Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Public Transport Management Systems with Real-Time Passenger Information

Catarina Alexandra do Espirito Santo Nunes

Orientador(es) | Vitor Beires Nogueira

Évora 2022

Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

Public Transport Management Systems with Real-Time Passenger Information

Catarina Alexandra do Espirito Santo Nunes

Orientador(es) | Vitor Beires Nogueira

Évora 2022

A dissertação foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor da Escola de Ciências e Tecnologia:

Presidente	|	Lígia Maria Ferreira (Universidade de Évora)
Vogais	|	José Saias (Universidade de Évora) (Arguente) Vitor Beires Nogueira (Universidade de Évora) (Orientador)

Évora 2022

For my family.

Agradecimentos

I would like to thank my advisor, Vítor Beires Nogueira, for all the encouragement, support and availability given during the course of this project.

I want to thank my parents and sister for their help, care, guidance and for always believing in me.

I also thank everyone who helped me in some way, heard me talk about this project and shared their experiences with public transport, they inspired me to continue forward.

Lastly, I would like to thank my husband for his help, kindness, and understanding when I needed the most, without hesitation. I am thankful to have you by my side.

Contents

List of Figures

List of Tables

Acronyms

GTFS General TransitFeed Specification

API Application Programming Interface

TfL Transport for London

ITS Intelligent Transport System

CH Contraction Hierarchies

LC Linked Connections

CSA Connection Scan Algorithm

EMT Empresa Municipal de Transportes de Madrid

BVG Berliner Verkehrsbetriebe

RATP Régie Autonome des Transports Parisiens

TPG Transports Publics Genevois

NeTEx Network Timetable Exchange

SIRI Standard Interface for Real-time Information

CAD/AVL Computer-Aided Dispatch / Automatic Vehicle Location

CEN European Committee for Standardization

VOSA Vehicle and Operator Services Agency

OS Operating System

SDK Software Development Kit

UI User Interface

UX User Experience

JIT Just-In-Time

DRT Demand Responsive Transport

GRASP Greedy Randomized Adaptive Search Procedures

RTPI Real-Time Passenger Information

RAPTOR Round-Based Public Transit Optimized Router

OBA OneBusAway

OTP OpenTripPlanner

Public Transport Information Systems with Real-Time Passenger Information

Abstract

Public transport information systems play a key role in the usage of transport services by providing information and tools so that users can navigate complex transport networks. Included in those systems are often applications that explore topics such as accessibility, transport occupancy and journey planning. Developments in the area have been made thanks to the collaboration between public agencies and the open-source community through the usage and availability of real-time transport data. Here we evaluate the available solutions, the underlining problems in the public transport space and build a solution for a public transport mobile application for the city of Évora, Portugal.

Keywords: transport; application; passenger; information; real-time;

Sumário

Sistemas de informação para transporte público com informações para passageiros em tempo real

Sistemas de informação para transportes públicos têm um papel-chave no uso de transportes públicos, ao fornecer informação e ferramenta que permitem aos utilizadores navegar redes de transporte complexas.

Nesses sistemas é comum a exploração de temas como a acessibilidade, ocupação de transportes e planeamento de viagens. Os desenvolvimentos recentes na área foram realizados graças à colaboração de agências públicas com a comunidade de desenvolvimento de código aberto através do uso e viabilidade de dados de transporte em tempo real.

Aqui as soluções disponíveis no mercado e os problemas subjacentes ao espaço dos transportes publico são avaliados, e uma solução para uma aplicação para os transportes públicos de Évora, Portugal, é desenhada e desenvolvida.

Palavras chave: transporte; aplicação; passageiros; informação; tempo real

1

Introduction

This chapter introduces the main topics and aspects covered by this dissertation.

The aim of this dissertation is to explore the context of public transport applications and build a solution of a public transport mobile application for the city of Évora, Portugal. On the daily basis, users of public transport are expected to navigate and make decisions on complex transport ecosystems, sometimes with limited access to data and tools to do so.

Transport data has long been shared by transport agencies as open data that can be used and re-shared. This dissertation presents the main key players and open source projects in the area of public transport information systems, such as Google and OneBusAway (OBA), that have used transport data in their products and constitute the state of the art in this space, in Chapter 2. To set up a baseline of what is expected for the solution to be developed, local transport mobile applications from relevant metropolitan areas were analysed in Section 2.2.

To further investigate into public transport information systems, three relevant problems were identified and explored, accessibility, transport occupancy and journey planning. Accessibility is a topic that encompass all aspects of the development of transport apps, it can be addressed from a practical side (applications to make transport more accessible) and from a development side (how to make transport application more accessible). Both topics are explored in Section 3.1. Transport occupancy is a prevalent concern that can influence decision-making when correctly evaluated. Multiple strategies are suggested as mitigation strategies for low occupancy, such as Intelligent Transport System (ITS) systems (that often include mobile applications as information sharing platforms) and Demand Responsive Transport (DRT). The current known effects of the COVID-19 pandemic in transport occupancy are presented in paragraph 3.2. One of the most used features on public transport applications is journey planning, the underlying systems and algorithms of these kinds of apps strive to find a set of optimal journeys within a certain time range and parameters. In Section 3.3 the most relevant journey planning algorithms are presented.

In Chapter 4 the practical aspects of Public Transport Application Development are presented, such as the details and comparison between the public transit data specifications (the GTFS feeds, Network Timetable Exchange (NeTEx), Standard Interface for Real-time Information (SIRI) and TransXChange), mobile development methods and the most relevant mobile development frameworks and tools.

The work related with the goal of this dissertation is presented in Chapter 5, where the motivation and contexts of solution for a public transport mobile app for the city of Évora are presented. The work is divided into an introduction of the motivation and contexts, the requirements of the solution, the design and implementation of the solution, and the evaluation of the solution.

Finally, in Chapter 6 some conclusions are showed, and some remaining considerations are discussed.

2

Background

This chapter presents a background into transport information sharing, in particular the key players and open-source solutions in the subject. Then an analysis of the existing local public transport apps is made, to provide a context into this kind of applications.

2.1 Key Players and Open-Source Solutions

Transport systems are usually complex ecosystems by necessity, however they are expected to provide the necessary visibility for users to be able to navigate them. In the past, sharing the necessary transport information with users was with printed timetables and nothing else, now, besides the traditional methods for transport information sharing it is expected for it to be widely available online in a standardized format. The developments of transport information sharing as we know them today are relatively recent, so much that as late as 2005 no standard for public transit information existed. Today, public transit information can be easily found online and the source usually is Google and this is not coincidental as GTFS, the current standard for public transit information, was developed at Google and originally stood for Google Transit Feed Specification [Sta].

Back in 2005, employees from the TriMet[1] public transit agency of the city of Portland in the United States, noticed that passengers had difficulty navigating public transit due to scattered information in different schedule formats[Pio13]. After reaching out to Google, GTFS was developed and the city of Portland was the first area to be supported in Google Transit, the trip planning service from Google.

After the successful launch in Portland, Google offered Google Transit for free to any transit agency that formatted and maintained their data in the GTFS format, this decision settled Google as the main service provider for public transit information provider worldwide and GTFS static as the most used data format to describe fixed-route transit service in the world. In 2010 GTFS was changed to General Transit Feed Specification [Spe] to better represent the wider use of the format by application developers not related with Google [Rem].

Many transit agencies choose to share their GTFS data with the public, while other choose to restrict access only to selected partners (e.g., Google), some also provide route planning Application Programming Interface (API)s with the goal of fostering the innovation and creativity of anyone that wants to access and query route information quickly. GTFS data can be found in open transit data directories like Transitland[2] and OpenMobilityData[3]. Since this shift to open data, third-party developers not affiliated with Google, were able to use it for applications like trip planning, maps, timetable creation, mobile data, visualization, accessibility, analysis tools for planning, and real-time information systems.

Repositories of open data often cite TfL[4] for the availability of their static data files, real-time feeds and APIs of the transport networks of London. In 2017, there were over 80 TfL data feeds covering operational and corporate information across all modes of transport, with around 75% of the data available via APIs [fL17]. Over 17,000 developers have registered to use the open data from TfL, and their unified API [5] powers over 600 travel apps in the UK [Opea]. A 2017 assessment of the value of TfL's open data and digital partnerships determined that the open data supporting 42% of travel apps and real-time alerts used by Londoners saved between £70m and £95m p.a.[6] in saved time, reduced uncertainty and lowered information costs, as well as £14m pa in gross value added to London's Tech economy [fL17].

The evidence described above settled transit apps as a successful example of open data usage that benefits all the different parties involved, including developers, entrepreneurs, transit companies and users. Transport apps that are further examples of open data usage are CityMapper, Transit App, Ally, Moovit, among others. Besides the known successful results, resistance to share and engage with open data can still be found in many organizations, and this is generally attributed to political will and constraints.

ITS Directive of the European Union is one example of a policy that supports the implementation of ITS initiatives through the use of financial and legislative instruments with public authorities across Europe. ITS are defined as advanced applications that integrate telecommunications, electronics, and information technologies with transport engineering in order to plan, design, operate, maintain and manage transport systems. They aim to provide innovative services for different modes of transport and traffic management in order to enable users to use transport networks in an informed, safe and more coordinated way [EUR].

On a practical note, the CIVITAS[7] program created by the European Commission works to foster political commitment and boost collective expertise in the areas of sustainable urban mobility by enabling local authorities to develop, test and roll out measures to help achieve the mobility and transport goals set by

[1] `https://trimet.org/`
[2] `https://www.transit.land/`
[3] `https://openmobilitydata.org/`
[4] The public transport agency for Greater London, England.
[5] The abstraction layer that handles communication with many APIs and back end data models.
[6] per annum (meaning each year)
[7] `https://civitas.eu/`

the European Commission, including the European Green Deal. It supports the creation of products and projects in the areas of active mobility, behavioural change and mobility management, clean and energy-efficient vehicles, collective passenger transport and shared mobility, demand and urban space management, integrated and inclusive planning, public participation and co-creation, road safety and security, smart and connected mobility and urban logistics.

One successful project in the area of public transport information systems is OBA[8], an open-source project maintained by the non-profit Open Transit Software Foundation made of software components that store, manage and deliver transit information. It includes end-user applications for multiple platforms such as Android and iPhone native apps alongside with end systems and APIs to facilitate the development of third-party applications.

Similarly, OpenTripPlanner (OTP) is an open source multi-modal trip planner, that allows to plan journeys that combine multiple transportation methods, such as bicycling, bike sharing and walking in combination with the traditional public transport. The journeys planned with OTP can be customized to include optional routing criteria, like the fastest trip, the fewest transfers , the most bike-friendly routes, wheelchair accessibility and the maximum walking distance to public transport. The main architecture of OTP consists of a graph builder and a routing engine written in the Java programming language, accessible through a RESTful API, as well a debug user interface (i.e., map and visual representation of route)[OTP].

Another similar project in the area is Transportr, an open-source Android app marketed as a public transport assistant without ads or tracking. Transportr supports many transport networks worldwide through a public transport enabler library that accesses transit data in the Navita.io[9] dataset that subsequently collects raw data from different transit networks and data providers [Trab]. It can display real-time data of departures and delays in supported areas, provide routing and directions on a map with the access to the device's GPS, favourite locations and trips can be saved, however it can not verify the correctness of the data and therefore offers no guarantees for the displayed information.

In the area of public transport routing, GraphHopper[10] is an open-source routing engine that can calculate ideal routes for different modes of transportation for anywhere in the world, by default it uses the comprehensive map data from OpenStreetMap[11] and GTFS data. Developers can use it as a Java library or a server for very specific use cases since it allows for custom routing for personal preferences, vehicle properties or characteristics of the map data like avoiding motorways, ferries, toll roads or entire areas, as well as specific OpenStreetMap properties like surface, maximum speed, hazmat (hazardous materials), maximum weight etc. GraphHopper supports the Dijkstra and A* routing algorithms and its bidirectional variants, as well as the use of Contraction Hierarchies (CH) for faster as lighter (less RAM) responses, and it does not use heuristics. The downside of using CH is that it allows only pre-defined vehicle profiles (can use multiple) and requires a time-consuming and resource-intensive preparation. It can also be run in a hybrid mode that is flexible regarding changing properties per request or, e.g., integrating traffic data.

In 2015, another framework was proposed to improve the sharing of transport data, called Linked Connections (LC)[12]. The idea behind LC is the same as Content Delivery Networks, whereby creating small fragments of data in cacheable documents it will allow the processing to happen on the infrastructure of the data user therefore the raw data needed by users is published cost efficiently by the provider. This translates

[8] https://onebusaway.org/

[9] Navitia is an open-source web API, with and an extensive collection of datasets from public transport feeds. Developments on Navitia are lead by Kisio Digital, a subsidiary of Keolis (itself a subsidiary of SNCF, the French national railway company). https://www.navitia.io/

[10] https://www.graphhopper.com/

[11] OpenStreetMap is a collaborative project to create and distributes free geographic data for the world, https://www.openstreetmap.org/

[12] https://linkedconnections.org/

to the provider instead of sharing data dumps or route planning APIs, it would share a fragmented version of the data set in documents that can be published over HTTP making the ecosystem web querying based and where route planning happened on the infrastructure of the data user [Lin].

The representation of those small fragments of data, linked connections, are in this context the movement between one transit stop to another, defined by the location and time of departure and the location and time of arrival. The suggested approach made by the creators of LC in [CLV+15] is that an array of sorted connections is made available through paged documents with Hydra [13] links that are downloaded from the Web as a Connection Scan Algorithm (CSA) advances and calculates the most optimal route.

2.2 Local Public Transport Apps

To better understand the current environment in the area of local public transport apps, this section provides brief analysis of the local public transport apps from major urban centres. The public transport apps mentioned below were commissioned or developed by local public transport companies. Their features and capabilities paint a picture of what is expected from a public transport app at this moment in time.

From the region of **London** in England, the TfL Go app advertises to be able to help plan easy and accessible journeys and help avoid the busiest times under social distancing measures. The figure 2.1 shows some relevant screenshots of the app. From usage and observation, the following characteristics stand out:

I The app home screens show an interactive map, with options to visualize the accessibility facilities available at each stop in the top right corner of the screen;

II The main app navigation is horizontal and requires a scroll action to view all the navigation items. Navigation is aided by icons, with a navigation title and auxiliary sentence to ensure a clear navigation;

III The stop screen included information about the quiet times in that location. Through vertical scrolling, you can see the transport routes that operate on the stop and their direction, accompanied by when they will arrive to the stop;

IV A further drill down into a route that operates on this stop will show the next stops of this route and their predicted times;

V The directions screen gives options for journeys not only calculated using public transport but also has in mind common preferences and accessibility concerns like cycling, step-free and bus only journeys.

From the city of **Madrid** in Spain, the EMT app has all the necessary features of a transport app. The app design looks current, however some text is hard to read, and the translation is not consistent across the app. The stand-out features are marked in figure 2.2:

I It is possible to choose favourite stops, places, and routes for ease of access;

II In the stop detail screen it is possible to filter the results of the routes that operate on that stop, preventing confusion when visualizing busy stops;

III The route details show the direction of the route, the arriving time and only the next arriving time;

IV The line detail screens shown limited information about the line, a time range of frequency of which the line stops at the station, the first, and last departure from the station and a button to see the route diagram;

[13] The vocabulary of hypermedia-driven Web APIs, `http://www.hydra-cg.com/spec/latest/core/`

Tfl Go

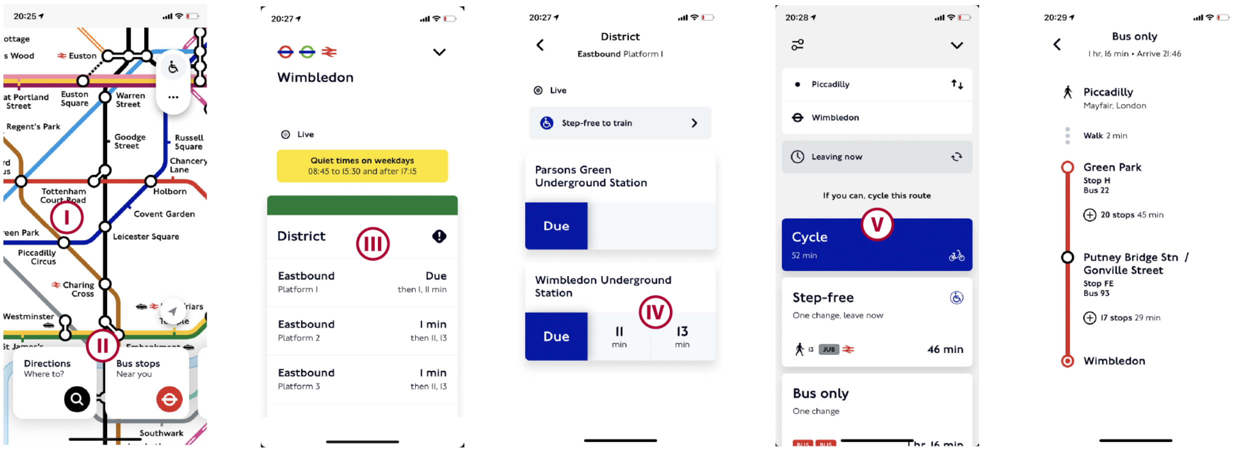

Figure 2.1: TfL Go London transport app

V Other relevant information like the map of the route and the disruptions to the line are available in tab at the top of the screen.

EMT Madrid

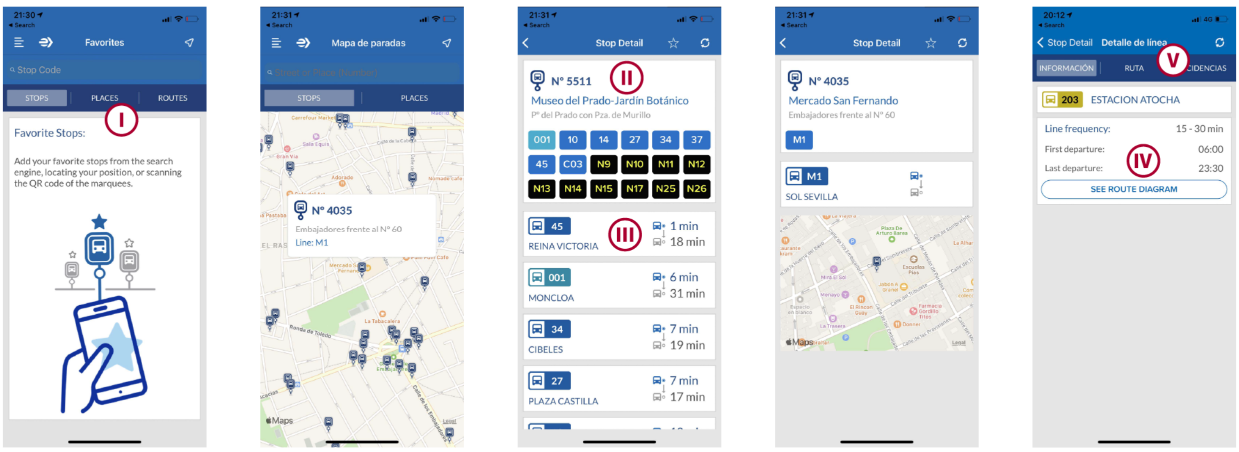

Figure 2.2: EMT Madrid transport app

The BVG Fahrinfo app combines all the transport information and ticket booking for **Berlin and Brandenburg**. The dark background of the app make it feel very modern, and most screens have a lot of information condensed. The app marketing materials set a big focus on personalization (marking locations as favourites and set up of alarms) and mobile payment. The stand-out features are marked in figure2.3:

I Through a hamburger menu, all the features are showed in list including some uncommon features like tickets and emergency contacts;

II In the Map screen, timetable information is shown at the bottom of the screen along with some relevant safety information;

III The route details show the time when the route passes each stop instead of the minutes or hours like other apps;

IV The Map screen shows all the transports moving on the map in real-time, this can give more confidence than only knowing the time when the transport will arrive at a stop;

V The results on the map can be filtered, for a more personalized experience.

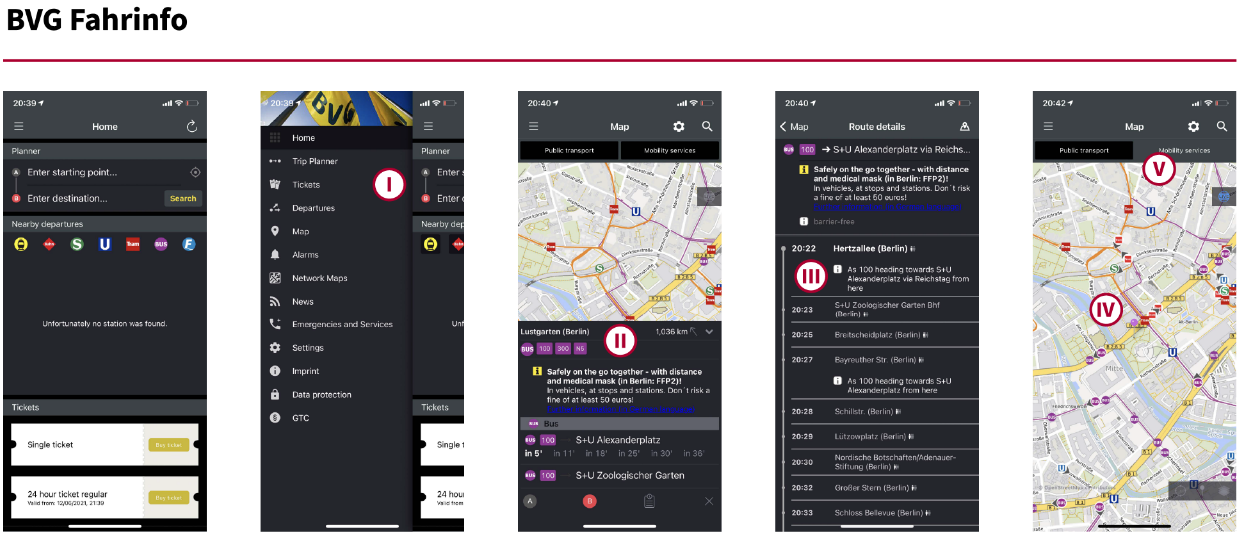

Figure 2.3: BVG Berlin

The RATP app combines all the transport information and ticket booking for **Paris and the Île-de-France region**. This app feels light and information is clearly displayed, even though some text size is small at time. The stand-out features are marked in figure 2.4:

I When viewing schedules, the interface allows filtering between different methods of transport like the metro, tram, or bus;

II In the map screen, the interface allows for filtering and each line is clearly marked;

III When viewing the schedule for a route on a particular stops, all the top times are displayed and the stop is highlighted on the map;

IV The schedule shows multiple predicted arrival times for the selected route at the stop;

V Text notifies the user that only the two firsts arrival times are certain, the ones after are only theoretical and for reference only.

The TPG app provides live service updates and transport information for the city of **Geneva** in Switzerland. The app is targeted at regular travellers, it is noticeable a clear choice to prioritize the display of information above the design aesthetic. This is noticeable in the stand-out features marked in figure 2.5:

RATP

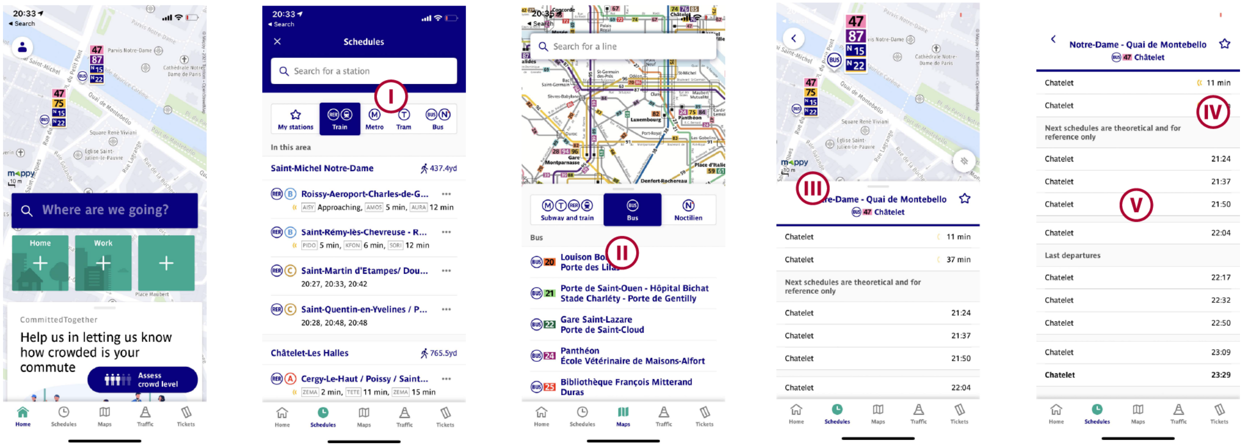

Figure 2.4: RATP Paris

I The home screen of the app is the timetables view, where users can search straight away for stops and add them to their favourites;

II The navigation menu is at the bottom of the screen, where not all the navigation is visible;

III The stop view shows all the routes in a list along with the new arrival time for each route;

IV The route view shows a timeline of the stops and predicted arrival time of each stop along with icons with the reference of intercepting routes at each stop;

V After selecting a stop in the route view, the next stop view shows all the routes that will arrive at the stop after the previous selected route, along with a banner with the predicted arrival time of the selected route at that stop;

VI Delays and notifications for routes and showed at the top of the screen to ensure that the user is informed during their journey.

The Carris app operates **Lisbon**'s buses, trams, funiculars (Glória, Lavra, and Bica), and the elevator (Santa Justa), meaning that it shows a comprehensive view of the public transport in the capital city of Portugal. This app is the baseline for what it is expected for public transport app in Portugal, making it the most relevant analysis of all the local transport apps. The stand-out features are marked in figure 2.6:

I The app allows adding different types of favourite stops like home and workplace;

II The navigation is at the bottom of the screen, and it is clear and simple;

III The stops are displayed based on location and show a detailed view of the next routes and their arrival times;

IV Selecting a route will prompt the user to set up an ETA (Estimated arrival time) alert;

V The stop screen shows the stop on the map and the routes below;

VI The route view will highlight the route on the map and show the following stops below.

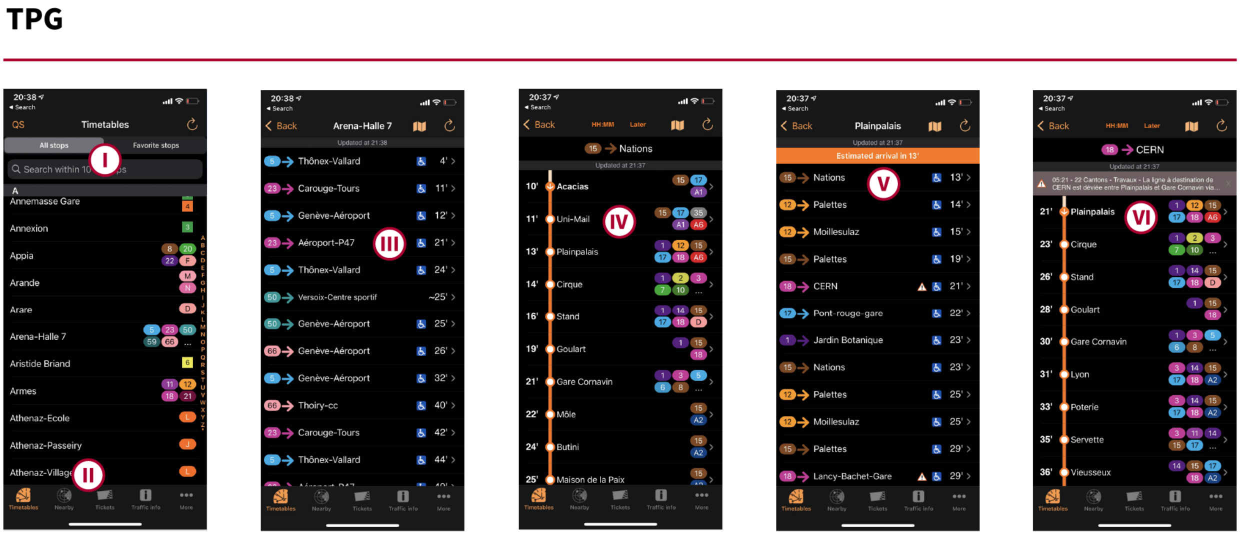

Figure 2.5: TPG Geneva

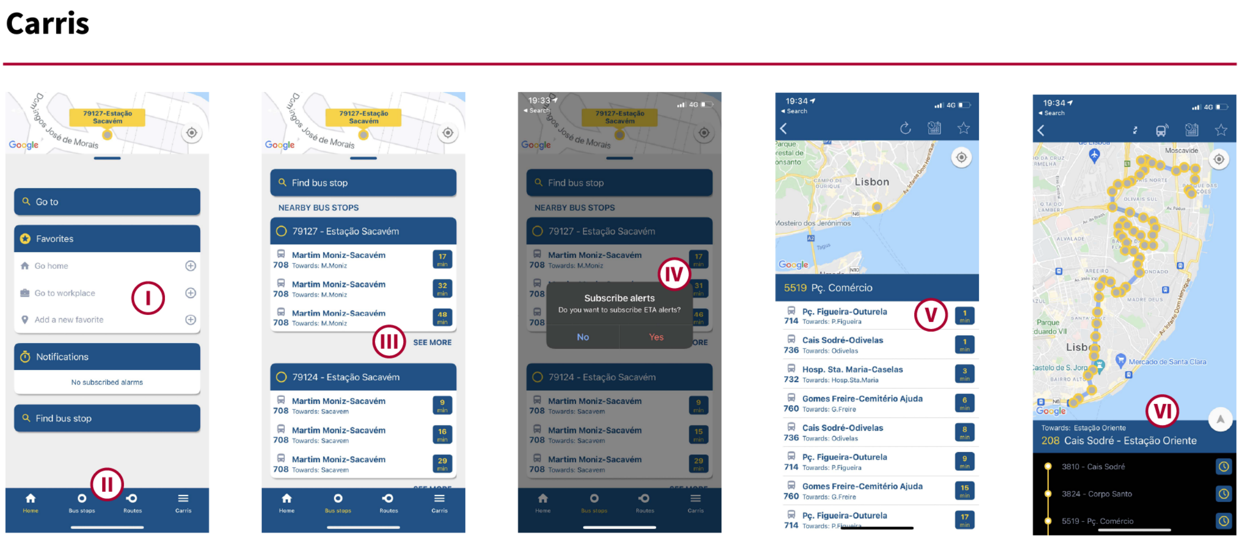

Figure 2.6: Carris Lisbon

3

Problems in Public Transport Applications

This chapter presents selected problems in the field of Public Transport Applications. An analysis of accessibility in transport applications is showed, with examples of current accessibility efforts in the field. The context of transport occupancy data is presented, with the details of how data specifications hold this data. The applications of the routing problem in the field of Public Transport are presented, as well as the algorithmic solutions for it.

Most public transit applications deal with challenges that go beyond the simple presentation of transit information to how to best make this information useful to users. The most common approach is by providing features not directly affected by transit like accessibility features, routing features and, as of recent, passenger count data and transport occupancy features.

3.1 Accessibility in transport applications

The UN Convention on the Rights of Persons with Disabilities states that persons with disabilities should be enabled to live independently and participate fully in all aspects of life, with access, on an equal basis with others, to the physical environment, to transportation, to information and communications, including information and communications technologies and systems [Art]. Therefore, it is indispensable to ensure that all transport applications and information systems are accessible and provide relevant information to maximize usability for users with disabilities.

Accessibility in practice Multiple companies have made efforts to champion accessibility in their services and applications, one example is the TfL Go app and Journey Planner from Transport for London[1] (TfL). In both services it is possible to select 'accessibility and travel options' to specify if the user can use stairs or escalators, which types of transport to use, how far they are prepared to walk, and if they require step-free access in case they can not manage the step or gap onto a train. Other strides for accessibility [Pla] made by TfL are:

- Live bus arrivals information for any bus stop by text message using the bus stop code on a sign above the timetable;
- All buses have audible and visible announcements with the name of the stop, the route, and the destination of the bus.
- Countdown signs at 2,500 bus stops that show which buses are coming and when they will arrive.

Specifically designed for the effect, the Transreport app was created to focus on inclusion and accessibility for passengers with disabilities using UK rail. The app supports voice control and enables passengers to request assistance, review their journey and update their profile using their smartphone [Trad]. Previously, passengers that required assistance to travel in the UK rail would need to call the train company 24 hours in advance and give their personal details for every request, in a process that could take up to 40 minutes. Station staff would only have access to a printed document of the assistance's booked that could not be updated if any alterations were necessary, leading to errors and wasted time. With the app, the user only needs to fill their personal details in the app once and alterations are shared instantaneously with station staff though the app on their phone, per email as a backup, as well as notifications 30 minutes before the passengers' arrival.

Accessibility features in applications Relating to user interface and experience design, the transport app Moovit is an example of community engagement for accessibility efforts. With the help and feedback from a blind developer, Adi Kushnir, using the VoiceOver and TalkBack screen reader native functionalities on iPhone and Android, the company was able to optimize their app for people with vision impairment thanks to a feature that allows users to hold their finger on the phone screen to hear which button or icon is beneath it [Wor]. The company also asked 550,000 local contributors to help map transit systems for the app and identify wheelchair-accessible stations in their cities, this enabled the company to add a feature that shows only routes with stations with ramps and elevators [Trac]. Other specific user interface design functionalities that Moovit made available are:

- Menus and buttons for easier use with one hand, especially on large phones, for users with hand motor disabilities;
- The name of the line is displayed, instead of just a coloured dot or symbol (a space-saving practice in many maps), for people who are colour-blind and use colour-coded transit systems;

[1] Journey Planner from Transport for London can be found at `https://tfl.gov.uk/plan-a-journey/`

- Assurance that no text will be broken or overlapped when a user needs to magnify the font.

The company also advertised that it is studying how to use a phone's vibration and torch to serve users with hearing loss and continues to work with people with disabilities to improve or customize the app [Trac]. The efforts described stand as a good example of the use of the community to improve a product, and the features archived by Moovit are golden standards for accessible features in the field of public transport applications.

Accessibility data Some information about the accessibility of transit services can be obtained from the standard open-source data provided by transit agencies. The level of detail of the information may depend on the data standard used. The GTFS static standard includes the optional field of 'wheelchair_boarding' in the stops.txt file to indicate whether wheelchair boardings are possible from the location[2], but no other information regarding accessibility is included in data sets of this type. NeTEx is designed to support accessibility attributes of fixed infrastructure (e.g, stations, stops), vehicle types used for specific journeys and micro-navigation through interchanges such as equipment for different disabilities (e.g., wheelchair spaces, navigability, tactile strips etc) and accessibility services (e.g., boarding assistance). In the NeTEx model, entities' describe accessibility by the user need (e.g., wheelchair, hearing impaired, vision impaired, lift-averse) and or accessibility limitation of the location (e.g., Wheelchair, Step free, Escalator free, Lift free), grouped as what is called an Accessibility Assessment [Bou15]. NeTEx is a data standard flexible enough to allow for its models to be sparsely populated with accessibility characteristics in order to achieve basic functionality and later be more fully populated as a long-term effort to enable a high level of functionality when used by journey planning engines.

From the analysis of the examples provided above, it is possible to conclude what are the most relevant accessibility information andżfeatures that are expected from a public transport application. The most prevalent is the support for screen readers, followed by the station accessibility information. However, other features are also equally important to ensure accessibility in a mobile app, so it is recommended for accessibility testing to be performed before the final product reaches users.

When creating any product, it is important for it to be accessible to everyone, and in the case of mobile applications, providing an accessible experience will impact positively the overall experience of all users. To accomplish this, mobile applications should be subject to mobile accessibility testing to ensure that the product can be used by anyone, including people with disabilities such as hearing loss or colour blindness.

Mobile accessibility testing can be performed using instrumented tests such as Espresso for Android apps and XCUITest for iOS apps.

Accessibility testing for Android To test the interaction with UI components we can use the test automation framework for Android called Espresso where the components are checked for certain behaviours and conditions. This framework can check for accessibility using the accessibility checks in the AccessibilityChecks class of the framework[3].

Accessibility testing for iOS As for accessibility testing in iOS, developers can use the User Interface (UI) testing framework by Apple, XCUITest. XCUITest uses the accessibility tree of an app, meaning that tests

[2]Full specification can be found at `https://developers.google.com/transit/gtfs/reference`

[3]More information on how to use the AccessibilityChecks class can be found at `https://developer.android.com/guide/topics/ui/accessibility/testing#espresso`

use the same elements as screen readers and UI tests written in the framework will double as accessibility tests if correctly written[4].

Another option for accessibility testing in the iOS platform is GTXiLib, a Google Toolbox for Accessibility. GTXiLib is based on XCTest[5] unit testing framework. GTXiLib allows accessibility checks to be performed along-side with unit tests without writing tests specifically for the effect.

Accessibility principals The Google framework, Flutter, includes framework support for accessibility in addition to what is provided by the underlying operating system[7] in conjunction to documented guidelines for accessibility testing. Particularly usefully for the development of any application, Google suggests the following accessibility release checklist [Acca]:

- ***Active interactions*** — Ensure that all active interactions have a result (e.g., performs an action, triggers an event);
- ***Screen reader testing*** — The screen reader should be able to describe all controls on the page when tapped and the description should be clear. Apps should be tested with TalkBack (Android) and VoiceOver (iOS);
- ***Contrast ratios*** — There should be a contrast ratio between controls or text and the background of at least 4.5:1, except for disabled components. Images should also be vetted for sufficient contrast;
- ***Context switching*** — The user's context should never change automatically while typing in information. Generally, changing the user's context should be avoided without some sort of confirmation action;
- ***Tappable targets*** — All tappable targets should be at least 48×48 pixels;
- ***Errors*** — Important actions should be able to be undone and in fields that show errors a correction should be suggested if possible;
- ***Colour vision deficiency testing*** — Controls should be usable and legible in colour-blind and greyscale modes;
- ***Scale factors*** — The UI should remain legible and usable at very large-scale factors for text size and display scaling.

The assurance that a public transport mobile application is fully accessible will create confidence with users and aide in the adoption of the service and the overall use of the public transport systems it serves, which will impact the occupancy of the system.

[4] More information about how to use XCUITest for accessibility testing at `https://medium.com/@r.whitaker/xcuitests-for-accessibility-43c2bde551a5`

[5] XCTest is Apple's testing framework that allows to create and run unit tests, performance tests, and UI tests for Xcode[6] projects, `https://developer.apple.com/documentation/xctest`

[6] Xcode is Apple's integrated development environment (IDE) for macOS, used to develop software for macOS, iOS, iPadOS, watchOS, and tvOS[Accb]

[7] See how accessibility principles can be followed using Android native features at `https://developer.android.com/guide/topics/ui/accessibility/principles`

3.2 Transport occupancy

A prevalent concern in the area of public transport is if the services provided are efficiently serving the intended communities, one important metric for analysis into this concern is transport occupancy. This metric provides knowledge on the operation and performance of the services in order to improve or create mitigation strategies for any anomalies that may be found.

An example of this can be found in a 2020 evaluation study [BAN+21] of Cyprus' transit system that aimed to enhance the transit system and increase the transit mode share. GTFS data was used to represent the transit network, and combined with monthly transit travel demand information in the shape of passenger boardings, provided a view of into the transport occupancy and use of specific bus lines. The study also provided mitigation strategies for the anomalies found, such as the replacement of frequent services with Demand Responsive Transport (DRT) in cases of underutilized bus routes and the introduction and efficient operation of an ITS.

DRT DRT is a flexible mode of transportation that adapts to the user's needs. Usually, it involves users making requests with the desired pickup and drop-off details and the route and frequency of the service will adapt based on the demand. The benefits of this kind of transport systems compared to regular transport systems are lowered service costs, prevention of low occupancy rates and increased efficiency.

The design of DRT services in the context of economic austerity for case of the city of Porto, Portugal, was researched in 2015 [GdSD15] and a dynamic routing algorithm with Parallel Reactive-Greedy Randomized Adaptive Search Procedures (GRASP) type meta-heuristic was suggested as a more flexible and cost-efficient option when compared to a real DRT nighttime service called "Gato" (that ended operation on April 2012).

Meta-heuristics are high-level technics to find the most optimal solutions to combinatorial optimization problems. GRASP are iterative meta-heuristics, in which each iteration has a construction phase followed by a local search phase. In the construction phase, a solution is found. If the solution is not feasible, a repair procedure should be applied to try to achieve feasibility. If the solution can not be feasible, it will be discarded, and a new one will be created. Once a feasible solution is found, the local search phase will investigate its neighbourhood until a local minimum is found and the best overall solution will be the result [RR16]. The parallel aspect of the algorithm, means that each GRASP iteration is performed in parallel, with only a single global variable required to store the best solution found over all processors [GdSD14]. The designed algorithm is reactive because it includes a reactive mechanism that responds to solutions produced and tries to adjust the greediness of the process [GdSD15].

DRT systems are often suggested as a potential to answer to flaws in transport systems, specially when costs are a concern, but if not designed and realistically costed with a full understanding of the market they are to serve they are prone to failure.

ITS An ITS consists of a coordination of different technologies and applications from different modes of transport and traffic management systems [ITS]. With the use of different technologies from Information Technology and telecommunications, an ITS allows for information to be available from all areas of a transit network for the benefit of users, public and private institutions. Without necessarily having to physically alter existing road infrastructure, using sensor and control technologies, communications, computer informatics, and applications that cross disciplines such as telecommunications, computer science, finance, electronic commerce (such as electronic ticketing) and automobile manufacturing (such as intelligent vehicles), transport systems can be smarter, unlock innovation and better serve the intended communities [Abo].

How the COVID-19 pandemic impacted transport occupancy One important factor that affects the usage of public transport is the confidence of users in the system and the availability of pre-travel information[YCN+19]. The time after mid-March 2020[8], presented a big challenge to transport systems all over the world, as ridership and revenues decreased, transport agencies reported that passenger numbers dropped by up to 90 percent and were expected to remain at low levels for possibly 2 years or longer[UIT20].On an organizational level, most public transport companies adopted measures to try to regain the trust of users such as cleaning/disinfecting, providing masks to staff, and service modifications to better accommodate safe social distancing measures[UIT20].

As publicized, in 2020, the California Integrated Travel Project [FT20] incentivized the usage of GTFS and GTFS-realtime for the benefit of the systems that travel agencies and people used during emergencies and major service modifications or outages, like the ones experienced during the COVID-19 pandemic. It was found that in situations where services were continuously modified, it is costly to and therefore unlikely for agencies to constantly update their GTFS static feeds, so information regarding service modifications were less likely to reach users, than those with GTFS-realtime feeds integrated into applications like Google Maps. An April 2020 study [Who] by the company behind the Transit mobile app found that those still using public transit at the time, when ridership decreased dramatic and service modifications were put in place, were more likely to be women, in lower-income brackets, and African American, Latino, or Native American as compared to transit riders prior to the pandemic. This data shows that it is critical from a social equity standpoint for agencies to look into the best ways to share real-time service information in order to make it easier for vulnerable populations to learn of and cope with service modifications.

With the requirements and efforts regarding social distancing bought by the COVID-19 pandemic, more transit agencies have been experimenting with measuring and providing real-time vehicle occupancy information to users. An 2020 report [Lib20a] examined how this can encourage trust in the safety of public transport and what are the tools to achieve the desired result. It was found that, potentially, there are two dimensions in vehicle occupancy for which to determine passenger loading, the capacity of passengers that the vehicle was designed for and the safe capacity based on current guidance. On the contrary to the designed capacity of vehicles, the safe capacity of a vehicle is a mutable value, and it is the most important to users, so indications of occupancy levels should be made against this baseline.

Transport Occupancy in data specifications As far as how data interfaces accommodate the static dimension of vehicle occupancy, the NeTEx format includes PassengerCapacity as part of the VehicleType and has a structure that allows for detail — including seated and standing capacities. Neither GTFS static nor TransXChange include the ability to provide vehicle seating and standing capacity. Regarding the live data interfaces, both SIRI and GTFS-realtime have appropriate features to deal with live data of vehicle occupancy.

In SIRI, the Occupancy field is one of the optional properties of ProgressInfo of a MonitoredVehicleJourney that can be used in SIRI-SM[9] and SIRI-VM[10]. This reflects the actual current passenger occupancy value in

[8] The period in time when lockdowns related with the spread of the COVID-19 virus were introduced in most countries in the world.

[9] SIRI-SM is the Stop Monitoring service in the SIRI Interface. It allows the exchange of the real-time arrivals and departures at a stop of public transport services.

[10] SIRI-VM is the Vehicle Monitoring service in the SIRI Interface. It allows the exchange of the real-time positions of public transport vehicles.

a journey. There is also, an Occupancy field in the EstimatedCall structure of the SIRI-ET[11] that represents a predicted passenger load.

With GTFS-realtime, the fields that can contain real-time vehicle occupancy information are only experimental [GTFa], and those are:

- ***occupancy_status*** — The degree of passenger occupancy for the vehicle, defined by seven values added to the specification and included in the official reference site, but not formally adopted;
- ***occupancy_percentage*** — A percentage value indicating the degree of passenger occupancy in the vehicle, a new inclusion in the specification on the GitHub site in early 2020, but its adoption level is unclear[12];

With the accurate knowledge of the usage of transport systems, it is possible to make informed decisions and plan improvement efforts that can benefit the communities served by those systems.

3.3 Public Transit Journey Planning

Web-based information services have transformed our expectations of public transport information systems. In the current day-to-day usage of public transport, it is expected and necessary to have access to live schedule updates and the ability to find alternative routes for our journeys. Investments in customer information systems have had a significant impact on bus usage [BMW15], as they help to optimize journeys and improve the quality of the user experience.

Real-Time Passenger Information (RTPI) not only concerns public transport systems, but it impacts the whole transportation system by giving users the correct tools to change their travel behaviour, i.e., choosing more sustainable transport modes can help with the reduction of car use, contributing to road decongestion [MHC13]. RTPI systems ease the use of public transport through both perceived and actual reductions in passenger waiting time and improve passengers satisfaction. Access to this information before the journey can help decrease the actual waiting time, while when information is available to the passenger during the journey, it can help to reduce the perceived waiting time [DS08]. Other likely benefits that can affect the usage of public transport positively are as follows: increased willingness to pay; more efficient travel; positive psychological effects such as less uncertainty, more personal security, more trust in the services, easiness of use and a better overall image of the system [MHC13]. Passengers have reported a higher perceived quality of service when stops and buses are equipped with information devices [MHC13].

Research [MHC13] suggest that investments into RTPI systems, both on-board of vehicles and at bus stops and terminals, makes the image of public transport more competitive in relation with other modes of transportation, particularly cars. The common deterrent factors of the use of public transport, such as waiting time at stops, delays, uncertainty, payment time and lack of information are clearly reduced. It can produce environmental, social and economic benefits because of the increased occupancy of buses and decreased private car usage.

Users gather important information regarding the journey they wish to take by using journey/route planning apps or features (usually part of local public transport apps). Depending on the underlying systems and algorithms of these kinds of apps, they possibly can cover the real-time changes or not.

[11]SIRI-ET is the Estimated Timetable service in the SIRI Interface. It allows the exchange of the estimated real-time timetable for a public transport service along a route.

[12]See the current specification reference in GitHub at `https://github.com/google/transit/blob/master/gtfs-realtime/spec/en/reference.md#message-vehicleposition`

Journey Planning Algorithms

The underlying algorithms of journey/route planning apps or features, generally, try to find a set of optimal journeys within a certain time range from one stop to all other stops of the network [Paj]. The input use are timetables, comprised of a set of stops (e.g., platforms, bus stops, etc.), a set of routes (e.g., bus lines), and a set of trips (the operation of vehicles on routes along sequences of stops at certain times of the day). An optimal journey will depend on the criteria of travel time and the number of transfers. Some algorithms consider multimodal journey planning, where different modes of transport are included in the problem.

Dijkstra's algorithm and its extensions

Usually, the timetable is translated into a graph on which the shortest paths correspond to optimal journeys [Paj]. Those graph models can incorporate different levels of realism. Most algorithms are defined as extensions of the Dijkstra's algorithm. Those algorithms are written in the context of weighted graphs, so all the language reflects that (nodes, costs, etc).

We consider directed graphs[13] $G = (V, E)$ with n nodes and $m = (n)$ edges. An edge (u, v) has the non-negative edge weight $w(u, v)$ [DSSW09]. A shortest-path query between a source node s and a target node t asks for the minimum weight $d(s, t)$ of any path from s to t.

The **Dijkstra's Algorithm** maintains an array of tentative distances[14] $D[u] \geq d(s, u)$ for each node, the nodes are visited in the order of their distance to the source node and maintain the invariant that $D[u] = d(s, u)$ for visited nodes [DSSW09].

When a node u is visited, its outgoing edges (u, v) are relaxed[15], $D[v]$ is set to

$$min(D[v], d(s, u) + w(u, v))$$

The algorithm ends when the target node is visited. The size of the search space[16] of settled nodes[17] is $O(n)$ and $n/2$ nodes on the average [DSSW09], with a run time of $O(n^2)$.

Extensions and speed-up techniques can be added to the Dijkstra's algorithm to reduce run time, e.g., Bidirectional Search, Geometric Goal Directed Search (A∗) and CHs.

Bidirectional Search Dijkstra's algorithm can be run using Bidirectional Search, where two searches are made simultaneously: one forwards from the source, and one backwards from the destination. If a path to a node has been visited from both directions, it is possible to derive the shortest path from the information already gathered.

[13] A directed graph is a graph that is made up of a set of vertices or nodes connected by directed edges, often called arcs, where all the edges are directed from one vertex to another.

[14] The tentative distance means the up-to-date distance between a node to the starting node.

[15] The edge relaxation is the operation to calculate the reaching cost to the vertex lower. For the edge from the vertex u to the vertex v, if $d[u] + w(u, v) < d[v]$ is satisfied, update $d[v]$ to $d[u] + w(u, v)$[Edg].

[16] The search space that includes all possible solutions, including the desired solution, where each point in the space corresponds to one possible solution. [Sea]

[17] Settled nodes are the ones with a known minimum distance from the source.

Geometric Goal Directed Search In **Goal Directed** approaches, the edges that shorten the distance to the target are preferred and edges that cannot possibly belong to the shortest path are discarded. The **A∗** search algorithm can be seen as an extension of the Dijkstra's algorithm, it achieves better performance by using a Goal Directed approach. It does this by modifying the weight of the edge (u, v) to $w(u, v)$–$(u) + (v)$ where (v) is a lower bound on $d(v, t)$. This manipulation shortens edges that lead towards the target [DSSW09]. A∗ is usually used in route planning in road maps, where estimates $d(v, t)$ using the Euclidean distance[18] between v and t and the average speed of the fastest road anywhere in the network.

Contraction hierarchies is a speed-up technique used to optimize graphs. Shortcut edges are created in a preprocessing phase that are then used during a shortest-path query to skip over less-important nodes. In a first phase, nodes are ordered by importance, and a second phase contracts (removes) nodes in this order. When a node v is contracted, it is removed in a way that shortest path distances between the remaining nodes are kept [DSSW09].

Research [Paj] suggests **User-Constrained Contraction Hierarchies** as a more specified approach for multimodal journey planning, where restrictions are imposed on the sequences of transportation modes in the form of a regular language, to which any computed journey must obey.

Alternative approaches

Other alternative approaches exploit the basic elements of public transport networks to calculate routes directly on the timetables, such as Round-Based Public Transit Optimized Router (RAPTOR), CSA, and Transfer Patterns.

RAPTOR On the contrary to previous algorithms, in the RAPTOR algorithm, timetable data is organized into arrays that are then processed in rounds (one per transfer) and each route of the timetable is searched at most once per round[Paj]. With the use of simple data structures, memory locations can be efficiently processed and query performance on complex transport networks is faster by an order of magnitude compared to other algorithms[Paj]. With an extended version of RAPTOR, called McRAPTOR (the more-criteria variant of RAPTOR), it is possible to handle strict dominance, multicriteria range queries, and additional criteria, like fare zones and reliability of transfers. Because RAPTOR does not rely on preprocessing, it can be directly used in dynamic scenarios, including delays, route changes, and trip cancellations [Paj], such as the changes present in real-time transport data. RAPTOR is currently in use by OpenTripPlanner[Opeb].

CSA The CSA, like RAPTOR, is not graph-based. However, unlike RAPTOR, it is not centred around routes, but elementary connections, which are the most basic building block of a timetable. CSA organizes connections into a single ordered array, that is scanned once (linearly) to compute journeys to all stops of the network [DPSW13]. CSA does not include a preprocessing step, making it comparatively simple to others and able to handle dynamic scenarios including train cancellations, route changes and real-time delays, but it also makes the running time inherently dependent on the timetables size [DPSW18]. Multiple extensions are suggested to CSA, such as Connection Scan Accelerated (CSAccel) [DPSW18] and Goal Directed CSA (GDCSA) [FR21], that improve query times in large datasets.

[18] The Euclidean distance is the distance between two points that can be calculated from their Cartesian coordinates using the Pythagorean theorem.

Transfer Patterns Transfer Patterns is a speed-up technique specifically designed for public transit networks. It is based on two key observations: many optimal journeys share the same transfer pattern, i.e., the sequence of stations where a change of vehicle occurs and direct connections without change of vehicle can be looked up quickly [BCE+10]. In the basic use of the algorithm, these transfer patterns are precomputed for all pairs of stops during preprocessing, then given source and target stops, the query quickly builds a search graph of (at least) the relevant transfer patterns. When the query graph is small, the Dijkstra's algorithm can be applied. This process performs proportionally to the squared size of the input data set and the result is of a quadratic size [BCE+10]. With this, global searches are simpler and executed on preselected set of stops where routes intersect and are calculated faster. The Transfer Patterns algorithm is currently in use by Google Transit [BCE+10].

Take aways Many private organizations provide route planning software that can be incorporated into public transport information systems. Knowing and understanding different algorithms will help make design choices for journey planning solutions in public transport information systems and applications.

4

Practical Aspects of Public Transport Application Development

This chapter introduces practical aspects of Public Transport Application Development. The most relevant public transit data specifications such as GTFS, NeTEx and SIRI are presented and compared. The most relevant Mobile Development Methods and Frameworks are also showed.

4.1 Introduction

A good public transport network must be easy to use and reliable. All the information contained in the system should help users to use the public services available. The information is gathered from all the public services and collected in a Public Transport Information System.

These Public Transport Information Systems provide real-time information for users travelling on the public transport network such as schedules, maps, timetable changes, problems in the lines, etc. Most vehicles used, like buses and trains, should provide their users with information regarding the next stops, connections and useful additional information either using voice announcements or on displays [CIV10].

Systems may electronically deliver information through different mediums like a mobile phone application, LED displays inside stations, bus stops and shelters, through a website, or by telephone (either by SMS, a designated service or an automated system). Information should be available without any barriers of access or format, in ways that those who with less digital literacy, people with impairments[1] have access to the information needed. Additionally, a system can provide information in multiple languages and a personalized trip planning tool with features such as displaying opportunities for intermodality (e.g., car-sharing, biking or walking) and the environmental impact and costs of different trip possibilities.

The essential item for any Information System is the data managed and shared by the system. In Public Transport Information Systems, data can be stored and shared in different formats, the most common public transit data specifications are described below.

4.2 Public Transit Data Specifications

Public transit agencies generate a broad volume of mutable data that when shared in a reliable and accurate way can be highly beneficial for travellers and all users of the transit system. The most common specification for transportation data is called GTFS and it is used to generate datasets that can be openly shared with the public and consumed by developers to write all kinds of applications. GTFS feeds can be one of two types, GTFS static or GTFS-realtime.

4.2.1 General Transit Feed Specification feeds

GTFS static feeds contain data of stop and route locations and the scheduled transit services, composed of at **least six, and up to 13 CSV files (with extension .txt)**. Each file consists of comma-separated rows of text and numbers, which combined specify the information designated. The Figure 4.1 shows an example of a GTFS static feed specification and the relationship between files and their entities[2].

While GTFS static only provides transport data that is stable and unlikely to change in a short period of time, GTFS-realtime was designed to deal with the transport data that changes at every moment. It was designed to be easily implemented, compatible with the route and schedule information in a GTFS static feed, and provide accurate and useful time sensitive passenger information.

With GTFS-realtime it is possible to provide three types of updates in continuously updated feeds:

- Trip updates: Delays, cancellations and changed routes
- Service alerts: Stop moved, unforeseen events affecting a station, route, or the entire network
- Vehicle positions: Information about the vehicles, including location and congestion level

For data exchange, GTFS-realtime uses Protocol buffers, a binary format that is compact and easy to validate and will be shared in a feed served via HTTP by any type of web server or even a web application server as a response to a valid HTTP request.

The data structure is defined in the gtfs-realtime.proto schema using Protocol buffers that serialize the data with a language agnostic process. These can then be imported and used to build the GTFS real-time data

[1]The definition adopted by the World Health Organization (WHO) of an impairment is a structural or functional anomaly in the body[Lec17].

[2]Full specification can be found at `https://developers.google.com/transit/gtfs/reference`

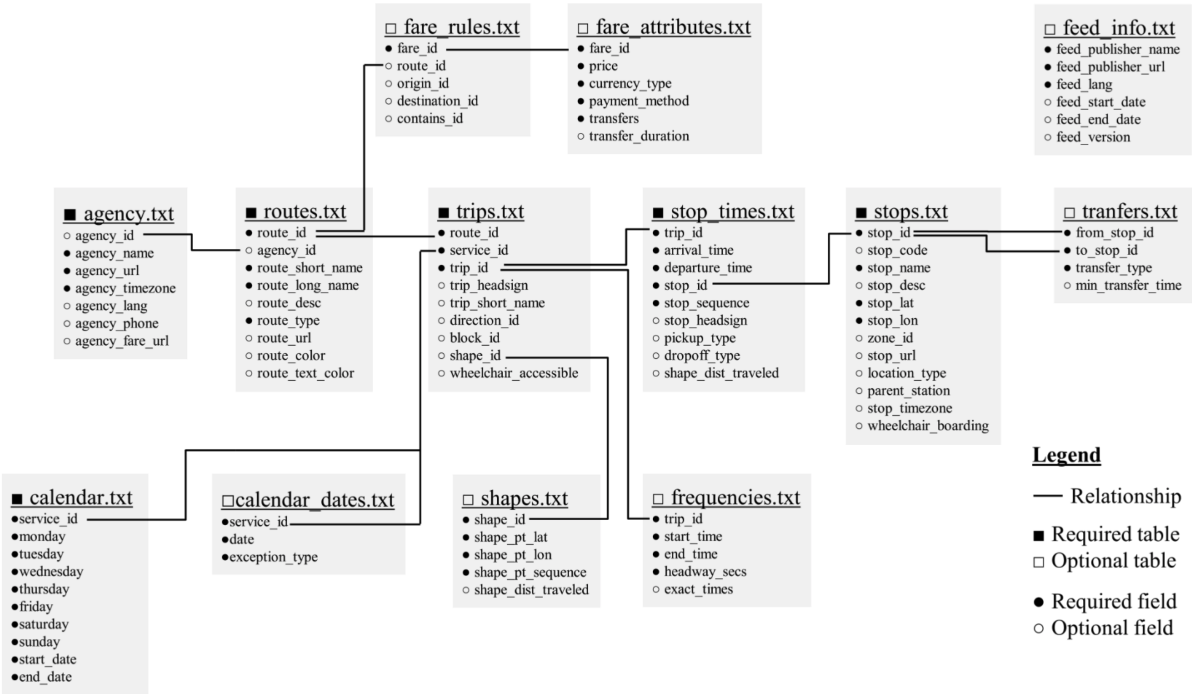

Figure 4.1: Example of a GTFS static feed specification and the relationship between entities, from [Won13].

model objects which will feed the system with uptime information as these buffers are small and simpler than other types of data formats, they allow for fast and efficient feed updates[Flu].

Despite the benefits of GTFS-realtime the adoption of the technology has been constrained by the expertise required for the task, complex overall system architectures and sometimes the lack of resources. Figure 4.2 shows an example of how data from transit vehicles is transformed into GTFS-realtime, to ultimately be displayed to the user. Starting in the transit vehicle, multiple data points are collected and shared through a modulator-demodulator, also called modem, to the transit agency's data centre. The internal system of the transit agency may have software for different tasks, like arrival time predication engines, Computer-Aided Dispatch / Automatic Vehicle Location (CAD/AVL) systems and GTFS-realtime transformers. The only connection between the internal complexity and the data consumer are GTFS-realtime feeds, that share trip updates, service alerts and vehicle positions. The Data consumers, like Google, Apple or other independent software developers, can draw this data from thousands of agencies and share it to users on any kind of software.

4.2.2 European standards

Most European transit agencies also choose to represent their transit data in compliance with the European Committee for Standardization (CEN)[3] Technical Standards for public transport information. Those technical standards include the European Standard "Public Transport Reference Data Model" (EN 12896)[4], also referred by its short name as Transmodel. The Transmodel standard defines a framework of data

[3]The European Committee for Standardization.
[4]Full specification of this standard can be found at `https://www.en-standard.eu/csn-en-12896-1-public-transport-reference-data-model-part-1-common-concepts/`

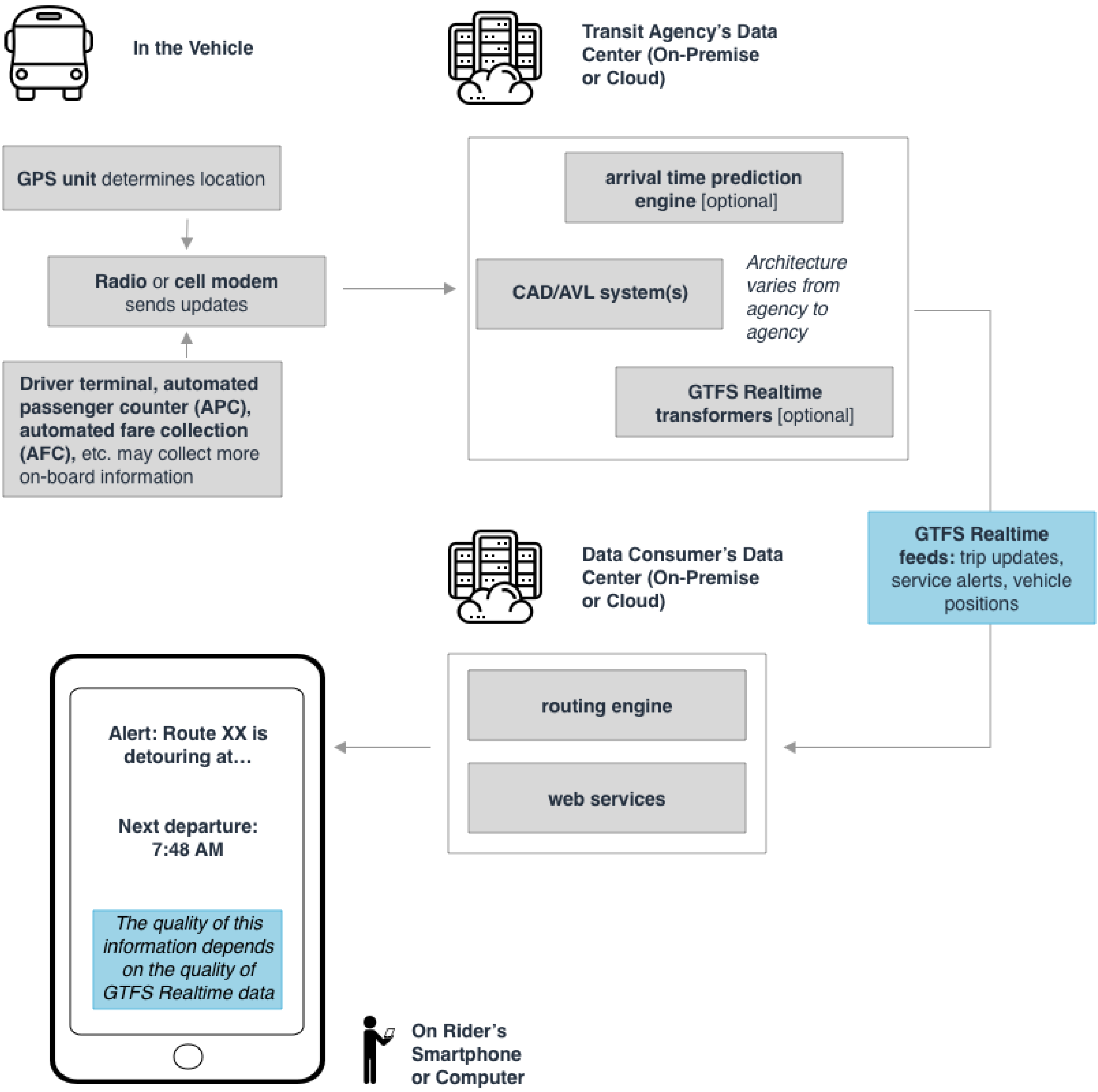

Figure 4.2: A diagram showing where GTFS-realtime feeds are typically produced and consumed (right) in an overall technical system architecture, from [DA20].

models for public transport information and service management that facilitate interoperability between the information processing systems of the transport operators and agencies by using matching definitions, structures, and meanings for their data [Traa]. It includes standards such as the NeTEx and the SIRI.

NeTEx

NeTEx is a CEN Technical Standard for exchanging Public Transport data, that covers public transport network topology, scheduled timetables, fare information and passenger information European profile. NeTEx is an XML format designed for the exchange of complex transport data among distributed systems that can be updated efficiently.

NeTEx and GTFS are complementary, while NeTEx is intended to be used in back office use cases under which data is generated and has all the additional elements used to construct timetables, GTFS primary propose is to provide information for route planning systems. Because NeTEx uses XML it can be packaged as a single coherent file that can be managed and validated, while GTFS uses a flat file format that requires multiple files to describe the different elements requiring specific tools to package files, interpret and process the data. It is possible to generate a full GTFS data set from NeTEx through the GTFS mapping package included in the NeTEx UML[5] but not vice versa.

GTFS covers stops, lines, timetabled journey information and simple types of fare information, NeTEx goes beyond that with a richer timetable model for the export of data that includes journey patterns, timing patterns, joined journeys, train makeup, connection timings and other information resulting in data sets that can be used to build new timetables and the ones from the export itself. NeTEx includes information needed for real-time systems (such as destination indicators[6]) and to connect to operational systems (such as for data storage). It also includes versioning and validation mechanisms to support the continuous integration of different data sources [NeT].

SIRI

SIRI is a European interface standard for exchanging real-time information about the planned, current or projected performance of public transit operations [Tec]. It covers planned and real-time timetable exchange, vehicle activity at stops, vehicle movement and information to aid in providing reliable connections between vehicles. GTFS-realtime services provide equivalent functions to certain SIRI services, it is possible to create adaptors to transform data from a feed in SIRI to a feed in GTFS-realtime however SIRI is designed for operational systems on the operator-to-operator level as it contains additional data elements not in GTFS-realtime feeds[NJS15].

As far as we know, both NeTEx and SIRI are not commonly used by the open-source development communities, since they are specifications specific to the European Union and not adopted in other regions. However, some open-source projects like OBA (has SIRI support) and OTP (extended to support NeTEx and SIRI) have recently started to support for data in those specifications.

[5] Unified Modelling Language is the modelling language intended to provide a standard visualization of a system's design.

[6] Destination indicators are signs mounted on the front, side, or rear of a public transport vehicle, that display route information.

TransXChange

Also based in the CEN standards, exists TransXChange, the UK nationwide standard for exchanging bus schedules and related data. TransXChange is used by transit agencies in other countries beside the UK[7] such as Spain. Although the bus schedules of the UK are available in the TransXChange format, they are often converted into GTFS by the open data users[Why].

4.2.3 Data Specifications Comparison

Data specifications translate complicated information into more outlined formats that transit agencies can share easily with users and private organizations. Independently of their size, transport agencies can use data specification to better communicate information to their customers. Data specifications provide the transit agency the ability to share information such regulations, routes, stops, and location of vehicles that can make the system more user-friendly and transparent, affecting positively the overall transport usage. The choice of specification will depend on a multitude of factors, such as the existing infrastructure, the maturity of the transit data to be shared, organizational policies and the financial implications.

The data specifications presented were GTFS (GTFS-ST and GTFS-RT), NeTEx, SIRI and TransXChange, but many more exist that are either proprietary or less adopted. Table 4.1 shows a brief comparison of different characteristics of those specifications. The GTFS specifications are largely designed to host transit information relevant for journey planning, and using data in this format for any other purpose can be difficult. As far as we assessed from research, out of all the presented specifications, feeds in the GTFS specifications are the most used and shared overall in the transport communities, therefore more resources, support, and a bigger developer community exist for them.

NeTEx has been systemized and validated against leading National Standards, granting compatibility with different pieces of software in transport systems that other standards do not. It covers most of the commonly found and advanced features of public transport information systems while being flexible, and allowing a modular implementation. Examples of advanced features not covered by other data standards are frequency based services and flexible services, a comprehensive fare data standard, that covers not only classic multimodal fares, but also modern fare models such as card based pay as you go[Proa].

NeTEx uses a sophisticated model that is undoubtedly broad and quite complex, it can impose a significant learning curve, but with knowledge of engineering notations such as UML, the core principles can be easily assimilated. The consistent terminology and a uniform set of design patterns of the model reduces the effort needed to understand the transport system and facilitates learning of new areas of functionality.

SIRI provides a similar service to GTFS-realtime, allowing distributed computers to exchange real-time information about public transport services and vehicles. SIRI is often used by transit agencies to exchange operational data with internal systems, while GTFS-realtime is designed for exchanging passenger and journey oriented data with external developers. The core high-level use cases that SIRI supports are much broader than GTFS-realtime, such as[Lib20b]:

- to provide “real-time departure from stop” information for display on stops, internet, and mobile delivery systems;
- to provide real-time progress information about individual vehicles;

[7] Details of how different data standards are implemented in different European countries can be found at http://www.transmodel-cen.eu/category/implementations/

- to manage the movement of buses roaming between areas covered by different servers;
- to exchange planned and real-time timetable updates;
- to distribute status messages about the operation of the services; and
- to provide performance information to operational history and other management systems.

Using a standardized common interface like SIRI permits that real-time data from different areas to be linked up seamlessly both within, and across borders of transport systems.

TransXChange covers a functional equivalent subset of the NeTEx stop and timetable elements, both deal with the exchange of structural information such as timing information and journey patterns. The NeTEx representation is, in some regards, simpler, but it is also more flexible and less strict about which elements must be present, and what the restrictions of values are. TransXChange also includes some additional registration elements to support the Electronic Bus Service Registration process of the UK Vehicle Licensing authority (Vehicle and Operator Services Agency (VOSA)). It is possible to add a separate UK extension to NeTEx to include these.

	Public Transit Data Specifications		
Characteristics	GTFS-ST and GTFS-RT	NeTEx and SIRI	TransXChange
Place of origin	North America	Europe	United Kingdom
Where it is used	World-wide	Europe, some cities in North America, Asia and Australia	United Kingdom and some countries in Europe
Data type	Static (GTFS-ST) and Real-time (GTFS-RT)	Static (NeTEx) and Dynamic (SIRI)	Static
Data format	CVS flat file format	W3C XML Schema	W3C XML Schema
Sponsor Organization	Google	European Committee for Standardization	UK Department of Transport
Functional Scope	For GTFS-ST: Use case – data for journey planners Simple , flattened model o Stops & Timetables. o Fare prices.	For NeTEx: Multiple use cases (profiles) Rich model, reusable components o Network, Equipment o Stops & Timetables, o Operational data, Timings, Equipment o Fares, Products, Fare Prices o Accessibility, Train makeup, etc...	Use case — exchange of structural information such as timing information and journey patterns Rich model, but more restrict elements than NeTEx o Covers very similar information to NeTEx, however, lacks some of its elements

Table 4.1: Comparison between the most relevant Public Transit Data Specifications

Most large transit agencies share data in the formats of multiple data specifications in order to reach the most broad range of use cases possible. However, there can be some potential downsides to the use of specific specifications. Developing and translating already existing transit data to a data specification can be a long process and since specifications often overlap in their purposes, it can be a drain of resources if not successful[Thea]. Besides this, individuals and communities will have a limited benefit from solutions developed using transport data if they lack access to specific mobility options, infrastructure, or technologies like smartphones, as for example, elderly, disabled or young users of public transport.

4.3 Mobile Development Methods

Mobile applications are a critical part of many businesses, they have quickly grown in complexity since their beginnings to improve and meet industry standards demanding evermore short delivering times. Mobile development technologies can be divided as native, hybrid, cross-platform and multiplatform.

Native development focus on developing for only one target platform using platform specific programming languages, software development kits, development environments and other tools provided by the Operating

	Development methods of mobile apps		
Parameter	Native Development	Hybrid Development	Cross-Platform Development
Cost	High cost of development	Low cost of development, code can be reused from a already existing website	Relatively low cost of development
Code Usability	Works on only one target platform	Code can run on multiple platforms	Code can run on multiple platforms
Device Access	Allow for the use of the capacities of the platform in full	Difficulty or impossibility to use all the capacities of the platform	No assured access to all device APIs
Quality of the User Interface	Consistent with the UI components of the device. Seamless experience	Limited consistency with the UI components of the device	It is possible to achieve consistency with the UI components of the device
Performance	Seamless performance, given the app is developed for the devices OS	Dependent on browser's speed, can have a slower performance and transition between pages	High on performance, but hardware compatibility issues are not uncommon

Table 4.2: Differences between native, hybrid and cross-platform app development

System (OS) vendor.

While with both hybrid, cross-platform development the same code is reused to work on multiple OS's the approaches are fundamentally different.

Hybrid development uses WebView — a browser engine without any visible toolbar or status bar that when used by a native application can display web content — WebView apps are created using web technologies like HTML5, CSS, and JavaScript and commonly deployed as business assets alongside with business websites. Because every OS displays each website the same, hybrid applications will run comparably on all devices.

With a cross-platform development framework, it is possible to reuse all or part of the code for multiple target platforms. Frameworks for cross-platform development provide a set of tools that act as a proxy to native OS features through a set of libraries created using native programming languages for each platform with one common Software Development Kit (SDK). Common code from cross-platform frameworks are converted into platform specific components and provide native device experience at run time.

Kotlin Multiplatform Mobile introduced a new paradigm in cross-platform development, which allows to adapt the Kotlin language for both iOS and Android platforms. Kotlin is already widely used for Android development as a replacement to Java, but with Kotlin Multiplatform Mobile it is possible to share the business logic of an application to different platforms without introducing new languages to the codebase. As seen in Table 4.2 there are benefits and drawbacks in all the methods, therefore the choice of the development method should be specific to the project and context of the business.

4.4 An Overview of Mobile Development Frameworks and Tools

This section covers the most popular mobile development frameworks and tools.

Flutter

Flutter is an open-source cross-platform UI toolkit created by Google. Flutter code can be reused to create applications for multiple target platforms such as Android, iOS, Linux, Mac, and Windows.

Flutter applications are built in the Dart platform and programming language, which was developed with a focus on fast development cycles while still resulting in high-quality applications[Flu].

Code written in Flutter runs in the Dart virtual machine that allows for Just-In-Time (JIT) execution,

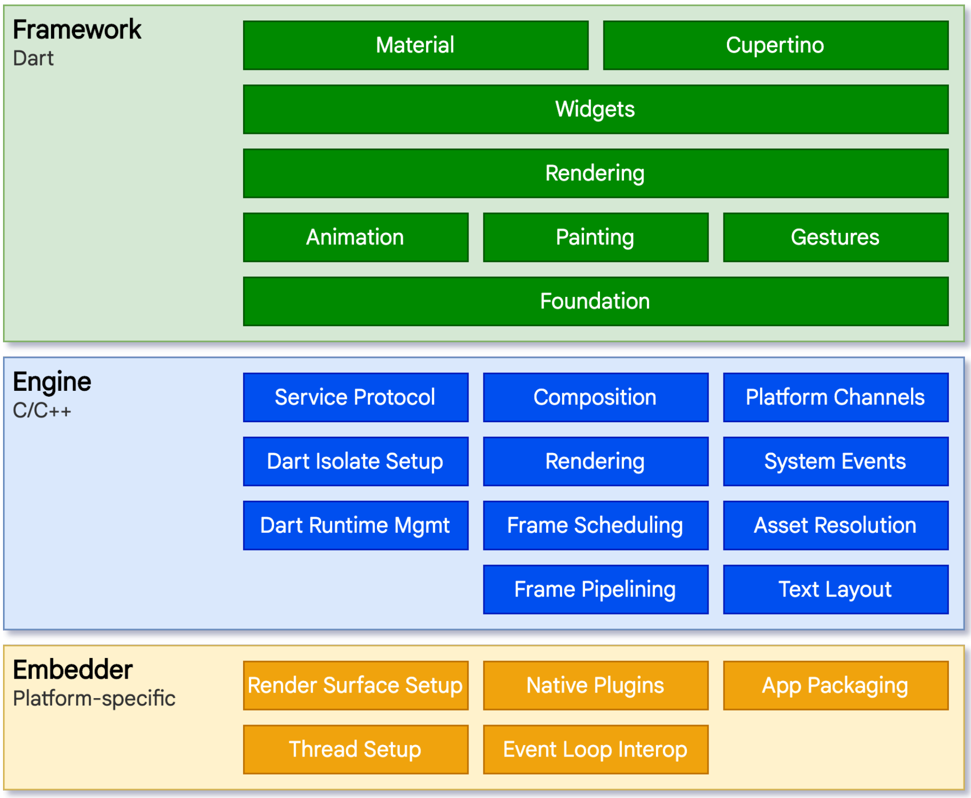

Figure 4.3: Flutter Architectural layers[Flu]

where the program is compiled during its execution with access to run time information that enables better optimization. The JIT compiler unlocks many benefits for development, like incremental recompilation, where modifications to the source code are immediately reflected in the running application, live metrics collections and rich debugging support[Flu]. Flutter works as a layered system where every layer is optional and replaceable[Flu].

At the base of the system that underlays a Flutter application is a platform-specific embedder that provides an endpoint to the target platform and coordinates access to services like rendering surfaces, native plugins, app packaging, thread setup, accessibility, input and manages the event loop. With the embedder, Flutter can be integrated as a module into an existing application, or it can be the entire content of the application.

The Flutter engine at the centre of the Flutter system deals with the rendering of composed scenes and supports the primitives necessary for all Flutter applications through a low-level implementation of Flutter's core API.

The Framework layer of the Flutter accesses the engine through the dart:ui, the library that exposes the lowest-level services that the Flutter framework uses to transform the underlying C++ code into Dart classes, such as driving input, graphics, text, layout, and rendering subsystems.

The Framework is the point of interaction with the system for developers, it includes all the necessary libraries to build an application.

Other Cross-platform options

Other cross-platform frameworks relevant in this space are Reactive Native and Ionic.

Reactive Native is an open-source framework created by Facebook in 2015, where applications are written using the JavaScript programming language and can be compiled into all the major target-platforms. React Native runs on the JavaScript run time environment and uses its own user interface abstraction layer over the one of the target platform. It includes native components that cover all the native UI elements and can be integrated with JavaScript libraries for other elements of the native user experience like tab navigation. Because of this, Reactive Native is an easy entry point into mobile development for anyone with previous experience in web development.

Ionic is an open-source SDK that uses web technologies like HTML, CSS, and JavaScript to build the frontend User Experience (UX) and UI of the application while being integrated with other development frameworks, such as such as Angular, React, or Vue. Ionic is able to be compatible with all kinds of JavaScript frameworks because it is made of Web Components[8] that also can run standalone in a web page with just an HTML script tag. The integration with different frameworks make Ionic a good option if an application already exist in one of the supported frameworks and the goal is to transform it for the mobile platforms.

Ultimately, the choice of framework and tools used in a project will mostly come to the programming language preferred by the developer and the details of the project.

[8] A suite of different technologies used to create custom reusable elements.

5 Public Transport Mobile App for the City of Évora

This chapter describes the motivation and contexts for a public transport mobile app for the city of Évora, Portugal. The requirements and functionalities were identified and the front-end and back-end components were designed and implemented to fulfil those. Lastly, the results are evaluated and discussed.

5.1 Introduction

The goal of this dissertation is to create a solution to share local public transport information with passengers of the city of Évora. By building upon the research presented in the previous chapters, a mobile application was conceptualized, designed and implemented to fulfil this goal. The application was designed with a focus on meeting the requirements raised, and to contribute positively to the use of public transport in the city.

Motivation

Évora is a city in the south-central region of Portugal with strong cultural and religious tourism, as a UNESCO World Heritage Centre. With a population of 56,596 (2011) Évora can be considered a mid-sized city. The public transport of the city of Évora is composed of 10 bus lines that run on schedule and 1 line that runs on a flexible schedule from 09:00 AM to 17:00 PM every 30 minutes leaving the terminal. The scheduled lines 21, 22, 23, 24, 25, 31, 32, 34 and 41 cover the main metropolitan and suburban areas of Évora with convenient times to serve schools, hospitals, and other public services. The flexible schedule line *"Linha Azul"* covers the historic and tourist centre of Évora, with stops in strategic tourist spots.

At the moment there is not a real-time information system in present in the public transport of Évora and because of the small scale of the transport system in question, it is a perfect candidate for a case study in this area. In comparison to other components of a passenger information system, such as information in displays at stops and stations or through a website, a mobile application has the possibility to deliver information in different kinds of contexts efficiently.

Unfortunately, after contacts with the city council a partnership for the use of real information from the buses running, it was concluded that it will not be possible at this time. However, looking forward to the future, the work developed in this thesis can still be used in the transport system when the conditions to do so allow.

The motivation remains to create a solution to inform passengers of the state of the public transport, by implementing a mobile application that will meet this goal.

5.2 Requirements

As discussed, the app is intended to share public transport information to passengers. The usage of the app can cross multiple contexts and use cases. For regular transport users, it can provide them reassurance about their recurrent journeys. An occasional user, may still be exploring the public transport network and need to discover new routes and connections. A new user, such as a tourist, will not know anything about the public transport network and will need complete directions and advice to navigate the transport network. But it would not be unlikely that a single passenger would need to use the application in all the contexts mentioned before. Consequently, observations were made on existing transport apps for other public transport networks, the full analysis can be found in Chapter 2.2. As a result of such comparison, an assessment was made to conceptualize requirements into "baseline functions" and "interesting functions" to limit the development to specific functions.

5.2.1 Functional Requirements

The main features/goals of the app are :

1. The mobile application must show a real-time (or perceived real-time) view of when the transport will reach its destination;

2. The mobile application must allow to see the route of the transport in a chronological order while still showing a temporal value for every stop in the route;

3. The mobile application must show the interceptions between routes at each stop;

4. The navigation between the views of each stop in the mobile application, must be sequential and chronological, meaning that you can travel between views of stops and see results based on the time and route selected previously;

5. The mobile application must give the ability to add stops to a favourites list.

5.2.2 Nonfunctional Requirements

The nonfunctional requirements define system attributes that can be used to access the operation of a system, rather than specific behaviours. The nonfunctional requirements are the following:

- The application should be developed into one single self-contained app;
- The application should be connected to a back-end system to handle user requests;
- Usability: The user interface should be easy for new users to use, even without prior knowledge of the transport network or other public transport applications;
- Flexibility: The app is intended to be able to adapt to a change in requirements or the addition of new features;
- Reliability: The app should have error messages when operations are not usefully;
- Accessibility: The application should be accessible to those with disabilities.

5.2.3 Functionality to be accomplished

The observation and analysis of possible requirements is necessary to understanding the functionality to be accomplished. This process translates the original ideas into objectives, functional requirements, non-functional requirements, and business rules. The constraints of the app are also clarified by this analysis. According to the requirements described, there are several functions to be accomplished. First, the user can see a list of all stops, then he will select one and a request will be sent to the back-end system. The request will return all the routes that operate on the stop and the times for when they will reach it. The response will be translated to a view, where the user will see the information in the interface. The user then can select one of the routes and a request will be made that will return the route details. The route details will consist of the next stops after the one selected initially and the times at which the route will reach the destinations. Those are the most basic functions to be accomplished, the application will be flexible enough to allow for others, such as adding favourites, seeing maps and get transit information, with the possibility for more.

5.3 Design and Implementation

The background analysis and requirement gathering mentioned above, enabled the understanding of user requirements for the public transport information app for the city of Évora. Those instruments and hands-on experience of using the public transport system can help to make the application as user-friendly, useful and fit for purpose as possible. This information was fundamental when developing the prototype of this application and aligning the user interface design. Meanwhile, based on the requirements and functionalities expected, the business logic was designed to ensure that the development would remain in line with those. Consequently, the user interface was continuously optimized in order for the app to remain consistent.

5.3.1 User Interface Design

To set up the basis of the user interface, it was necessary to clearly establish the user requirements, the user can broadly be fitted into the following roles:

1. *Casual User*: The user can view all the stops in the network and then in another view, the routes that travel through the chosen stop. The user can also know when that route will reach the stop. Then the user can see in another view the route journey;

2. *Regular User*: The user can add his favourite stops to a list for easy access.

Although more user roles can be found in transport applications of this kind, with this clarification of the basic user expectation, the user interface design as conceptualized. The figure 5.1 shows the first hand sketched wireframe, and it is evolution into medium-fidelity wireframes that have more detail — including accurate spacing, headlines, and buttons. The figure 5.2 shows the wireflow design that clarifies the navigation in the application. The high fidelity wireframe step was omitted from the design process to allow for flexibility in the front-end implementation.

In the figures 5.3, 5.4, 5.5 and 5.6 it is possible to see in detail idealized user journeys for different scenarios. In 5.3 the user starts in the Stops page, where it is possible to see all the available stops. On the right to the stop name there will be the initials of the routes that pass through that stop (I). When a stop is selected, the interface will show the stops page where the list of routes that pass through the selected stop are displayed sorted in chronological order, with the minutes until that route reaches the stop (II). As the user locates and clicks on the desired route, selecting it will lead to the Route page where the next stops of the selected route will be displayed in chronological order (III). This journey allows the user to determine information that is helpful in the navigation and usage of the transport system, such as how many stops the Route has until the selected Stop and how many transport vehicles will pass through the selected Stop until the one from the Route reaches the Stop.

In the figure 5.4, the user starts by selecting a specific Route and navigates until they know when that route reaches a specific Stop. In a particular practical scenario, this can happen when a user uses the transport system regularly and knows that a specific route will reach a specific Stop, therefore only needing the to know when it will happen or information for a more complex navigation pattern that may include multiple routes and stops. Navigating from the Routes page, the user can see on the right side of the routes list the intercepting routes of each route (I). Selecting a route will lead to the Route page where the next stops of the Route are displayed in chronological order (II). From this point, selecting one stop in the Route page it will lead to that Stop page, where the next routes to reach the Stop are displayed in chronological order, with the information of how many minutes until it reaches the stop and the direction that the route is travelling (III).

The favourites' page, in figure 5.5, allows for easy navigation to specific information for the user. In this page, shortcuts to the routes and stops favoured by the user are displayed (I). The shortcuts to the favoured Routes will lead to the Route page where the next stops of the route are displayed in chronological order (II). The shortcuts to the favoured Stops will lead to the Stop page where the next routes to reach the Stop are displayed in chronological order (III), same as the previous user journey.

The figure 5.6, exemplifies a user journey starting at the Maps Screen, where the location of stops is showed on a map with the relevant information about them displayed below it (I). Selecting a stop on the map will lead to the stop page with the information the of routes that service that stops displayed in chronological order (II). From there, a selected route journey can be followed in the route page (III).

Figure 5.1: Evolution from hand sketched wireframe to medium-fidelity wireframes.

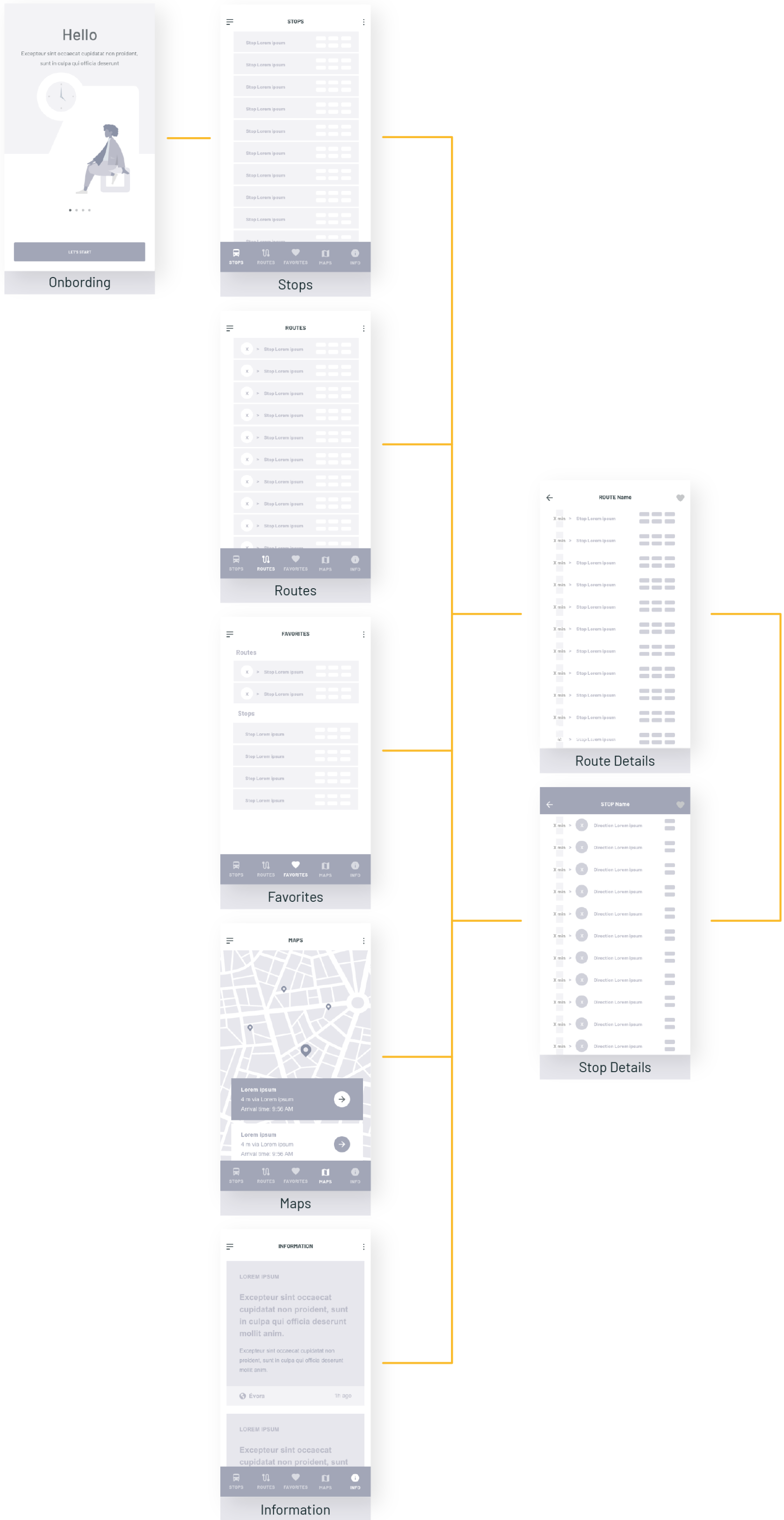

Figure 5.2: Transport App Navigation Wireflow

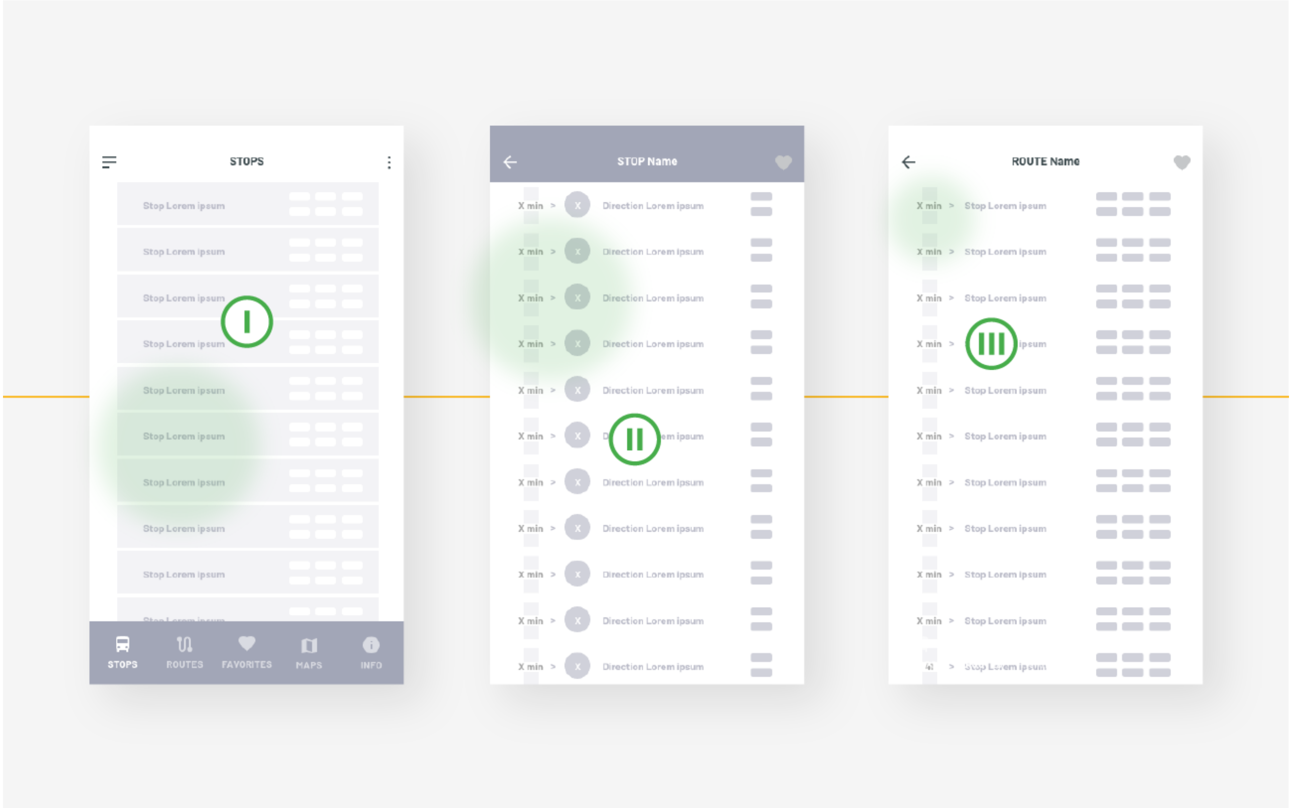

Figure 5.3: User journey to discover the route progress starting at a selected stop.

The user journeys exemplified show how information can be displayed and organized in distinct ways to fulfil the need of different kinds of users. While the information is the same, the delivery of it changes in all user journeys, allowing users to contextualize the information.

This conceptualization of the user interface is an important step towards the overall goal of a transport information application for the specific purpose of servicing the public transport of Évora. Without this conceptualization, it would not be possible to build any solution for this use case. As far as we know, at the moment, no solution exists that would permit users to navigate and gather the transport information available in the way that it is commonly expected using a technological tool.

The resources used to improve and influenced the UI design include the Google Material Design[1], projects in the Behance website[2], open source widgets from the Flutter community, and the visual identity of the city of Évora[3].

[1]Documentation of the Google Material Design can be found at `https://material.io/design`.

[2]The Behance semantic panel made for this project can be found at `https://www.behance.net/collection/187298721/Transport`.

[3]The visual identity for the city of Évora can be found at `https://www.cm-evora.pt/en/visitante/agenda-e-noticias/media-center/identidade-visual-do-municipio/`

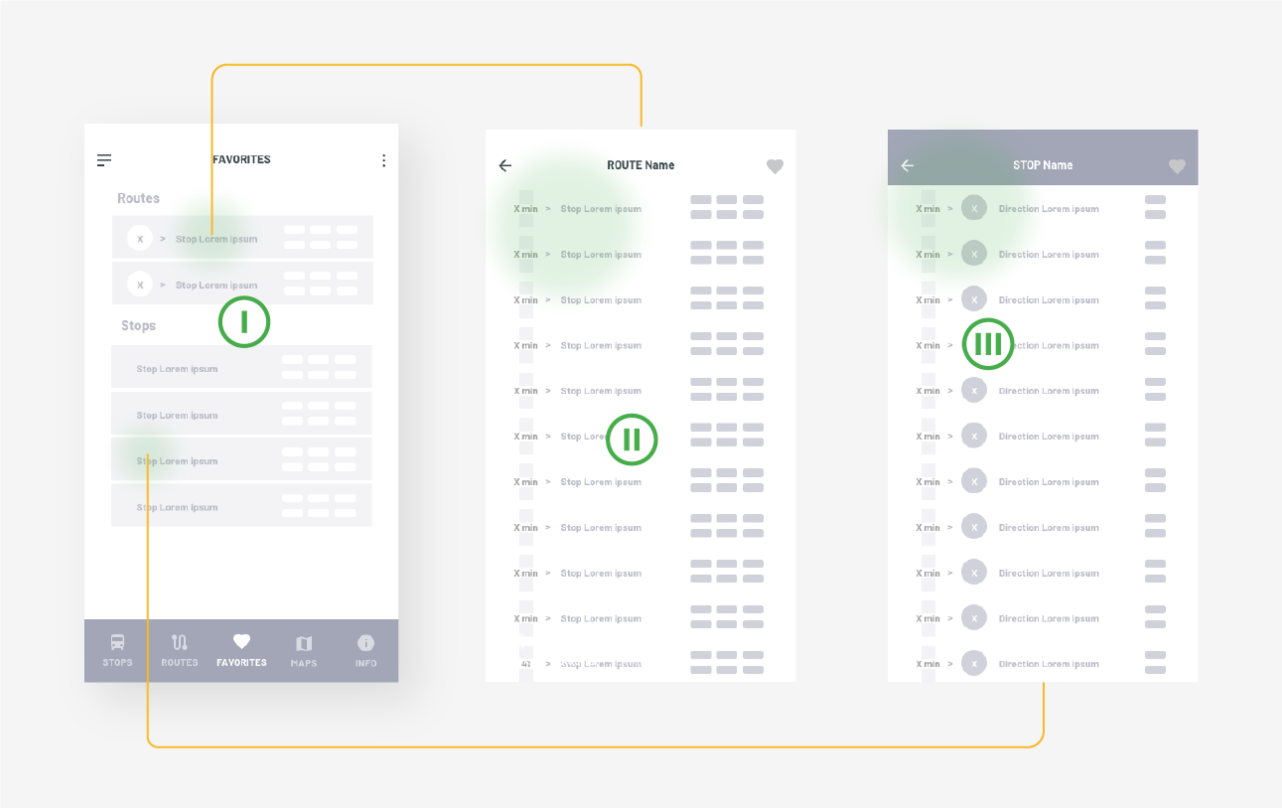

Figure 5.4: User journey to discover when a specific route will reach a specific stop.

5.3.2 Implementation

Back-End Implementation

As far as back-end implementation, both the OBA and OTP software were considered as possible suitable solutions since they include components that can perform the necessary functionalities. The primary function of OBA is to share real-time public transit information with riders across a variety of interfaces, for that purpose it includes software modules for many types of relevant applications (more details about all the features of OBA can be found in 2.1). While OTP is primarily a multi-modal trip planner that exposes REST and GraphQL APIs that can be accessed by clients through HTTP requests. The representation of the transport network can be built from GTFS, NeTEx and SIRI and OpenStreetMap data. It supports real-time updates of disruptions and service changes for itineraries when the real-time data source is used.

The Transport Data As previously established, the transit data is an essential element of any public transport application. As of the time of the writing, the transport data for the city of Évora is unstructured, in the form of images of the timetables[4].

The OBA project, includes a Java library for reading and writing GTFS feeds, including database support,

[4]At the end stages of writing this thesis, it was noticed that Google Maps has access to the transport data of the city of Évora. The dataset being used by Googles was not found in any of the common transport dataset aggregators, leading to believe that it was exclusively shared with Google. A future follow-up contact with the transport agency for the city of Évora in the hopes of getting access to this dataset would be ideal. This does not negate the usefulness of the solution idealized here, and if a complete and accurate dataset is gathered, the implementation of the solution would be better fit for purpose.

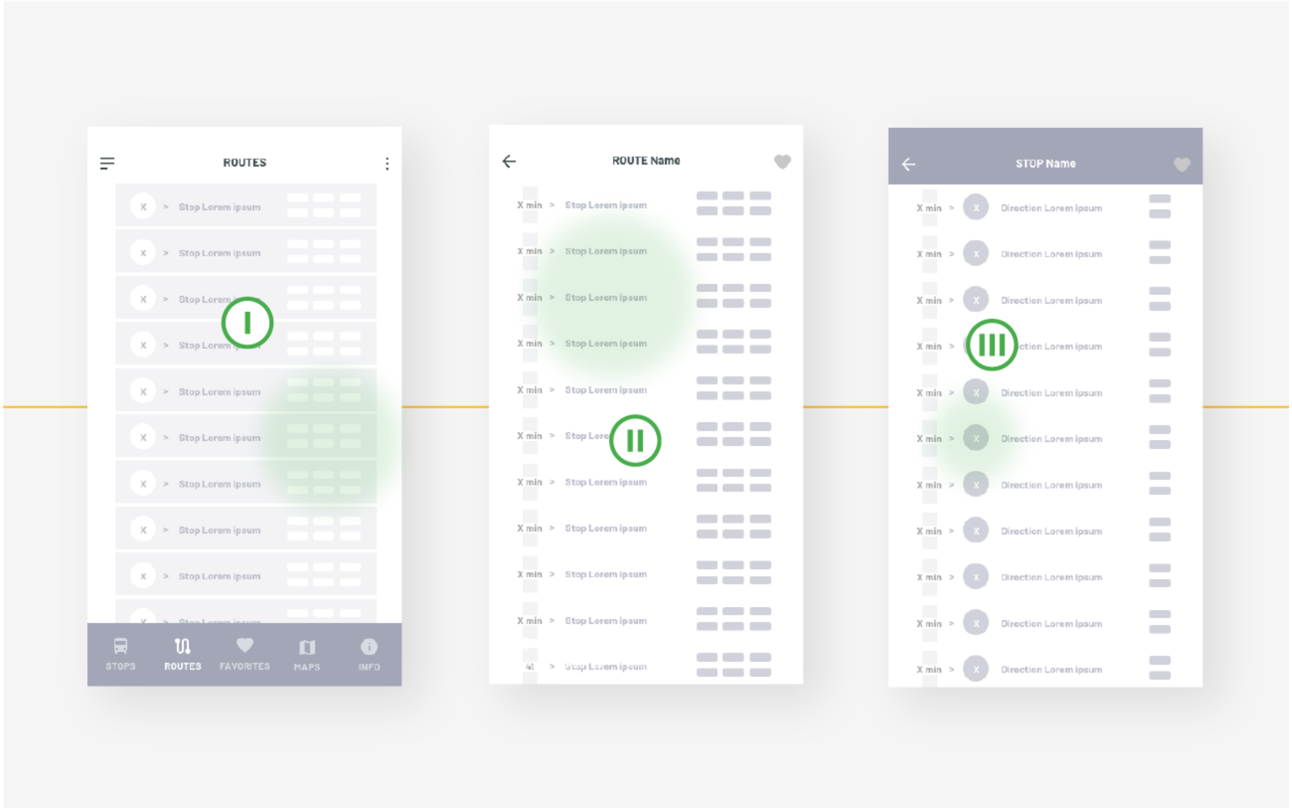

Figure 5.5: User journey from the favourites' page to discover when the routes and stops favoured by the user will reach the destination.

called *onebusaway-gtfs-modules*[5]. With this library, it is possible to read and write GTFS data as Java objects, and so, translate the timetables into Java objects and export them with the correct formatting into the appropriate GTFS-static files.

For easier translation into GTFS, the schedule should be structured into a table composed on the first row with the stops, then a row per each stop feature like wheelchair accessibility (the stops features are optional as described in the GTFS documentation). There must also be a number on the first cell of the document specifying how many lines the table header takes for the algorithm to know when the actual timetable starts without parsing the text.

4	Bacelo	Horta das Figueiras	Lagoa	Norte Sul
Wheelchair	N	N	Y	Y
Connections	N	Y	Y	N
Bike Access	Y	Y	Y	N
1	07:30	08:15	09:10	10:10
2	09:20	10:20	11:20	12:30
3	10:30	10:45	11:30	11:45

Table 5.1: Example timetable for a route

The algorithm starts by reading the first line and checking on the dataset for the existence of each stop, if

[5]The *onebusaway-gtfs-modules* repository can be found at `http://developer.onebusaway.org/modules/onebusaway-gtfs-modules/current/index.html`

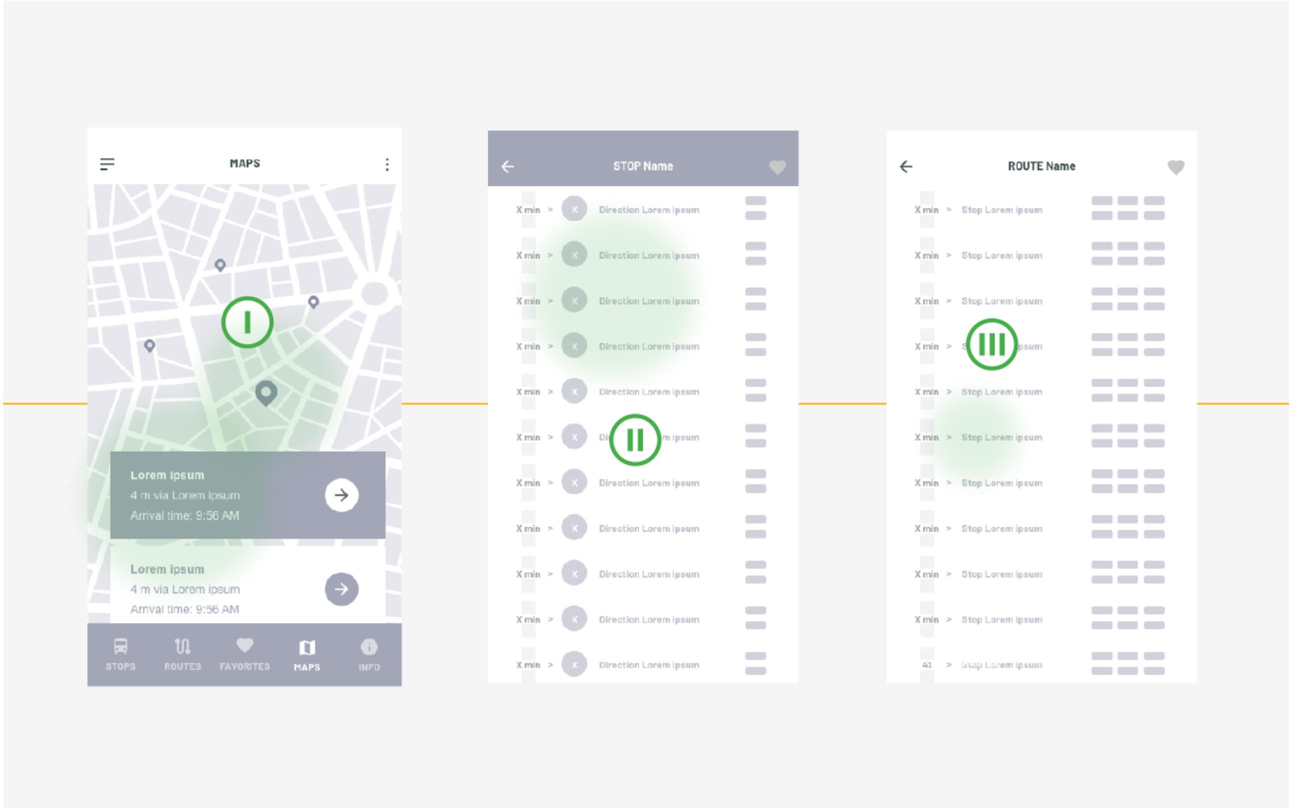

Figure 5.6: User journey from the Maps Screen to when a route will reach a specific destination.

it does not exist it creates the stop. Then it checks the next feature lines to fill the stop, ideally this would include latitude and longitude for extra information when using the OTP system. After adding the stops, the timetable is read, cell by cell, creating a new entry on the stop_times document calculating the index of the stop in the current route in order to fill the sequence diagram as well as the trips document. If this trip is not available on the routes document, it is also created there. With all the files created it is simply exported using the *onebusaway-gtfs-modules* library in order to be used in the OTP system which will let the system function with all the features required.

For the purposes of this dissertation the OTP was left without any modification and its API was explored using some testing tools and the browser. Based on the interface, the network calls used by the OTP default application were checked and used to build our own calls to retrieve the JSON formatted response required to use on our application.

The figures 5.7, 5.8, 5.9, 5.10, 5.10, are examples of OTP API calls that are necessary to display the data planned in the user interface of our application.

Outside the scope of this solution, with OTP it is also possible to perform API calls to explore the routes given the latitude and longitude of both the starting and ending point to get an itinerary. These API calls run fast enough to be used on the use case of this dissertation in personal hardware. As most of the pages in the APP only require one OTP API request it is near instant to display the requested changes.

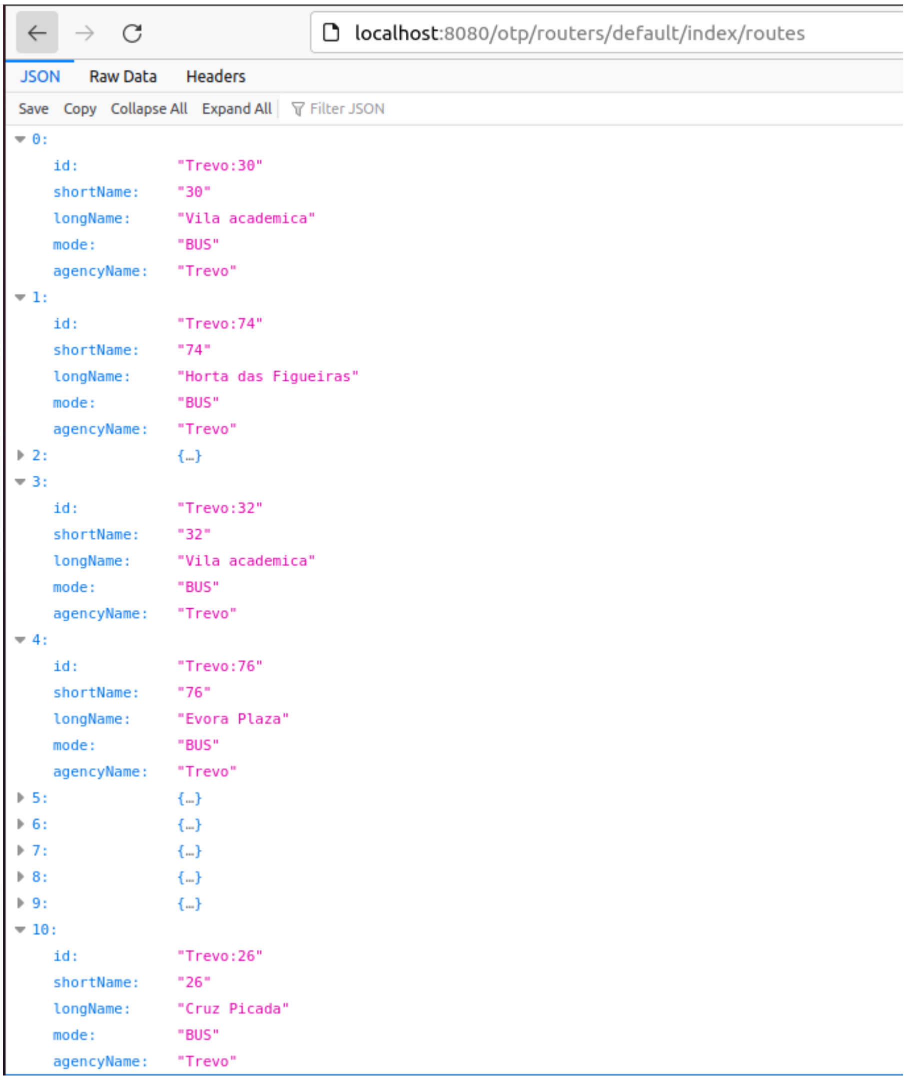

Figure 5.7: Example OTP API request that returns all routes in the system.

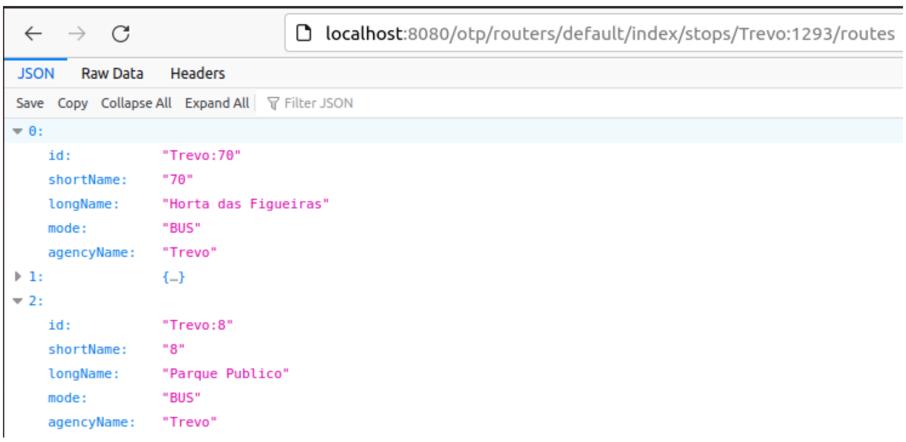

Figure 5.8: Example OTP API request that returns all routes that service a stop.

Front-End Implementation

The application is developed within Android Studio using the Flutter framework. The Flutter framework was chosen with consideration of its main features, such as:

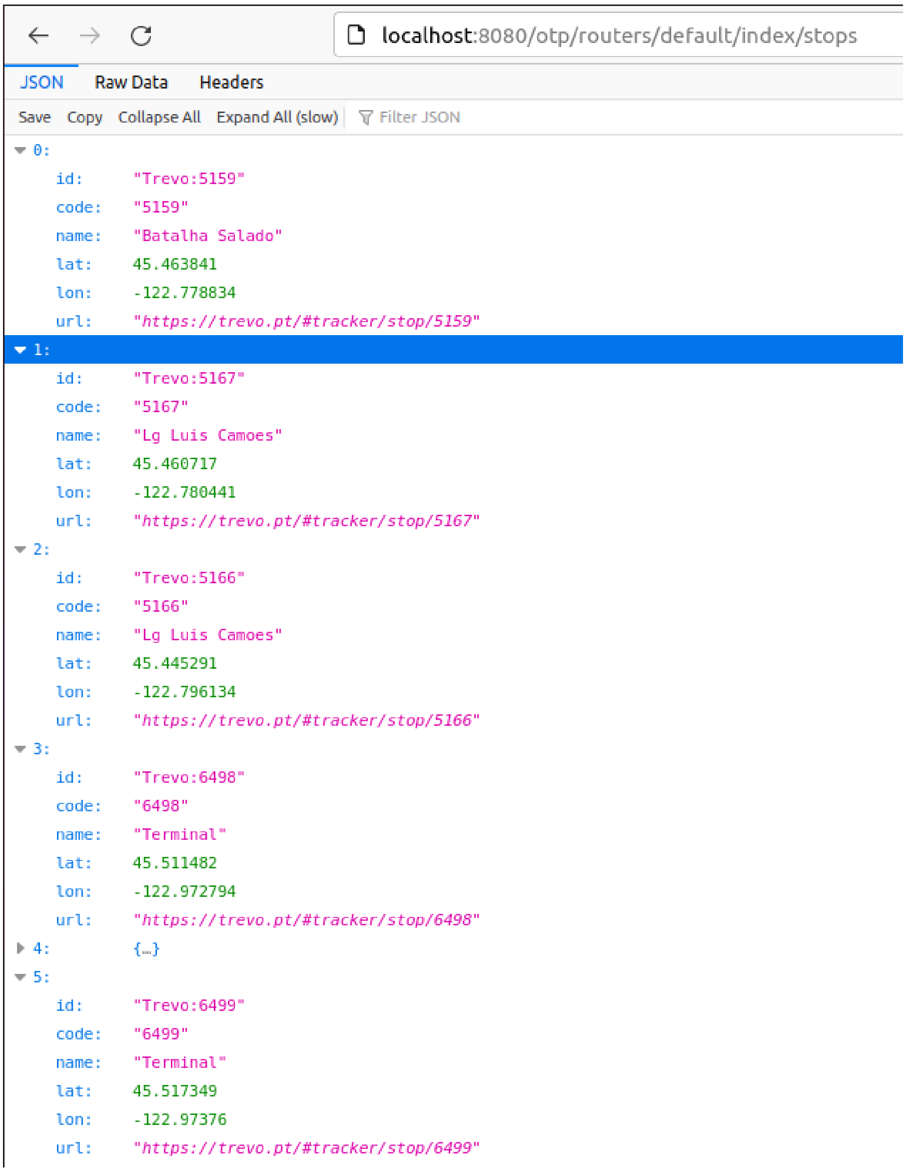

Figure 5.9: Example OTP API request that returns all stops in the system.

- Faster development speed compared to other frameworks;
- The same application UI and logic for different platforms, will allow for further development in the future into more platforms;
- Close to native application performance;
- No reliance on platform specific UI components;
- Close to no limitations to UI customization.

The comparison with other development frameworks can be found in Section 4.4. Building the UI is the first front-end related concern, by setting up the navigation first the work needed to be done was divided into clear blocks defined by the pages of the app. Then UI was developed one single interface element at a time, such as the stop information element and the route information element. After the UI elements were

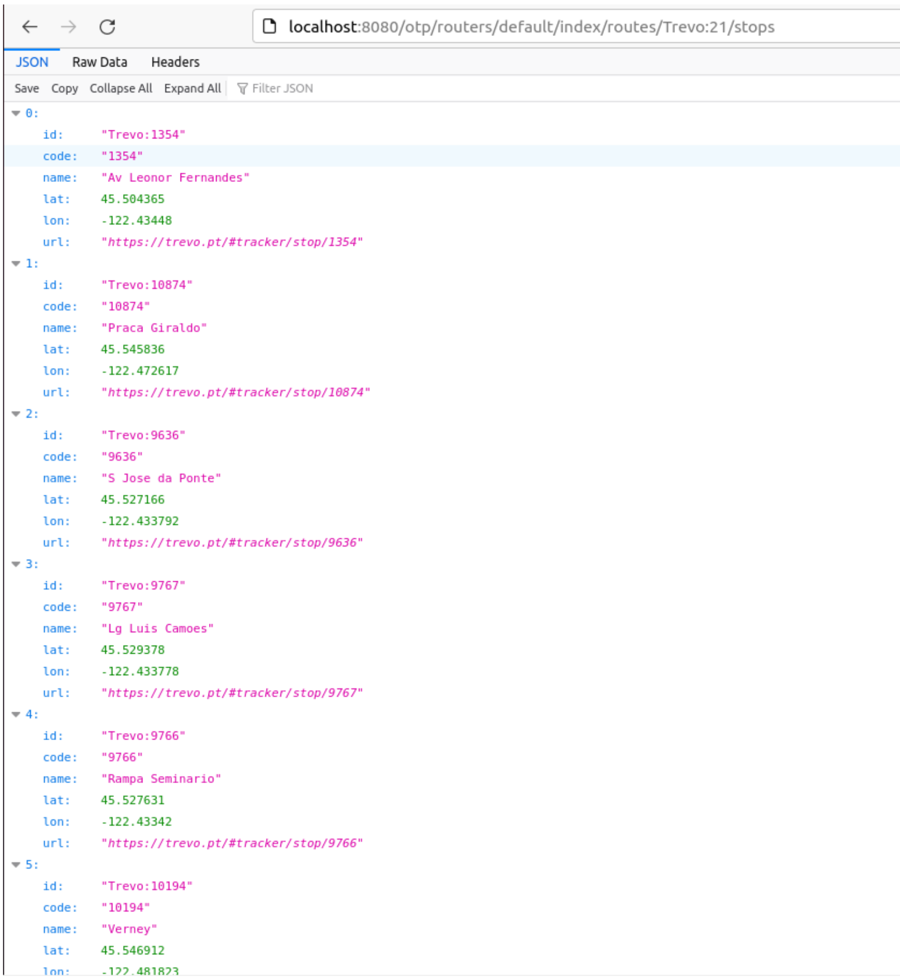

Figure 5.10: Example OTP API request that returns all stops that a route services.

developed, network requests are performed to feed the interface with data. In Flutter network requests are made using the built-in *http* package, when the response is received the response body is converted into a JSON *Map* with the *dart:convert* package to be displayed in custom Dart objects.

The figures 5.12, and 5.13, show the prototype app while running locally on Android Studio. In figure 5.12, the list of all stops returned by the OTP API is displayed on the stops screen. In figure 5.13, the list of all routes returned by the OTP API is displayed on the routes screen. Because the front-end application and the back-end system are both running locally, there were no problems in running the network requests, however in a real-world scenario they world be separate and possible network issues should be analysed. At this stage, the data returned was not analysed for accuracy and because of this it should not be considered as reliable. At a later stage, it would be necessary to validate the displayed data to ensure the usefulness of the solution. The other screens of the prototype not included here are still in development, the screens pictured stand as a proof of concept that the methods and technologies chosen are feasible and applicable.

5.4 Results and Evaluation of the solution

After the design and implementation, the proposed application fulfils the baseline functions — for example, the user can view all the stops in the network and, in another view, the routes that travel through the

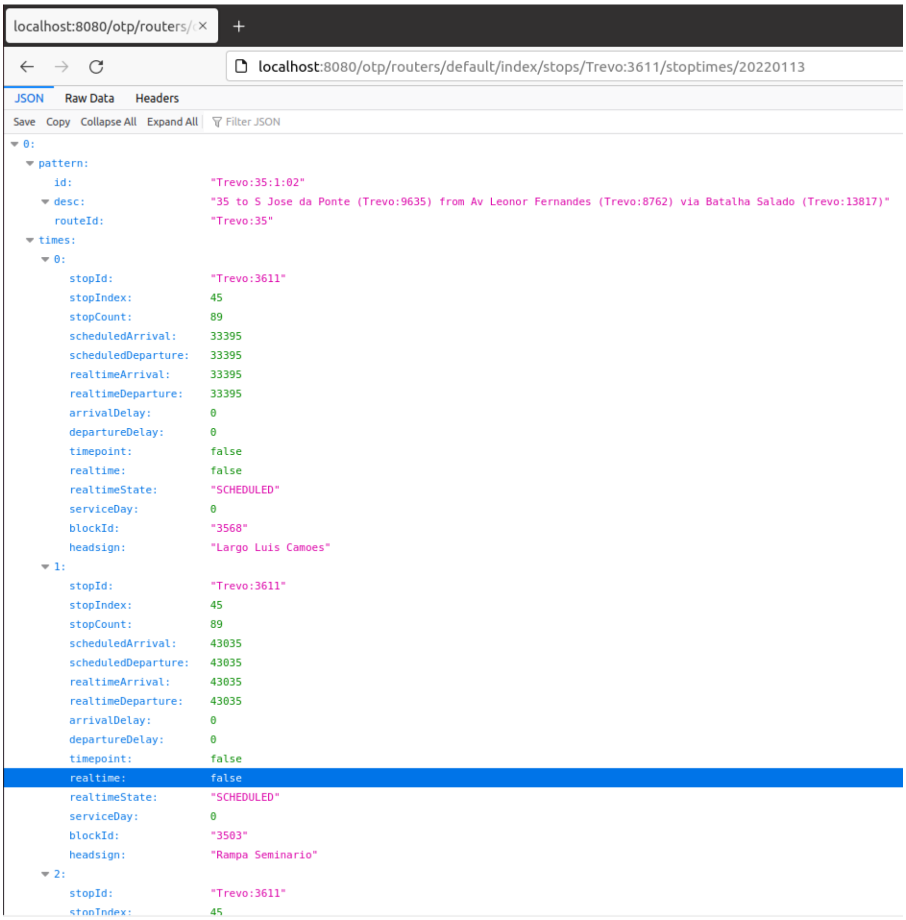

Figure 5.11: Example OTP API request that returns the arrival time of a Route to different stops, given a starting date.

chosen stop. At present, the application was only used locally, but ideally in the real-world deployment of the application, a specialized server based on OTP would be created with an accurate dataset to run the first real-world tests. Since obtaining this data is subject to approval of the organizing entities, anecdotal transport data was used instead to get the conclusions of the feasibility of the project. The Android application was implemented with layouts that are user-friendly from the beginning, and it sends HTTP requests and receives responses from the server. The solution can be evaluated by the requirements previously established, such as:

- Usability: The user interface utilizes simple elements that can be easily used by someone without prior knowledge of the transport network or other public transport applications;
- Flexibility: The Flutter framework used and interface choices allow for easy changes or adding of new features;
- Reliability: The display of error messages was taken into consideration during development;
- Accessibility: The accessibility was taken in consideration by applying the accessibility release checklist that can be found in Section 3.1.

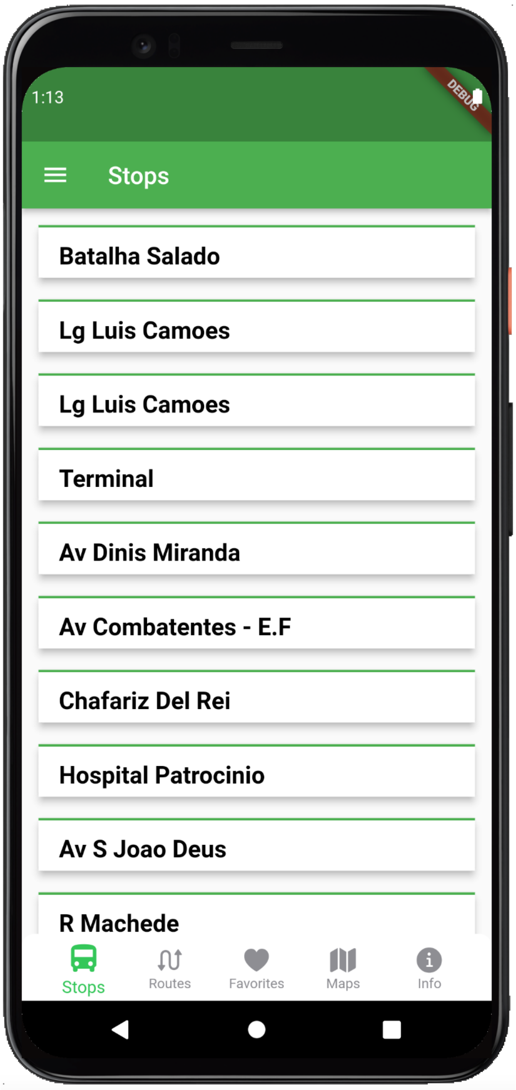

Figure 5.12: Image of the prototype application at the stops screen.

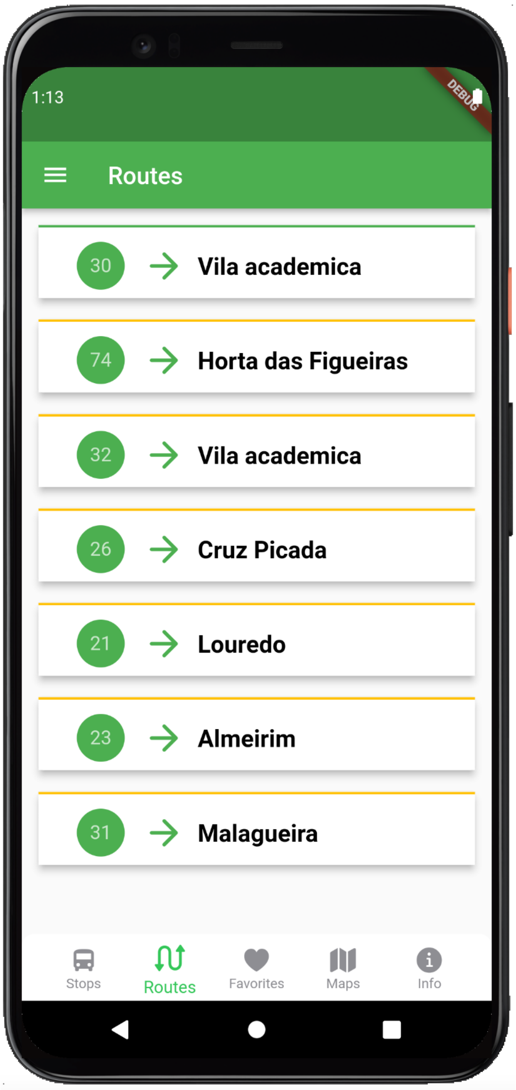

Figure 5.13: Image of the prototype application at the routes screen.

6

Conclusion and Remaining Considerations

This chapter shows some final remarks about the conclusions reached during the work presented in this dissertation. Some remaining considerations are presented that can built upon the work done in this dissertation.

In this dissertation, the context of public transport applications and associated topics was explored to ultimately build a solution to share information with the users of public transport in the city of Évora. The developed prototype application uses some of the most current development methods and frameworks, cross-platform development and the Flutter framework.

The development encountered some difficulties in the deployment of the back-end system, but at last, a functional solution was found. The application was evaluated against nonfunctional requirements established, such as usability, flexibility, reliability, and accessibility.

6.1 Conclusion

The city of Évora only has a passenger information system in the shape of timetables, which makes the solution of a mobile application extremely relevant and necessary. Using the mobile application developed will bring much needed autonomy to passengers that can lead to increase usage of the public transport network.

By the assessments made of other similar solutions, it is known the added value that those can bring to communities, and as the solution achieved in this dissertation meets the requirements elicited, it is fairly safe to assume that it can have some similar benefits in the same areas.

The passengers benefit from being able to access current public transit information from anywhere, allowing for them to make better decisions regarding their mobility, possibly saving time and reducing uncertainty. The transport agency, the city council, and all others business or people that are directly or indirectly involved with the public transport network, benefit from the possible increase in transport usage, lowered information cost and increased satisfaction with the network.

The research and work done in this dissertation is particularly relevant in the current social environment, impacted by the COVID-19 pandemic, as described in Section 3.2, efforts to improve transport occupancy are more important than in the past.

The technologies discovered and presented in this dissertation, such as OTP and GTFS, proved to be helpful to solve the problems encountered during the development of the proposed solution that could be implemented in the future. The approach taken was based on open-source projects, which is very useful when developing new applications, due to the amount of support on such communities. While some mentioned projects, such as GTFS, are backed by organizations like Google and the associated documentation is easily accessible, this approach can have disadvantages. In the case of ready-made solutions, the overall weight of the application can be heavier than a bespoke solution and there could be unforeseen limitations when may limit de completion of necessary features.

6.2 Final considerations

There are many aspects that could improve the solution presented, that could not be achieved at this time.

The lack of real-time information from the transport network, greatly hinders the benefits of the work done. If this information was available, the final solution of this dissertation could be improved and expanded.

Lastly, if the solution is deployed, long term evaluation and testing of the usage of the solution by passengers could give insight into possible shortcomings as well as data points related with transport usage that may not have been yet identified.

Bibliography

[Abo] About its uk – its united kingdom : Better transport through technology. https://its-uk.org.uk/about-its-uk/. Accessed: 2022-01-28.

[Acca] Accessibility | flutter. https://flutter.dev/docs/development/accessibility-and-localization/accessibility. Accessed: 2021-11-02.

[Accb] Xcode - wikipedia. https://en.wikipedia.org/wiki/Xcode. Accessed: 2021-11-03.

[Art] Article 9 – accessibility | united nations enable. https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/article-9-accessibility.html. Accessed: 2021-10-31.

[BAN+21] Haris Ballis, Filippos Alogdianakis, Paraskevas Nikolaou, Katerina Stylianou, and Loukas Dimitriou. Fusion of general transit feed specification (gtfs) and passenger count data: A case study on cyprus transit system. *International Conference on Models and Technologies for Intelligent Transportation Systems*, pages 1–6, 9 2021.

[BCE+10] Hannah Bast, Erik Carlsson, Arno Eigenwillig, Robert Geisberger, Chris Harrelson, Veselin Raychev, and Fabien Viger. Fast routing in very large public transportation networks using transfer patterns. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6346 LNCS:290–301, 2010.

[BDS+08] Reinhard Bauer, Daniel Delling, Peter Sanders, Dennis Schieferdecker, Dominik Schultes, and Dorothea Wagner. Combining hierarchical and goal-directed speed-up techniques for dijkstra's algorithm. *WEA 2008, LNCS 5038*, 2008.

[BMW15] Candace Brakewood, Gregory S. Macfarlane, and Kari Watkins. The impact of real-time information on bus ridership in new york city. *Transportation Research Part C: Emerging Technologies*, 53:59–75, 4 2015.

[Bou15] Kasia Bourée. Public transport network timetable exchange (netex) support for accessibility in netex. Technical report, CEN, 2015.

[CG08] Tonci Caric and Hrvoje Gold. *Vehicle Routing Problem*. InTech, 9 2008.

[CIV10] CIVITAS. Civitas - innovative information systems for public transport. Technical report, Institute for Transport Studies, University of Natural Resources and Applied Life Sciences (BOKU), Vienna. L, 2010.

[CLV+15] Pieter Colpaert, Alejandro Llaves, Ruben Verborgh, Oscar Corcho, Erik Mannens, and Rik Van De Walle. Intermodal public transit routing using linked connections. *Proceedings of the 14th International Semantic Web Conference: Posters and Demos*, 2015.

[DA20] Drew Dara-Abrams. An open platform for transit agencies to improve the quality of their real-time data final report for transit idea project 93. www.trb.org/idea, 2020.

[Dep] Deployments – onebusaway. https://onebusaway.org/onebusaway-deployments/. Accessed: 2021-09-04.

[DPSW13] Julian Dibbelt, Thomas Pajor, Ben Strasser, and Dorothea Wagner. Intriguingly simple and fast transit routing. *undefined*, 7933 LNCS:43–54, 2013.

[DPSW18] Julian Dibbelt, Thomas Pajor, Ben Strasser, and Dorothea Wagner. Connection scan algorithm. *ACM Journal of Experimental Algorithmics*, 23, 8 2018.

[DS08] Nikolaos Daskalakis and Anthony Stathopoulos. Users' perceptive evaluation of bus arrival time deviations in stochastic networks. *Journal of Public Transportation*, 11:25–38, 12 2008.

[DSSW09] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. *Algorithmics*, chapter Engineering Route Planning Algorithms, page pp. 117–139. Springer-Verlag Berlin Heidelberg, 2009.

[Edg] Understanding edge relaxation for dijkstra's algorithm and bellman-ford algorithm | by yasufumi taniguchi | towards data science. Accessed: 2021-11-18.

[EUR] Eur-lex - 32010l0040 - en - eur-lex. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEXAccessed: 2021-08-29.

[fL17] Transport for London. Assessing the value of tfl's open data and digital partnerships. Technical report, Transport for London, 2017. Accessed: 2021-08-08.

[Flu] Flutter - build apps for any screen. https://flutter.dev/. Accessed: 2022-01-28.

[FR21] Arthur Finkelstein and Jean Charles Régin. Using goal directed techniques for journey planning with multi-criteria range queries in public transit. *ICORES 2021 - Proceedings of the 10th International Conference on Operations Research and Enterprise Systems*, pages 347–357, 2021.

[FT20] Authors Frick and Karen Trapenberg. The general transit feed specification (gtfs) makes trip-planning easier-especially during a pandemic-yet its use by california agencies is uneven. Technical report, Institute of Transportation Studies, Berkeley, 2020.

[FWB10] Brian Ferris, Kari Watkins, and Alan Borning. Onebusaway: A transit traveler information system. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 35 LNICST:92–106, 2010.

[GdSD14] Rui Gomes, Jorge Pinho de Sousa, and Teresa Galvão Dias. A grasp-based approach for demand responsive transportation. *International Journal of Transportation*, 2:21–32, 4 2014.

[GdSD15] Rui Gomes, Jorge Pinho de Sousa, and Teresa Galvão Dias. Sustainable demand responsive transportation systems in a context of austerity: The case of a portuguese city. *Research in Transportation Economics*, 51:94–103, 9 2015. Paper on a simulation of a demand transportation system in the city of Porto. Conclusions aren't very positive as this model can easly fail with bad service design.

[GTFa] Gtfs realtime overview | realtime transit | google developers. Accessed: 2021-08-03.

[GTFb] Gtfs static overview | static transit | google developers. https://developers.google.com/transit/gtfs. Accessed: 2021-08-03.

[His] History – open transit software foundation. https://opentransitsoftwarefoundation.org/overview/history/. Accessed: 2021-09-04.

[How] How to perform top-notch mobile accessibility testing | perfecto by perforce. https://www.perfecto.io/blog/web-mobile-accessibility-testing-automation. Accessed: 2021-11-02.

[Int] Intelligent transport systems | mobility and transport. https://ec.europa.eu/transport/themes/its_en. Accessed: 2021-08-08.

[ITS] Intelligent transportation system - wikipedia. https://en.wikipedia.org/wiki/Intelligent_transportation_system. Accessed: 2022-01-28.

[Lec17] Marie Lecerf. European disability policy - from defining disability to adopting a strategy. Technical report, European Parliamentary Research Service, 2017.

[Lib20a] Rtig Library. Providing vehicle occupancy data:- data interfaces. Technical report, RTIG, 2020.

[Lib20b] Rtig Library. Real time information exchange using the siri standard. Technical report, RTIG, 2020.

[Lin] Linked connections. https://linkedconnections.org/#what. Accessed: 2021-09-04.

[MHC13] Andres Monzon, Sara Hernandez, and Rocio Cascajo. Quality of bus services performance: Benefits of real time passenger information systems. *Transport and Telecommunication*, 14:155–166, 6 2013.

[NeT] Netex | network timetable exchange. http://netex-cen.eu/. Accessed: 2021-09-22.

[NJS15] NJSK. Public transport-service interface for real-time information relating to public transport operations-part 1: Context and framework. Technical report, European Committee for Standardization, 2015.

[Opea] Open data policy - transport for london. https://tfl.gov.uk/info-for/open-data-users/open-data-policy. Accessed: 2021-08-08.

[Opeb] Opentripplanner. http://www.opentripplanner.org/. Accessed: 2021-11-20.

[OTP] Opentripplanner - openstreetmap wiki. https://wiki.openstreetmap.org/wiki/OpenTripPlanner. Accessed: 2022-01-14.

[Paj] Thomas Pajor. Algorithm engineering for realistic journey planning in transportation networks. Doctoral Thesis from: 2013-11-15.

[Pio13] Pioneering open data standards: The gtfs story. https://beyondtransparency.org/chapters/part-2/pioneering-open-data-standards-the-gtfs-story/, 2013. Accessed: 2021-08-03.

[Pla] Plan an accessible journey - transport for london. https://tfl.gov.uk/transport-accessibility/plan-an-accessible-journey?intcmp=5331#on-this-page-4. Accessed: 2021-10-26.

[PP16] Caroline Prodhon and Christian Prins. Metaheuristics for vehicle routing problems. *Metaheuristics*, pages 407–437, 1 2016.

[PPD+20] Postsavee Prommaharaj, Santi Phithakkitnukoon, Merkebe Getachew Demissie, Lina Kattan, and Carlo Ratti. Visualizing public transit system operation with gtfs data: A case study of calgary, canada. *Heliyon*, 6:e03729, 4 2020.

[Proa] Pros and cons | netex. http://netex-cen.eu/?page_id=185. Accessed: 2021-11-15.

[Prob] Provide vehicle occupancy data with gtfs - transit partners help. https://support.google.com/transitpartners/answer/10106342?hl=en&ref_topic=10119651. Accessed: 2021-10-26.

[Rem] proposal: remove "google" from the name of gtfs. https://groups.google.com/g/gtfs-changes/c/ob_7MIOvOxU. Accessed: 2021-08-08.

[RES] What is a rest api? https://www.redhat.com/en/topics/api/what-is-a-rest-api. Accessed: 2021-11-25.

[RR16] Mauricio G.C. Resende and Celso C. Ribeiro. Optimization by grasp. *Optimization by GRASP*, 2016.

[Sea] Search space. https://courses.cs.washington.edu/courses/cse473/06sp/GeneticAlgDemo/searchs.html. Accessed: 2022-01-28.

[Spe] Spec update for january 11, 2010. https://groups.google.com/g/gtfs-changes/c/iC18a8NY6N4. Accessed: 2021-08-08.

[Sta] Reference | static transit | google developers. Accessed: 2021-08-06.

[Str] How google and portland's trimet set the standard for open transit data – streetsblog san francisco. Accessed: 2021-08-03.

[Tec] Technical guidance: Siri-vm - gov.uk. https://www.gov.uk/government/publications/technical-guidance-publishing-location-data-using-the-bus-open-data-service-siri-vm/technical-guidance-siri-vm#the-bus-open-data-service. Accessed: 2021-09-22.

[Thea] Mod learning center: The role of data specifications in creating an integrated transportation system. https://learn.sharedusemobilitycenter.org/casestudy/the-role-of-data-specifications-in-creating-an-integrated-transportation-system/. Accessed: 2021-11-15.

[Theb] The state of open data - transport. https://www.stateofopendata.od4d.net/chapters/sectors/transport.html. Accessed: 2021-08-08.

[Traa] Transmodel at a glance – transmodel. http://www.transmodel-cen.eu/transmodel-at-a-glance/. Accessed: 2021-10-29.

[Trab] Contribute | transportr. https://transportr.app/contribute/. Accessed: 2021-09-04.

[Trac] How moovit improved its app to help people with disabilities ride transit with confidence | transform. https://news.microsoft.com/transform/moovit-helps-people-disabilities-ride-transit-confidence/. Accessed: 2021-10-26.

[Trad] Transreport app improves accessibility for disabled passengers. https://www.railway-technology.com/features/transreport-app-improves-accessibility-for-disabled-passengers/. Accessed: 2021-10-26.

[UIT20] Open letter : European ceos and city representatives call for local public transport to be a key sector in the european recovery plan, the international association of public transport. https://cms.uitp.org/wp/wp-content/uploads/2020/08/PUBLIC-TRANSPORT-IS-CRITICAL-FOR-EUROPEAN-RECOVERY-FINAL-VERSION.pdf, 2020. Accessed: 2021-11-08.

[Who] Who's left riding public transit? a covid data deep-dive. – transit. https://blog.transitapp.com/whos-left-riding-public-transit-a-covid-data-deep-dive/. Accessed: 2021-11-08.

[Why] Why the bus open data service should include gtfs | by tom quay | medium. https://medium.com/@tomquay/why-the-bus-open-data-service-should-include-gtfs-67c6ef4208b4. Accessed: 2021-10-29.

[Won13] James Wong. Use of the general transit feed specification (gtfs) in transit performance measurement. *undefined*, 2013.

[Wor] World's largest public transport app now accessible for people with vision impairment - rnib - see differently. https://www.rnib.org.uk/nb-online/largest-public-transport-app-now-accessible. Accessed: 2021-10-26.

[YCN+19] Godwin Yeboah, Caitlin D. Cottrill, John D. Nelson, David Corsar, Milan Markovic, and Peter Edwards. Understanding factors influencing public transport passengers' pre-travel information-seeking behaviour. *Public Transport*, 11:135–158, 6 2019.

UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:
Universidade de Évora
Instituto de Investigação e Formação Avançada — IIFA
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94
7002 - 554 Évora | Portugal
Tel: (+351) 266 706 581
Fax: (+351) 266 744 677
email: iifa@uevora.pt