BUSRUSH

Technical Report - **Product specification**

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, 3rd of January 2023
Students:	98384: Inês Castro Baptista 103154: João Manuel Pedrosa Fonseca 103183: Diogo Luís de Albuquerque Paiva 103530: André Gabriel Butuc
Project abstract:	BusRush is web and mobile application that allows users to monitor bus metrics such as delays in real-time. It hopes to solve the existing transparency issue attached to public bus service, where commuters aren't aware of the current state of the bus they want to take. In a more administrative perspective, it can provide tools for control and analysis of a bus fleet, helping on the decision making process. The system revolves around the Spring Boot framework, which provides a bridge between the frontend and backend components.

1 Introduction

BusRush is web and mobile application that allows users to monitor bus statistics in real-time. This data – location, speed, fuel consumption, number of passengers - is collected using a device onboard of every bus, that periodically sends relevant information about its current state to the system, to be processed, stored or shown to end users. Using this real-time data in combination with static information such as bus schedules, the system can, for instance, provide an estimation for the time of arrival at any given stop - or the user's closest stop, based on his current location.

2 Product concept

Vision statement

This system combines information collected from buses in service real-time with static data, allowing for a more detailed monitoring over a fleet of buses. There will be two types of users: administrators and commuters.

Administrators can access information provided by the system using the web app. They have control over all aspects of the bus fleet – assign buses, drivers, monitoring devices, routes, stops, schedules – and all the collected information – real-time statistics and historic data for any given bus or route. This provides information to the company for evaluation of their drivers performance, upcoming bus maintenances and quality of their services to customers.

Commuters can access information provided by the system using the mobile app. They can view bus schedules for any given stop – or the closest one to their location – check the expected time of arrival of buses in real-time and define specific routes to be notified of delays. This allows the commuters to better manage their routine and time, by providing a mean to check the incoming buses and their expected delays.

In public transportation, there is a similar service offered by train stations, where there are boards with the schedules of departure and arrival for every train, including their delay. However, the delay information is not available online to be easily accessed. This can be a problem, especially for commuters who depend on other means of transport to get to their destination. They usually plan their schedule in advance, and a slight delay on a transport can have a great impact on their trip, making them spend more time waiting for the next transport.

In the bus context, this product is something new. Currently, there isn't any real-time information available to commuters in the stops – only the schedule, that can be often missing. Normally, users need to be at the stop earlier that the marked schedule, so that they don't miss their bus. However, it is common for a bus to be delayed – unlike trains, they share the road with other users. Why should people have to blindly wait for a bus to come, one that can compromise their entire commute? Why can't they have access to the real-time arrival time of their bus, so that they can plan their commute ahead, to not be constantly looking for the bus arriving, or to do something productive in another place knowing they won't miss their bus?

Personas and motivations

Rui (40 years old)

Rui is a forty-year-old Transportation Manager for the AveiroBus network. Having obtained his Management degree at ISCTE – University Institute of Lisbon in 2010, Rui is a very methodical, organized and responsible professional with peculiar hobbies such as soap carving and tree shaping.

During his professional experience at AveiroBus, Rui has seen with his own eyes consistent bus delays and spontaneous route changes by bus operators which led to multiple unsatisfied reviews by commuters. Besides this problem, the bus operator's negligence might lead to the breaching of their contract's clauses which in return may cause high-administrative problems.

Facing the dilemma of keep working with the old system or adapting to a better one, Rui is now open to look for existing solutions that allow real-time monitoring of bus operations.

Motivation: Rui would like to have a real-time monitoring system of AveiroBus's bus fleet which would provide him with statistical and meaningful information that will allow him to make administrative changes and recommendations.

João (20 years old)

João is a twenty-year old software engineering student at the University of Aveiro and he is currently in his third year of the degree. Besides studying, João's hobbies range from playing videogames to playing football on Sunday mornings with his friends.

Despite studying in the city of Aveiro, João is staying at his parents' house in Pardilhó, which is a village in Estarreja's Municipality, which forces him to daily commute between Aveiro and Pardilhó. Therefore, travelling by public transportation, such as train and bus, is normal for him.

Due to his academic schedule, João felt the need to build a tight daily routine of picking up certain buses and trains. However, this tight schedule only works if the bus and train is always on time, which, unfortunately, doesn't always happen.

Constant bus delays and train delays have made João's routine of buses and train commute fail, because one bus delay would mean losing the next bus and so on. This consequently has made João on many occasions spend a lot of time waiting at the bus stop wondering if the bus has already gone by or if it is just simply delayed since there isn't any platform or screen at the bus stop which gives insight regarding the state of the bus.

João's daily struggle of needing to guess and rely on external factors has made him feel frustrated for wasting time which he could use to study for his courses or work on a certain project.

Motivation: João would like to have something, such as an app, that would give him information in real-time regarding the bus that he wants to catch. This crucial piece of information would allow him to adapt his original schedule and make him optimize every free time he has, therefore contributing to his academic success and daily fulfilment.

Main scenarios

- **Rui (40 years old)**

    - ### Rui monitors bus operations in the map

      To monitor bus operations in real-time, Rui accesses the live map, only accessible by AveiroBus administrators and managers. In the embedded map is the area covered by the AveiroBus fleet buses. He sees the movement of the buses in real time. Due to the number of spontaneous route changes, Rui is particularly interested in verifying if the bus operators make a stop at all the planned stops. He visualizes the bus movements and if they are respecting the planned routes. He notices that a bus operator did not stop at the stop in Aveiro Hospital. He taps on the bus to achieve a modal with the driver's information and now he can report the event.

    - ### Rui checks bus real-time metrics in the map

      Due to recent complaints about the lack of buses to cover a specific area and overcrowding of buses, Rui started checking the real-time metrics of the buses for future administrative decisions. He accesses the dashboard of the web application. He sees the buses moving. He taps on a bus to see real-time metrics like velocity, occupation and others from that bus. These metrics are sent in real-time by the attached device and camera of each bus and then is processed by the business logic side of the application. Rui notices that a bus has reached its maximum capacity of 90 people. He can take this information to his colleagues for analysis and decision-making.

- **Rui checks statistics relative to bus fleet performance**

The embedded map has been useful to check bus fleet performance and provide valuable data for AveiroBus administrators and managers. Now Rui wants to check the impact of these decisions over time on the bus fleet performance and productivity. In the web application, Rui taps the statistics tab. In the statistics tab, Rui can see graphs relative to bus delays and bus occupation in the last week. He is happy to find that over the last week of using the app, there was a great decrease of bus delays and spontaneous route changes by the bus operators.

- **Rui adds a bus and associated device to the fleet topology**

As the company AveiroBus grows, there will be an increasing need for convenience in administrative operations. Rui wants to add a bus to the bus fleet after recognizing the need for more buses for a specific route or a new route. In the fleet interface, he clicks on the add a bus button and device button and links the two. He then specifies the driver and routes of the bus.

User Stories

- **Monitor bus routes in the embedded map (4 points)**
- **Tap on bus in embedded map to get real time metrics (4 points)**
- **Check weekly reports relative to the bus fleet performance (3 points)**
- **Add buses, devices, routes, drivers, schedules and stops to the bus fleet (2 points)**

- **João (20 years old)**

- **João checks the expected time of arrival of a bus in real-time from closest stop or given a certain route.**

Motivated by wanting to get in time to college, João opens the mobile application to check the expected arrival time of the bus at the stop. He authorizes the app access to its location so that he can get closest bus stop from his position. In the main interface, he receives a welcoming message and a general guide to the app. João notices that the closest stop to him is highlighted at the top of the app. He is now able to explore all the possible bus routes for that particular bus stop. He selects the desired route. Finally, he checks the expected arrival time of the bus. The bus is not late so he does not need to wait more time than expected.

- **João checks bus details (occupation)**

  João got early to Aveiro from his home town so he has more freedom choosing a bus schedule to his college. He wants to avoid buses full of people so that he has a seated spot in the bus. João opens the application and chooses the desired route. He is informed of the current occupation of multiple buses all to the same destination. He can now choose the best fit for his needs at that particular moment.

  User Stories

  - **Check bus routes starting in the closest bus stop (3 points)**
  - **Check the expected arrival time in real-time and if there is a bus delay (4 points)**
  - **Check bus occupation (4 points)**

3 Architecture notebook

Key requirements and constrains

- Commuters must be able to access the system through the mobile application without any authentication.

- Administrators must be able to access the system through the web application with authentication. They must provide their username and password. This information must be stored in the database. Passwords must be hashed before being stored.

- Commuters must have access to the bus schedules for any given stop, including the expected time of arrival of buses in real-time for every route. There must be a mechanism in place to detect the closest stop to the commuter, allowing for the implementation of a home page that quickly displays all the information regarding that specific stop.

- Administrators must have access to all information regarding the bus fleet. They must be able to assign monitoring devices, drivers and schedules/routes to any given bus. They must be able to monitor real-time statistics and historic data for any given bus.

- The web and mobile applications must be continuously available.

- The monitoring device on board of every bus must periodically send information such as location, speed, fuel consumption and number of passengers to the system over mobile data network. Every update must be sent every 5 seconds.

- Messages sent by monitoring devices must enter a message queue in the system. Every message received must be properly parsed and the currently delay should be determined. The data should be persisted in a database.

- The system must provide an API with endpoints for the mobile application (such as closest stop to the user, the next schedules arriving at a given stop and information about those schedules), allow CRUD operations over every entity and retrieve historic data of the bus fleet for statistical analysis.

- Similar API call involving real-time information (in the mobile application) must be cached for at least 20 seconds to allow for a better system scalability.

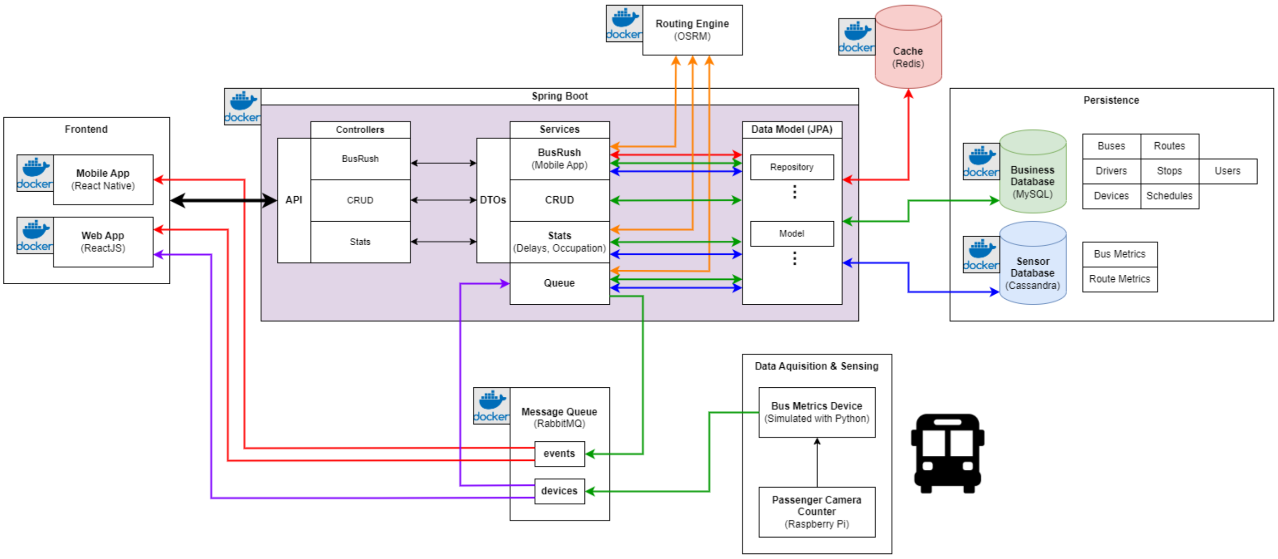
Architetural view

The architecture of the system is composed of 5 main parts:

- **Data Acquisition & Sensing:** collects real-time metrics from the bus and sends it to the message queue. A camera is connected to a script running on a Raspberry Pi and is used to count the number of passengers by determining the number of people who enter and leave the bus.
- **Message Queue:** provides two queues – one for the metrics received from the buses (devices) and one for events (such as delays) detected over the real-time metrics (events).
- **Spring Boot:** provides and API for the frontend, uses a routing engine to determine the duration of a trip by road given points on the map, receives the metrics and sends

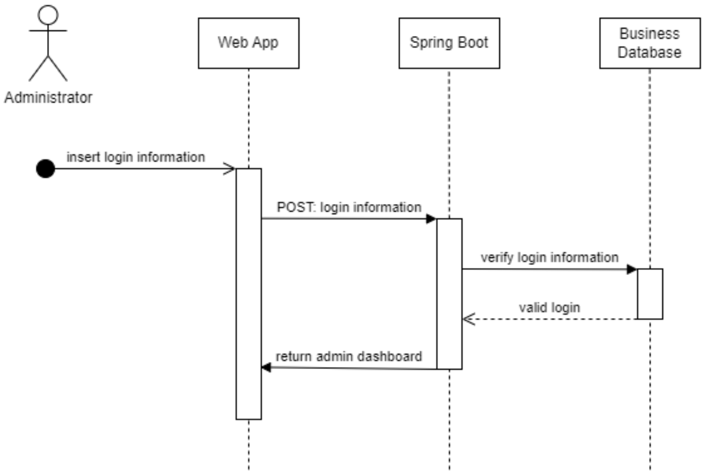
events to the message queue, has a caching mechanism to answer similar requests from the API and communicates with the business database and the sensor database to fetch the required data needed for any given operations.

- **Persistence:** the business database contains information about buses, drivers, devices, routes, stops, schedules and users (the administrator accounts); the sensor database contains a record of all collected metrics from the buses, organized by bus and organized by route (to help with bus and route specific queries). The stored data includes: bus id, route id, device id, timestamp, location, speed, fuel, passengers and delay.
- **Frontend:** the web and mobile applications communicate with the system's API to fetch information to show the users. The mobile application fetches events from the message queue to alert users of a bus being delayed; the web application fetches the metrics from the message queue to display the buses moving on the map in real-time and to show their current metrics.

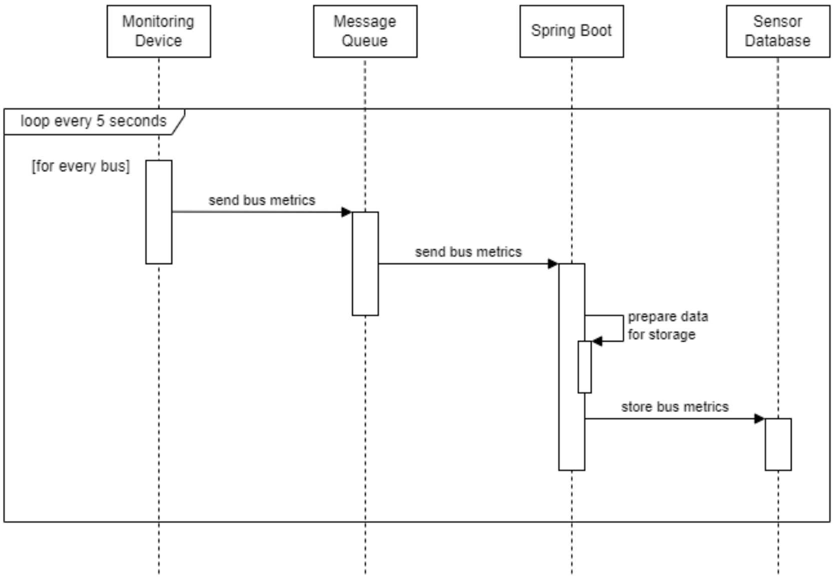


Module interactions

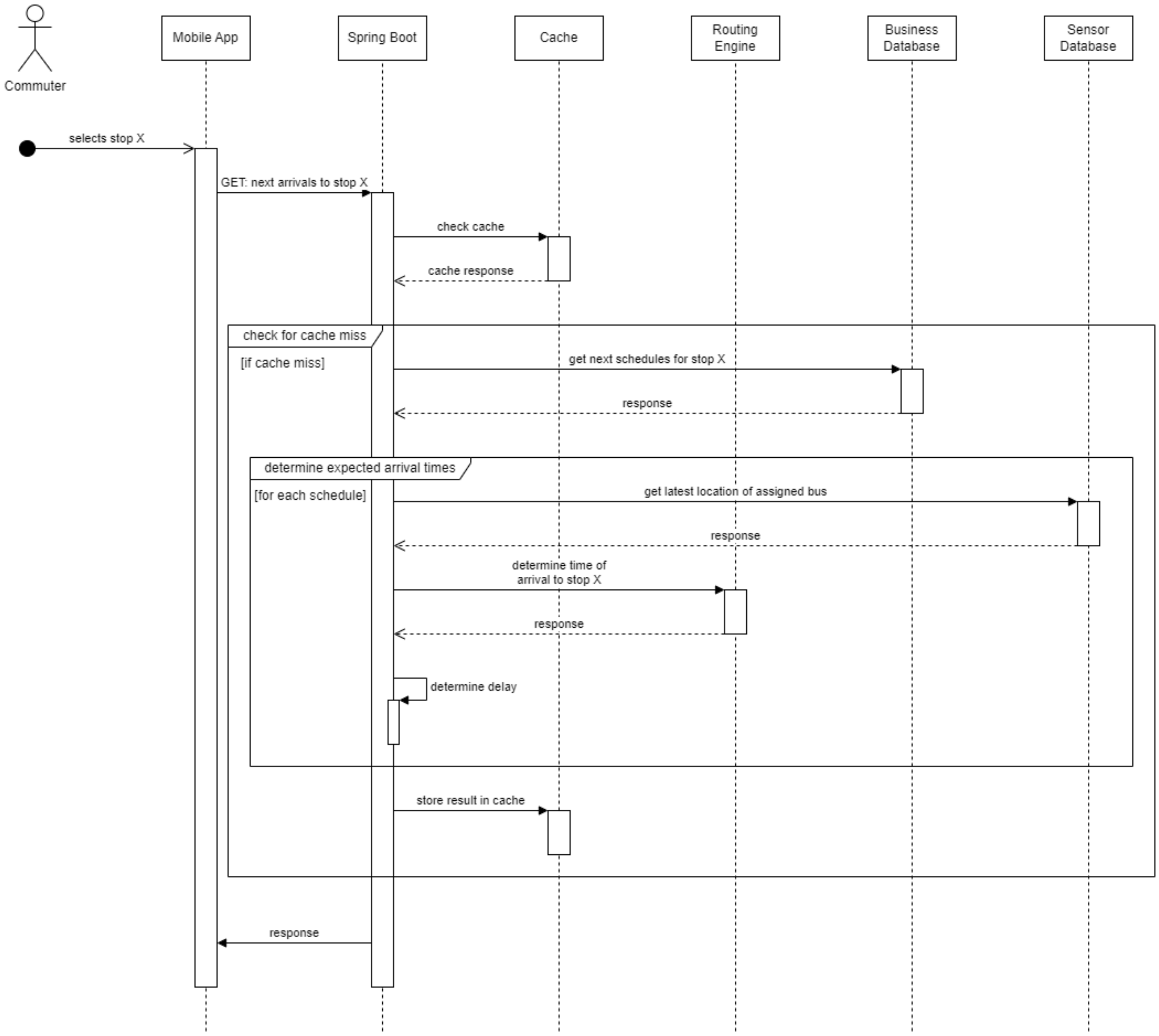
To access the administration dashboard in the web application, the administrator must provide its login information to the system. The validation of the provided credentials is checked by comparing them to the information stored in the business database.



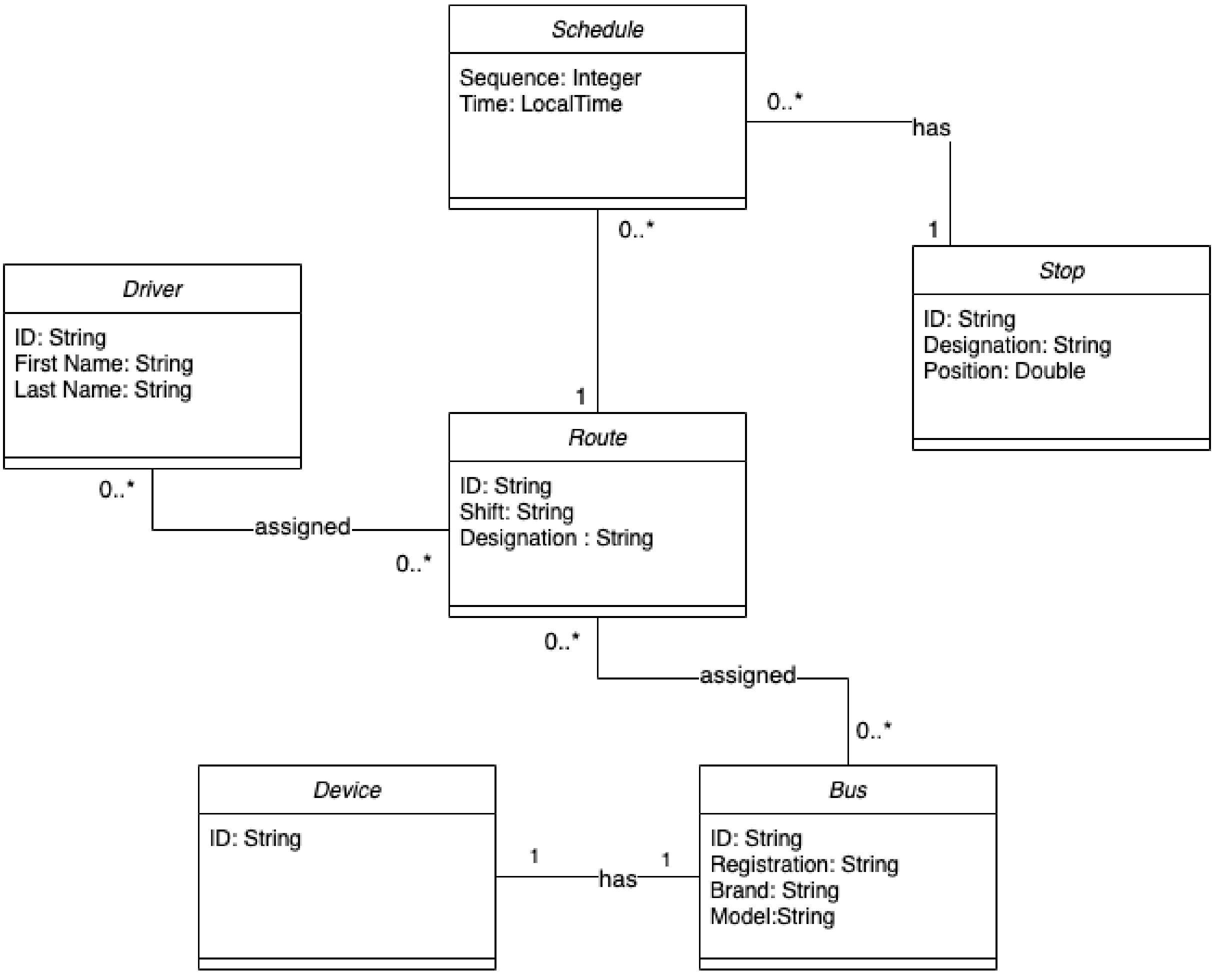
On board of every bus, the monitoring device periodically sends bus metrics to the system's message queue, that forwards it to the backend to prepare it or persistence in the sensor database.



While using either the mobile application, a commuter might select stop X to check all incoming buses to it. That request is done to the system's API. Spring Boot checks if a request for stop X was done in the last 20 seconds by checking the cache. If there is a cached response, return it to the application. Otherwise, the system should query the business database for all the next schedules arriving for stop X, and for each of the schedules, determine the expected arrival time (and delay) for the assigned bus. These results are then cached with a TTL of 20 seconds.



4 Information perspective



5 References and resources

1. Personas photos are AI-generated in this website https://thispersondoesnotexist.com/
2. Diagrams were created using draw.io

   https://draw.io/
3. Article that mentions monitoring of bus fleet is correlated to decrease of delays in Lisbon: https://lisboaparapessoas.pt/2022/05/23/carris-metropolitana-tml-pontualidade/?doing_wp_cron=1667664935.9123439788818359375000