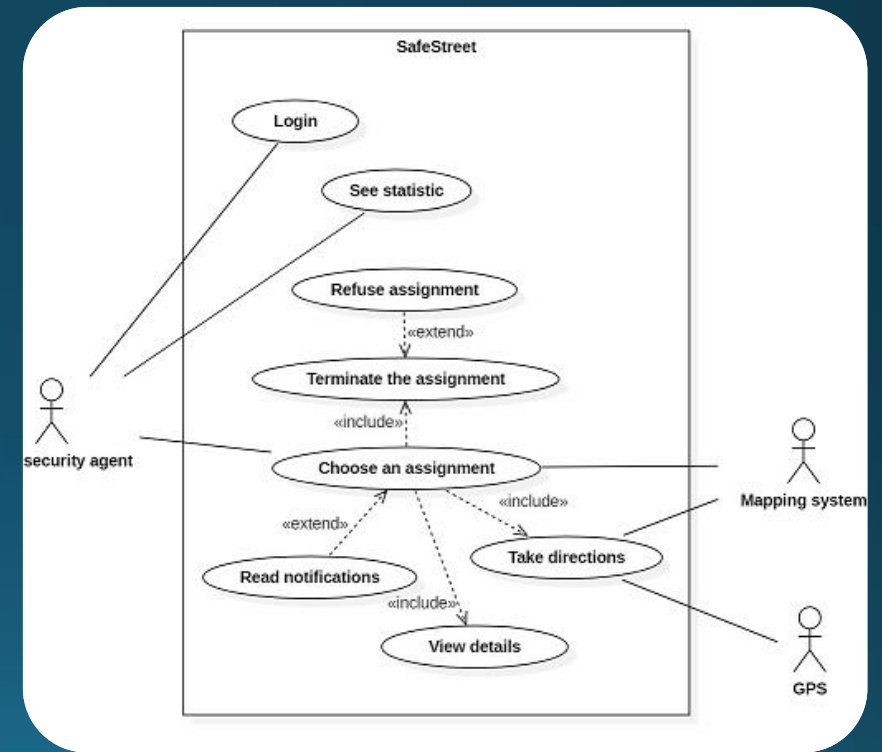
- R₉) When the creation of an account is successful the system must notify the Visitor sending an email to the address provided in the sign up process.
- R₁₀) When GPS is not available the user can input the position from a map.
- R₁₁) Users to use the full service must be able to login providing the right credentials.
- R₁₂) The camera of the mobile phone must be accessible to take photos of violations.
- R₁₃) Suggestions must be available when municipalities request them.
- R₁₄) The User must be able to select the licence plate between the ones in output from the Licence Plate Recognition algorithm.
- R₁₅) Each Username is unique

Main Use Case

Citizen



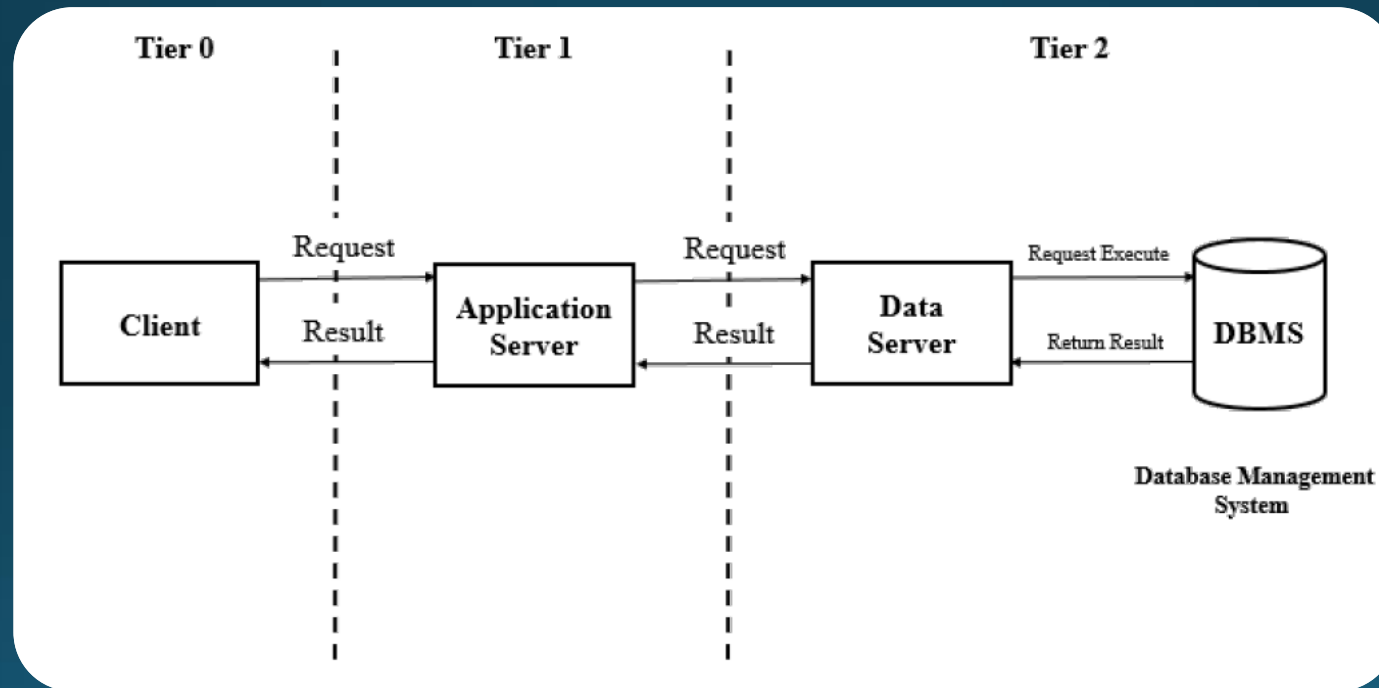
Authority



Design

Architecture

We've chosen the three-tier architecture:

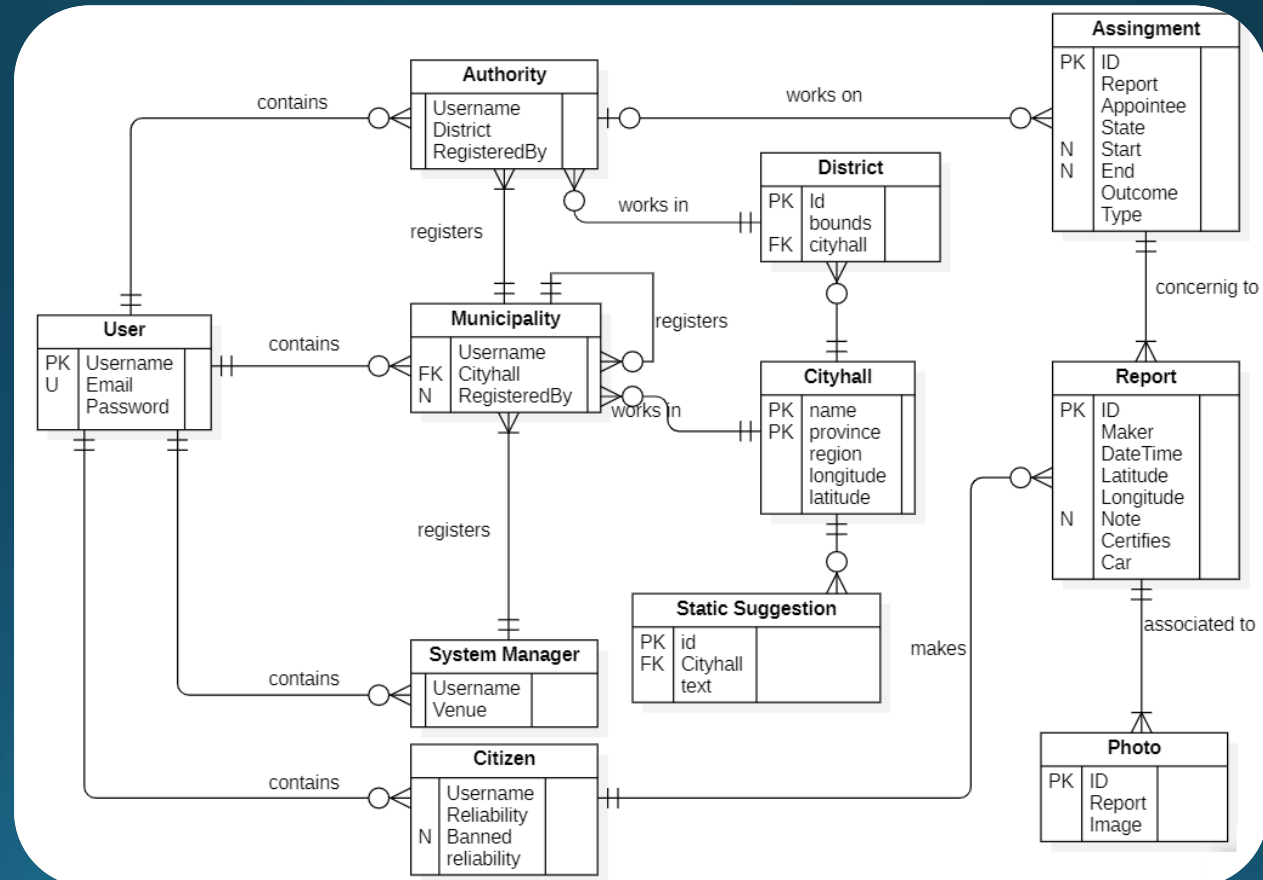


Why three-tier architecture ?

- Lightness
- Flexibility
- Scalability
- Security
- Maintainability
- Reusability

DATABASE STRUCTURE : ER diagram

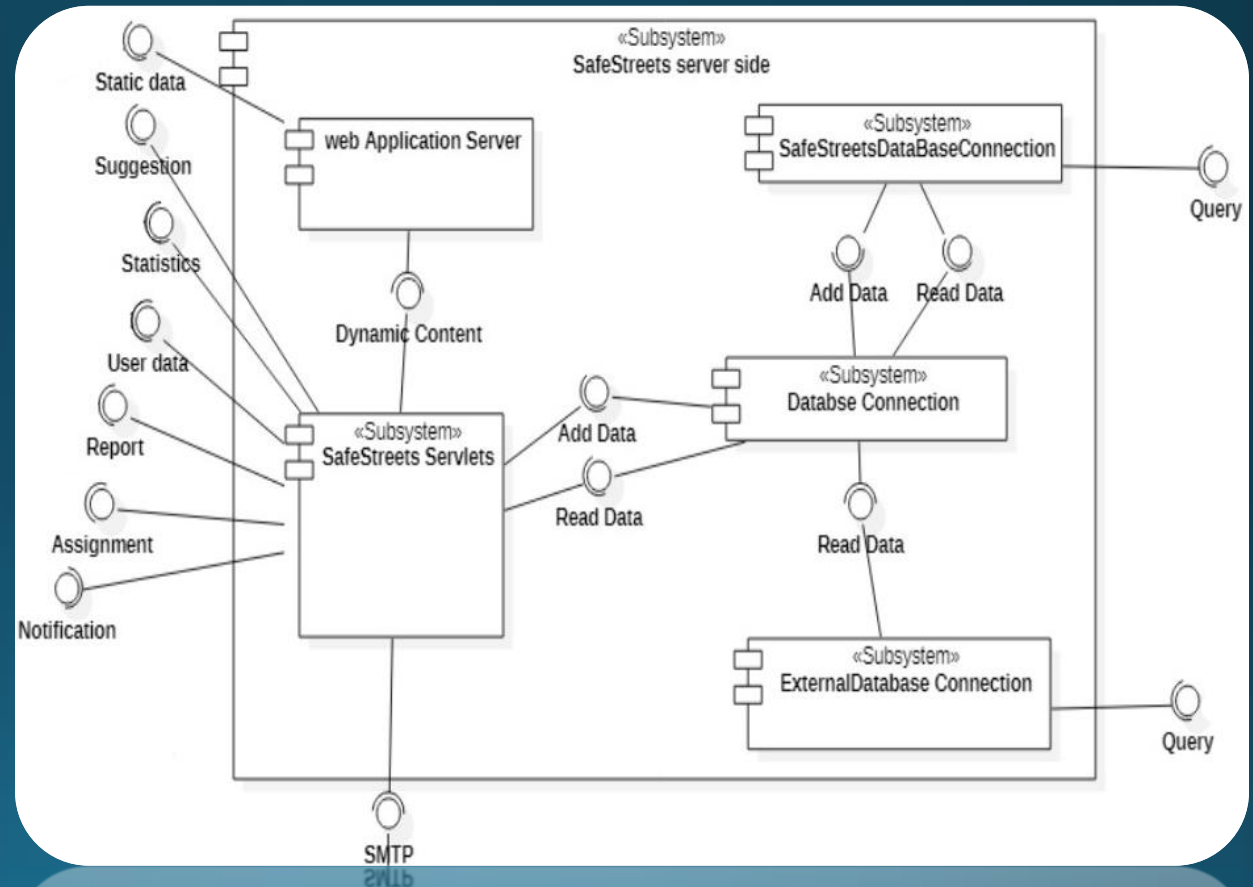
- User
- System Manager
- Municipality
- Authority
- Citizen
- District
- Cityhall
- Statistic Suggestion
- Report
- Photo
- Assignment



SERVER SIDE: General component view

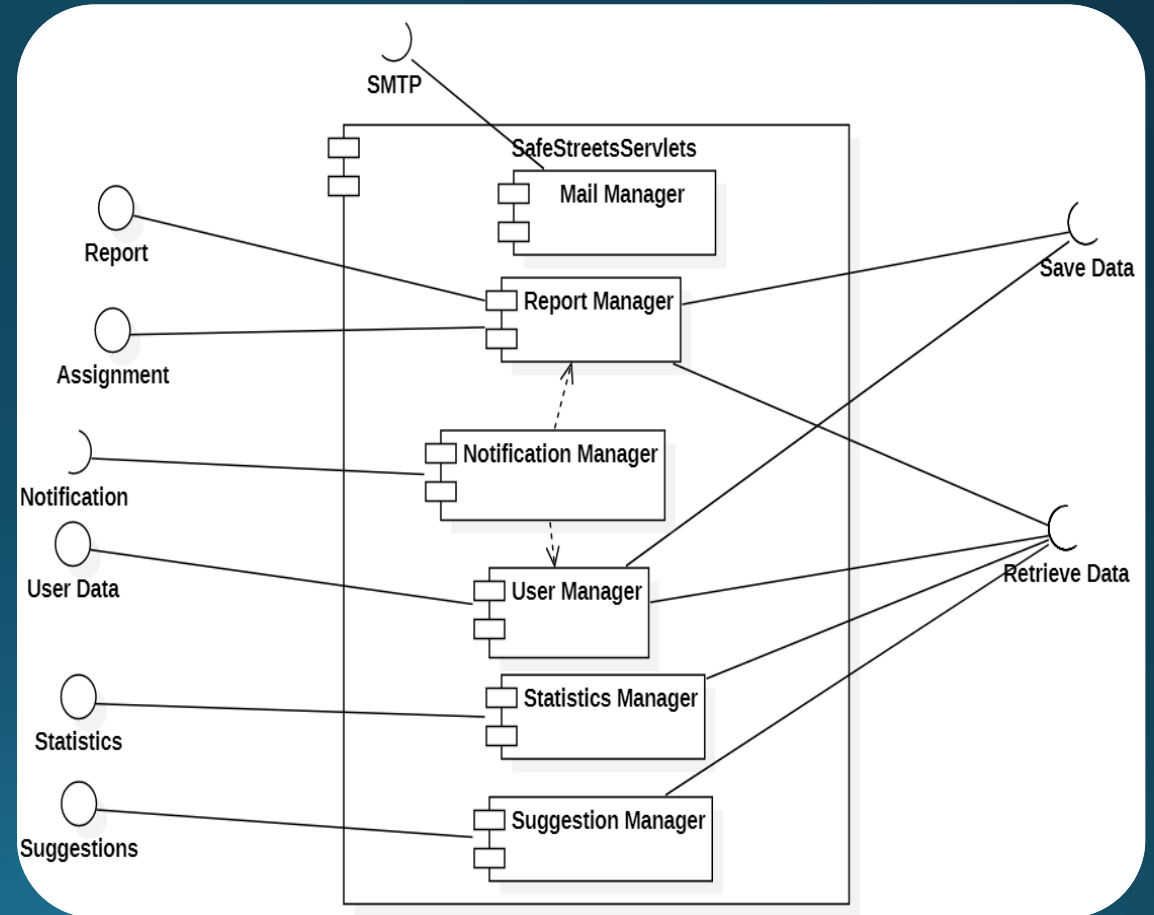
We have divided the system in various part:

- Safestreets Servlet
- Web Application Server
- Database Connection
- SafestreetsDatabaseConnection
- ExternalDatabase Connection



SafeStreets Servlets

- Mail Manager
- User Manager
- Notification Manager
- Statistics Manager
- Suggestion Manager



Why multiple Servlet server?

- Different access points
- Expansion
- Parallel testing
- Maintainability
- Scalability

Web Application Server

- Provides to Web Application the static data (HTML,CSS,JS)
- We have introduced this component because it is essential to divide the static data from dynamic data.

Database Connection

- We have used the pattern façade for the realization of this component because it hides the complexities of the larger system and provides a simpler interface.
- It provides the functionalities to access data to Servlets, communicating with Connection Subsystem.

SafestreetsDatabaseConnection

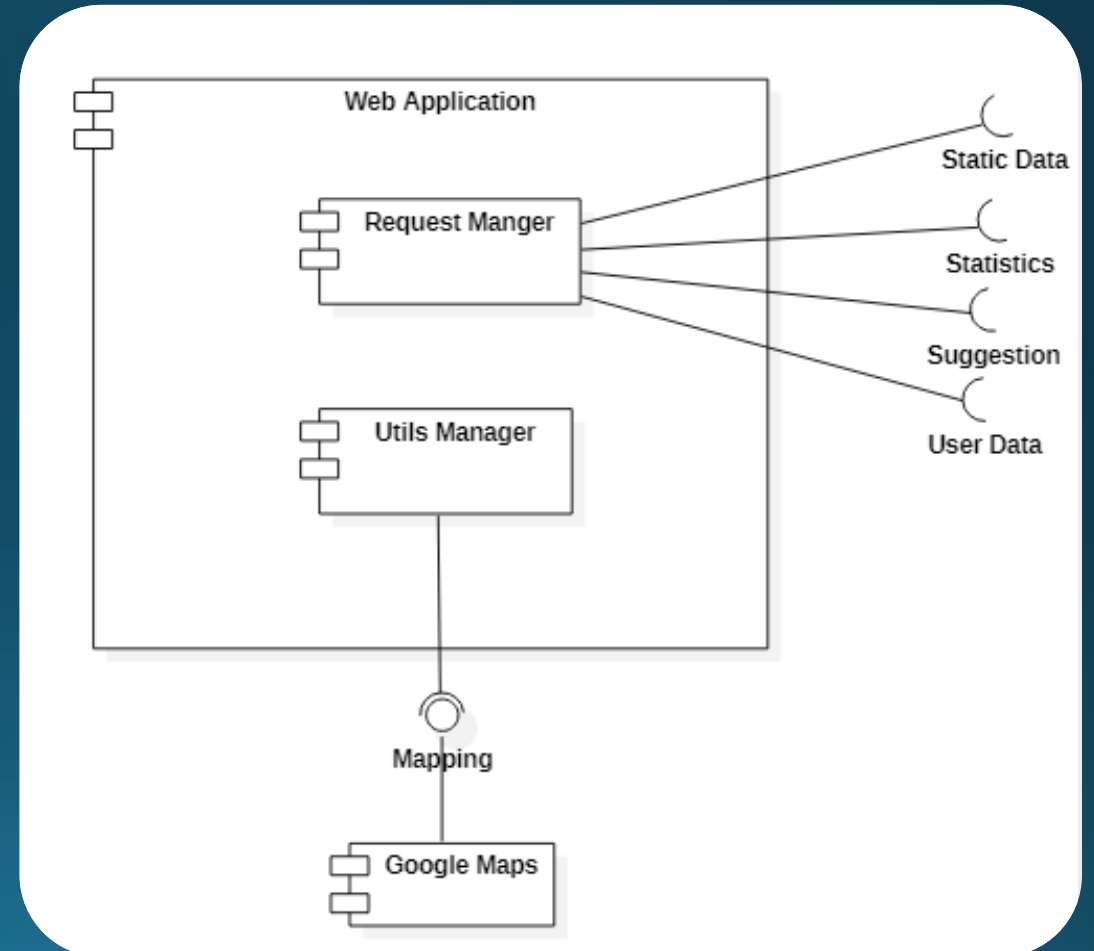
- It communicates with SafeStreets DataBase
- It executes query (select / insert / update)

ExternalDatabaseConnection

- It communicates with external databases
- It executes query (only read)

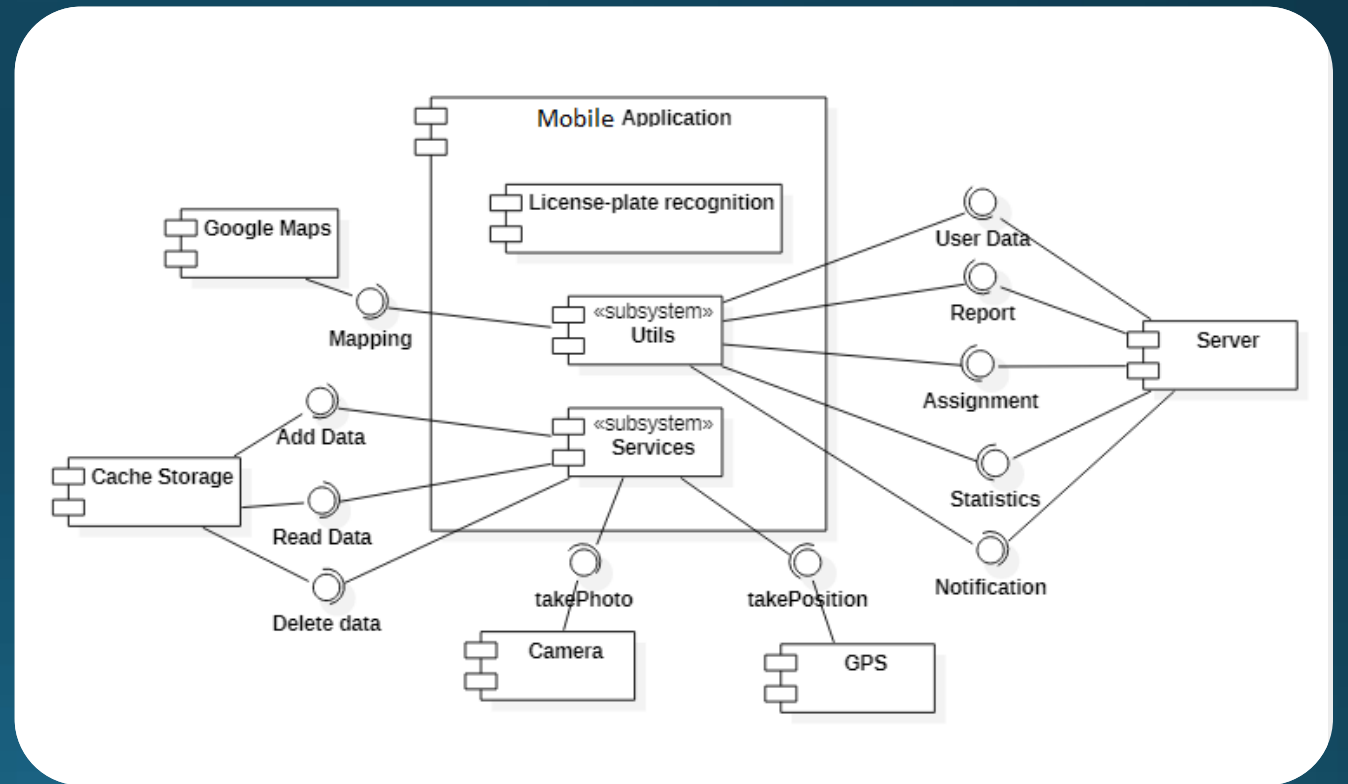
CLIENT SIDE: Web Application

- Request Manager
- Utils Manager



CLIENT SIDE: Mobile Application

- Services
- Utils



Implementation & Testing

DEVELOPMENT CHOISES: Back end

- Eclipse IDE for Java EE Developers
- Apache Tomcat
- MySQL Workbench



Why java?

- Cross-platform
- Variety of functionalities
- Object oriented
- Annotations
- Garbage collector

DEVELOPMENT CHOISES: Front end

- Software

- Eclipse IDE

- API

- openStreetMaps (leaflet)
 - OpenALPR (OCR)



Why web application?

- Advantage of Javascript with jquery
- Chaching mechanism
- Leaflet efficiency