deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

Igor Coelho [113532], João Capucho [113713], Zakhar Kruptsala [114478]
v2025-05-07

1 Introduction **1**
1.1 Overview of the project 1
1.2 Known limitations 1
1.3 References and resources 2

2 Product concept and requirements **2**
2.1 Vision statement 2
2.2 Personas and scenarios 2
2.3 Project epics and priorities 2

3 Domain model **3**
4 Architecture notebook **3**
4.1 Key requirements and constrains 3
4.2 Architecture view 3
4.3 Deployment view 3

5 API for developers **3**

1 Introduction

1.1 Overview of the project

<contextualize the objectives of this project assignment in the scope of the TQS course>
<introduce your application/product: brief overview of the solution concept. What is it good for? Introduce the name of the product if it has one>

1.2 Known limitations

<explain the known limitations, especially the features that were planned/expected but not implemented (and why…)

To be reviewed and completed by the end of the project >

1.3 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>

2 Product concept and requirements

2.1 Vision statement

Our product, NikoGrid, envisions a future where charging of electric vehicles (EV) is globally accessible, trustworthy and reliable and is seamlessly integrated with various networks. Our main goal is to eliminate the existing fragmentation in the EV charging ecosystem, offering a unified digital system that empowers drivers, station operators, and third-party services to interact effortlessly and efficiently.

NikoGrid addresses the critical challenge of fragmented EV charging services, offering a solution that provides the following key features:

- Driver services - Allow EV drivers to search for EV charging stations, book a slot, plan trips with stations in mind, unlock and use stations, effortlessly make payments, and monitor personal electric charging consumption.
- Back-office services - Allow operators to register charging stations, change their availability, monitor consumption and configure prices.

This product differentiates itself from other platforms by prioritizing interoperability and user-centric design. Unlike other platforms that may lock users into a specific network, our goal is to make a plethora of different providers easily accessible for our users.

By addressing the core challenges of EV charging and focusing on the interoperability, and user experience, NikoGrid has as one of its primary goals to accelerate the adoption of EVs and contribute to a sustainable transportation future.

2.2 Personas and scenarios

In our system we forecast many different users, each with its own background and objectives, as such our platform needs to adapt to all these differing realities in order to satisfy the requirements imposed by all the persons that will use it. To do so, we model some personas that roughly approximate the main groups that will use our application.

Our first persona is George, a 42 years old man that works in a large office building in the middle of a bustling city. He has a wife and 2 small children that he needs to take to school every work day. He lives in the vicinity of the city, where public transportation is not viable and the distance makes walking or cycling a non option. George is an environmentally conscious person, so given his constraints, he decided to buy an EV vehicle in order to minimize his impact on the environment. One of the problems he faces every day is finding a charging spot near his workplace that is available. He values family time on top of everything so finding a solution to this problem that would allow him to charge closer to his job without wasting time searching in the city for a spot, allowing him to

clock in and out earlier, is important to him, but he doesn't want to waste time checking each station operator's platform for online booking.

Our Second persona is Maria, a 29 years old woman owner of some small electrical charging locations. She studied engineering in college and after graduating decided to invest in the EV market by creating a couple of vehicle charging locations. She now owns and operates a dozen successful locations and is looking to expand even more. One of the main problems she has identified when starting new locations is the lack of discoverability, since most popular charging spot sites are based on crowdsourcing. She can manually submit to add for each of these platforms, but this is a process that consumes a lot of time, and most sites require verification or multiple submissions before showing the location. She also is looking to add support for online booking and charging to her locations, however she isn't ready to invest in a platform that would only apply to her business. The company is important to her as it's her creation, however she also likes to go hiking and hanging out with friends, so reducing her workload without compromising the business is important to her.

Our third and final persona is Mark, a 37 years old man working for Maria, doing maintenance of the charging locations. As part of this work, Mark sometimes needs to close down stations, or even the whole location, for scheduled and unscheduled maintenance. Right now, he does this by affixing a paper to the station explaining the situation, but this causes drivers to still drive to the station as they don't know until they are there. This also leads to sometimes forgetting to remove the maintenance paper, and perfectly functional stations being unused. Mark is looking to be promoted in order to finance his hobby of travelling, however a significant portion of his maintenance job is marking the station as unavailable and answering customers' questions on the status of the stations. As such he is very receptive to the idea of a digital system that would allow him to mark a station as out of service with a few clicks, and even schedule it, and at the same time would inform customers of the situation, this would reduce the time it takes to perform maintenance, allowing him to perform more in a day, increasing the chances of being promoted.

Scenarios:

1. **George needs to park his car at work** - When leaving his home, George opens the NikoGrid companion app to search for parking spots with charging capability nearing his workplace at the time of arrival. After finding a parking spot that is available for the needed timeframe (his working hours), he books the spot to ensure it is available when he arrives to work. After his daily commute, George parks his EV in the booked slot and unlocks the charger to ensure his car is charging.

2. **George consults the statistics of his booked slot** - After lunch, George is interested to know how much energy he has consumed this day, and how much charging his EV has cost him. In addition, he also wants to compare today with the past week's average on energy consumption. Lastly, he wants to see if energy is still being consumed, to know whether his car is still charging or if it is already fully charged.

3. **George leaves his job and commutes back home** - After clocking out at work, George gets to his car and wants to leave. To do that, he first has to pay the bill of the charger, followed by unlocking the charger and marking the spot as unoccupied. He is then free to take his car and commute back home.

4. **Maria wants to add a new station** - Following the construction of a new charging station, Maria wants to make it available in the NikoGrid platform to ensure it is used by customers in need of parking spots with charging capabilities. She considers this new parking lot to be on a premium location, and therefore wants to set the prices accordingly.

5. **Mark sets up a periodic inspection in a parking lot** - It has been almost a year since Mark has last inspected a parking lot's chargers, and he wants to make sure that the equipment is running smoothly. Therefore, using the NikoGrid platform, he schedules a maintenance period on the parking lot a week from now, ensuring that for a day no vehicles are going to be using the chargers.

6. **Mark has to perform emergency maintenance on a charger** - Yesterday a customer reported that a given charger had presented problems when charging the vehicle, and reported it to Mark. Since he wants the charger to have the least amount of downtime possible, he wants to fix the charger as fast as possible, and therefore, he sets the charger to maintenance mode, meaning that no vehicle is able to use the charging capabilities of the parking space.

2.3 Project epics and priorities

The epics identified in the project are the following, ordered by their priority from highest to lowest:

2.3.1 Station Discovery

This epic's main goals are the features that enable users to locate and filter charging stations by their status, availability, proximity and other criteria.

1. **Browse nearby charging stations** - As a user, I want to browse nearby stations, so that I can make a more informed decision on my next car purchase.

2. **Filter stations** - As a user, I want to filter for active and available stations, so that I can book a slot and charge my electric vehicles.

3. **Find the closest available charging station** - As a user, I want to find the closest available charging location to my or another location, so that I can head there to charge.

2.3.2 User Profile Management

To support some of the following epics, a user management system is necessary to associate operations and records to distinct accounts:

1. **Account register** - As a user, I want to register an account, so that I can use it to store information and make operations that require an account.

2. **Account login** - As a user, I want to be able to log in into my account, so that I can access it from a different device.

2.3.3 Slot Booking and Scheduling

In this epic we will focus on implementing features that would allow users to book a slot, make use of that reservation or cancel it. This epic will contain the following stories:

1. **Slot reservation** - As a registered user, I want to reserve a slot in a charging station, so that I can use it later.
2. **List reservation** - As a registered user, I want to check all reservations that I've made, so that I can verify if I already have a reservation.
3. **Cancel reservation** - As a registered user, I want to cancel a reservation I no longer need, so that I can free the charging spot.

2.3.4 Charging Session Management

The management of the charging session is important, especially for the user to be able to use the chargers, since it will be required to unlock the charger for usage, and then lock it after usage. It will also allow the user to view information about the current charging session. Therefore, this epic will consist of the following stories:

1. **Unlocking chargers** - As a registered user, I want to unlock the charger with my reservation, so that I am able to plug it into my car.
2. **Consumption accounting** - As a registered user, I want to be able to see the cost of my current charging session, to make sure I know how much I have to pay.
3. **Energy and duration stats** - As a registered user, I want to see how long I have been charging my car, and how much energy I have consumed.
4. **Charging prediction** - As a user, I want to predict how much it's going to cost me to fully charge my car's battery and how long it will take, so that I can plan around it.

2.3.5 Advanced User Profile Management

Each user has a profile that contains personal information and statistics about the usage of the platform. It is imperative that the user is able to update his information, and that he can visualize the statistics that have been gathered on his profile. Therefore, this epic comprises the following stories:

1. **Profile editing** - As a registered user, I want to be able to edit the personal information associated with my profile, to ensure that info can be kept up to date.
2. **Car association** - As a registered user, I want to associate an EV car to my profile, so that I can quickly select it when predicting charging costs and time.
3. **Statistics visualization** - As a registered user, I want to be able to visualize the statistics that have been gathered on my profile, such as total cost or charging time, so I can analyze my usage patterns of the platform.
4. **Charging history** - As a registered user, I want to be able to see my charging history, to be able to detect patterns in my EV charging habits.

2.3.6 Station Management

Focusing on station operators, this epic will bring features that allow them to register and unregister charging stations. Additionally, station operators may need to update the status of each charger to, for example, alert drivers about maintenance periods. It is also important for operators to be able to monitor usage statistics of their stations to make decisions. Lastly, they may want to update prices to issue discounts and other offers. In this vein, this epic will be made of the following user stories:

1. **Apply for station operator** - As a registered user, I want to apply to be a station operator, so that I can manage my charging locations.
2. **Verify station operator** - As an admin, I want to accept or reject a station operator request, so that I can welcome station operators to the platform.
3. **Register a station** - As a station operator, I want to register a station, so that I can reach more people and have an easier way to manage my stations.
4. **Unregister a station** - As a station operator, I want to be able to unregister an existing station, so that the users can know that the station is no longer operational.
5. **Manage collaborators** - As a station operator, I want to add and remove registered users as a collaborator, so that I can delegate work to them.
6. **Update a station's status** - As a collaborator, I want to be able to update a station's status, so that I can fix a recent issue that the charger has.
7. **Monitor statistics** - As a collaborator, I want to monitor the statistics of a given charger, so that I can detect issues early and analyze usage for a better monetization plan.
8. **Issue discounts** - As a station operator, I want to issue discounts, so that I can improve the overall sales of the charger.

2.3.7 Payment Integration

Payment is a crucial part of this platform. The main goal is to offer a unified platform where our users are able to perform all the operations needed, without having to interact with different, provider dependent platforms, that change depending on the charger provider in use. The most crucial part is the payment. This allows us to offer a secure method for our customers to pay the usage, while offering the convenience of doing it all in one place. This epic contains a single story

1. **Session payment** - As a registered user, I want to pay for my usage of the charger to ensure that I can use the services that are being provided.

3 Domain model

<which information concepts will be managed in this domain? How are they related?>
<use a logical model (UML classes) to explain the concepts of the domain and their attributes, not a entity-relationship relational database model>

4 Architecture notebook

4.1 Key requirements and constraints

The main requirements that affect the architecture choices that will be made are the following:

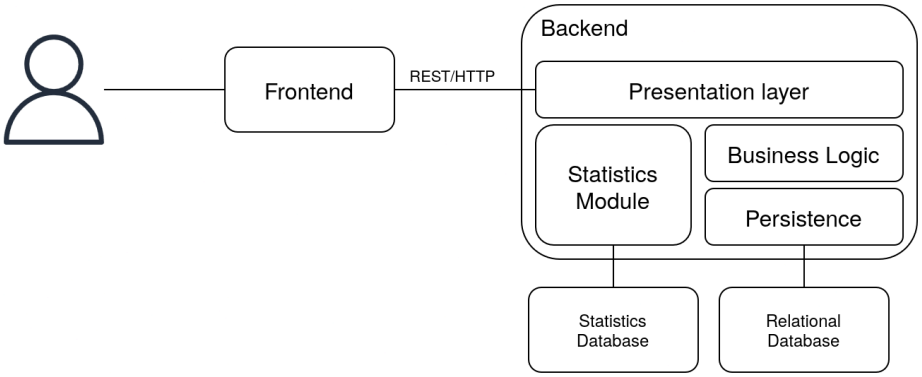
- Support for using the application both on desktop, which has a large landscape screen and good internet connectivity, and mobile, which has a small vertical screen with not ideal connectivity.
- Integration with an external payment processor, the application will utilize this service for all payment related operations, however in case the payment processor is offline the application should continue to process other requests that don't require payment services.
- Integration with an external mapping service, the application needs to display a map to the user as such a mapping service is needed, the application is centered around this map view so most of the application would stop working, exceptions are the payments, existing bookings, and statistics, these must continue to work.
- Support for statistics data that can arrive in a larger volume than other requests and needs to be queried and transformed efficiently to respond to user requests.
- Cost, the architecture needs to handle a number of users in the tens to the hundreds in a small machine as it's the only thing we have available.

Besides these previous requirements there are some extra constraints we need to be aware of:

- Performance, the application at this stage is expected to consistently handle for long periods of time 10 concurrent users with a p95 lower than 500ms in all requests, except payment requests.
- Elasticity, the application is expected to comply with the performance requirements outlined in the previous point with 5 times the number of concurrent users for 2 minutes.
- Scalability, despite the cost requirement, the architecture should be versatile enough such that with better funding (for example) after a successful launch it should be possible to serve thousands of users with more powerful machines.
- Robustness, all data, except for statistics, should be durable even in case of sudden failures (hardware failure or power for example). This is from the point of view of the user, only when a user receives a response to the request is considered durable, if the user makes a request but never receives a successful response the durability of the data is not considered.
- Upgradability, it should be possible for most updates to be done transparently without disrupting user activity.

4.2 Architecture view

An overview of the architecture is given in the following diagram:



The architecture follows a standard three tier architecture of frontend, backend, and database. However instead of one database, two will be used, one for regular data and another for statistics data. This will be done because the access patterns of both differ too much to allow only one database to be used efficiently.
The backend will mostly follow a standard layered architecture style, maybe with some modular monolith architecture style in the logic responsible for handling statistical data.
Integration with the mapping services will be done in the frontend, while the integration with the payment services will be done in the backend, more specifically the business logic layer.
The use of a monolith helps with reducing the overall cost of running the project, but still allows some scalability by adding some more replicas on faster hardware. Overall the architecture follows the paradigm of a sacrificial architecture, it follows current requirements and constraints but it's probably not going to be able to handle increasing scalability concerns forever.

4.3 Deployment view

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Anotar, no diagrama, as tecnologias de implementação, e.g.: colocar o simbolo do PostgreSQL na Base de dados,…]. Indicar a existência de containers (Docker), endereços IP e portos,...
Esta parte será completada quando houver efetivamente deployments

5 API for developers

[Explicar genericamente a organização da API e coleções principaus. Os detalhes/documentação dos métodos devem ficar numa solução *hosted* de documentação de APIs, como o Swagger, Postman documentation, ou incluída no próprio desenvolvimento (e.g.: maven site)
 Be sure to use best practices for REST Api design. Keep minda REST API applies a resource-oriented design (APIs should be designed around resources, which are the key entities your application exposes, not actions)