

transitway

Popescu Andrei-Cristian

Soare Cătălin-Ștefan

Stanciu Gabriel

1. What is transitway?

Transitway is an app that assists with public transport by providing accurate estimates as well as the exact location of transport. Transitway also processes payments to provide ticket and pass purchase service.

2. What do we aim to achieve?

Transitway aims to be an integrated system that will benefit both passengers and transport companies. All solutions offered are designed to be very easy to integrate with existing solutions.

3. How to get the location?

3.1 Location of means of transport

3.1.1 Via the transitway tracker (detailed in section 4)

3.2 User location

3.2.1 Via the user application, using the device location

4. Transitway trackers

4.1. Real-time location tracker

4.1.1 Operating principle

Every second, the [GPS](#) module (see section 4.1.2.1) records data such as latitude, longitude and speed. This data is then sent to the microprocessor (see section 4.1.2.3) for analysis and encryption. The location information is sent to the SIM module (see section 4.1.2.2), which uses cellular connectivity and, via an HTTP POST method, uploads the encrypted data to the server that can use the received data to serve the application users.

4.1.2 Components

The transitway tracker is made up of three essential components for locating and transmitting location:

4.1.2.1 GPS module

GPS module [U-Blox NEO 6M](#) offers good performance at a good price. It has a low data acquisition time thanks to the [AssistNow Autonomous](#) which uses the [GNSS](#) to generate location predictions in around 2 seconds. In the future, the module can be replaced with other chips such as [UBX-M9340](#) which offer additional functions: [Dead Reckoning](#), better stability and a smaller package with more efficient management.

4.1.2.2 SIM module

SIM module [SIM800L](#) uses the 2G cellular network to send location data to a server. In the future, it may be enhanced with LTE or 5G compatible chips such as [SIM8260E](#) for greater connection stability and extended uptime.

4.1.2.3 Microprocessor

Microprocessor [ATmega328](#) is cheap and efficient, meeting the requirements for transitway. It

operates at 16MHz and has 32KB of flash memory, sufficient for the required operations.

4.1.2.4 Power supply

The circuit receives power from a power source via an XT60 connector, with a voltage of up to 40V (recommended 12V at 3A). The circuit includes a battery that constantly charges from the main source and takes over power when the source goes off.

Thus, the connection with satellites and modules [RTC \(Real Time Clock\)](#) is maintained .

4.2. Tracker for monitoring passenger numbers

4.2.1 Principle of operation

The tracker is designed to record the number of passengers boarding and alighting a means of transport in real time. It uses an [infrared](#) (IR) [sensor](#) to detect the presence of passengers and an [Arduino UNO](#) microcontroller to record the relevant data. The collected information is then transmitted to a server via an [800L SIM](#) module.

4.2.2 Component

The passenger number tracker consists of the following main components:

4.2.2.1 IR sensor

[The infrared](#) (IR) [sensor](#) is used to detect the presence of passengers in the means of transport. It emits an invisible infrared beam and detects changes in the reflection of the beam. The sensor can be placed in different areas, such as access doors or seats. When a passenger passes through the sensor's range, it registers a change and transmits the information to the microcontroller.

4.2.2.2 Arduino UNO Microcontroller

The [Arduino UNO](#) microcontroller is the central component that receives the signals from the IR sensor and manages the passenger count registration. It is also responsible for managing the network connection and transmitting data to the server via the SIM 800L module.

4.2.2.3 SIM module 800L

[The SIM 800L](#) module is used to allow the tracker to connect to the cellular network and transmit data to the server via an HTTP POST request.

4.2.2.4 LED

An [LED](#) is used to indicate the status of the tracker (faulty/functional)

4.2.2.5 Power supply

The circuit receives power from a power source via a connector [XT60](#)connector with a voltage up to 40V (recommended 12V at 3A).

5. Generating the travel route

Transitway offers advanced trip route generation functionality, which optimises the route and travel time for users. Using intelligent planning algorithms, the platform analyzes information about the user's location, desired destination, transportation schedule and traffic conditions to identify the most efficient route available. This functionality allows users to quickly get precise directions on how to reach their destination using public transport. Transitway can suggest optimal combinations of buses, trams or other means of transport according to users' preferences.

Transitway also monitors real-time traffic conditions and possible delays of transport modes, providing real-time updates and alternatives to ensure smooth journeys and minimise waiting time, using the [Routes API from Google Maps](#).

6. Payments and subscriptions

6.1 Payments

Payments are processed by [Stripe](#) for a quick implementation. This allows users to securely top up their transitway+ account. The transitway+ account balance is used to pay for tickets or passes, making the purchase process easier.

6.2 Tickets and season tickets

Users can easily purchase tickets or season tickets directly from the current journey tab or from the dedicated section at the touch of a button. Tickets and season tickets are stored in the app as a QR code that can be easily checked and users are notified when their travel passes expire. Benefits include a streamlined experience and limits the need to stand in line or search for vending machines.

7. Backend and Frontend

7.1 Backend

The backend of the Transitway application is based on the programming language [Go](#) programming language, an efficient and powerful language for web application development. Instead of the basic HTTP service framework, Transitway uses [Fiber](#), a fast and easy alternative for developing web services in [Go](#). [Fiber](#) is based on [fasthttp](#), an HTTP library in [Go](#), which provides superior performance over the standard [net/http](#) library.

7.1.4 Database and file hosting.

For data storage, Transitway uses [MongoDB](#), a scalable and flexible [NoSQL](#) database. The integration between [Go](#) and [MongoDB](#) is achieved using the official [mongo-go-driver](#) library, which provides advanced functionality for interacting with the database.

To optimize performance and reduce access time to frequently used data, Transitway uses [Redis](#), an in-memory caching system. [Redis](#) is used in particular to store and access authentication codes quickly, ensuring a smooth user experience. Interaction with [Redis](#) is achieved through the [Redis for Go](#) library.

7.1.4 Authentication management

For managing authentication and sending authentication codes to users via SMS, Transitway uses the [Twilio](#) service and the [twilio-go](#) library. Twilio provides a powerful API for sending SMS messages and managing user interactions. For user authentication and token generation, Transitway uses the [jwt](#) library, which provides advanced functionality for handling and verifying [JSON tokens](#). This library offers enhanced performance over the standard library and is optimized to meet the needs of the application.

7.1.4 Data security and encryption

To secure sensitive data such as user passwords, Transitway uses the [bcrypt hashing](#) algorithm. Passcodes are stored in [Redis](#) as bcrypt-encrypted hashes using the [standard crypto library in Go](#). For [JSON](#) data serialization and deserialization, Transitway uses the [encoding/json](#) package from the Go standard library. It allows easy conversion of data structures to JSON and vice versa, facilitating communication between client and server.

7.1.5 Payment system

Integration between the external and internal payment system is achieved through [Stripe](#), a popular and secure online payment platform. To interact with the Stripe service, Transitway uses the [stripe-go](#) library, which provides modern, secure and standardised methods for managing transactions and the in-app payment process.

7.2 Frontend

The application was developed with the [Flutter](#) framework, a cross-platform toolset created by Google. [Flutter](#) enables the development of high-performance mobile, web and desktop applications using the same code base. The app uses various technologies and libraries for its functionality:

- [Flutter: The Flutter SDK](#) provides UI components, tools and APIs for native interface development. Uses the programming language [Dart](#) programming language and adopts a reactive and declarative model.
- [Google Maps Flutter](#): Google [maps](#) are integrated into the application with the `google_maps_flutter` package. This Flutter plugin allows you to display interactive maps, customize markers, polylines and more. It also provides functionality for interacting with users and obtaining location data.
- [Provider](#): The Provider library is used for state management in the application. It implements the Provider model, allowing data to be shared and updated in different parts of the application.
- [Geolocator](#): The geolocator package is used to obtain the current location of the device. It provides a simple API to retrieve latitude, longitude and other location related information. Geolocator abstracts platform-specific location services, making it easy to access location data on both Android and iOS.
- [Geocoding](#): The geocoding package is used to convert addresses to geographic coordinates and vice versa. It allows the application to perform geocoding and reverse geocoding operations required for functions such as location search and displaying markers on the map.
- [Dart Streams](#): Flutter uses Dart's built-in Streams to manage asynchronous events and data streams. In the application, streams are used to retrieve position updates from the Geolocator package and update the map view accordingly.

Overall, the app combines the power of Flutter's interface tools with libraries such as `google_maps_flutter`, `provider`, `geolocator` and `geocoding` to create an interactive and feature-rich app for public transport. Using these technologies, the app provides functionality such as map display, route calculation, user location retrieval and location search, providing a seamless user experience across all platforms.

8. User Interface/ User Experience

8.1 Design philosophy

The interface was found to be very intuitive. Thus, with a small number of taps you can reach any menu.

8.2 The design process

The UI was designed in [Adobe XD](#) to create prototypes of the application. The graphics were designed in [Adobe Illustrator](#) and [Photoshop](#) and was exported as vectors to maintain its quality. From here we refined the design using the criteria:

- Ease of use
- Modernism and minimalism
- Intuitiveness

9. Method of calculating estimated time of arrival

9.1 Current method

To provide accurate estimates of estimated arrival times, we combine multiple data sources and advanced technologies. We use the [RoutesAPI](#) API [from Google Maps](#) to get information about available routes and take into account real-time traffic data.

Thanks to our advanced approach and the use of multiple data sources, we can ensure high accuracy in providing arrival estimates. We focus on providing real-time information and adjusting estimates

according to any changes in traffic conditions.

9.2 Next steps

Knowing the distances between stations, the last 10 known speeds of the means of transport are averaged in real time. (The whole system operates at 1Hz) - one operation per second. This allows an accurate arrival time to be determined. The Transitway app constantly collects the times of all means of transport and the number of passengers in the vehicle, creating a prediction model that can be used to improve route estimates and recommendations. This data can also be provided to transport companies to make urban mobility more efficient.

10. Integration

We have thought transitway from the very beginning to be a platform as easy to integrate as possible. So simply installing the trackers, without constant driver intervention or trained staff, is all that is needed. Once plugged in, the module will automatically take its current position and send it to the server for use by the system (the tracker is optional, the [RoutesAPI](#) API mentioned above can be used, depending on the carrier's requirements).

11. Conclusions

The Transitway team has demonstrated that teamwork is essential to achieve a successful project in a very short time. We collaborated effectively, with an emphasis on equal roles, and created an integrated and user-friendly platform. By implementing our innovative solution, we are determined to bring significant benefits to local communities, making urban transport an attractive and efficient option to travel from place to place. As a small team of just 3 students, and with no help from adults, we have managed to develop a future-proof app with huge potential.

Bibliography:

U-Blox NEO 6M GPS module - Official website: <https://www.u-blox.com/en/product/neo-6-series>

SIM Module SIM800L - Official website: <https://simcom.ee/modules/gsm-gprs/sim800l/>

ATmega328 Microprocessor - Official website:

<https://www.microchip.com/wwwproducts/en/ATmega328>

Fiber - Official website: <https://gofiber.io/>

FastHTTP - Official website: <https://github.com/valyala/fasthttp>

MongoDB - Official website: <https://www.mongodb.com/>

Mongo Go Driver - Official website: <https://github.com/mongodb/mongo-go-driver>

Redis - Official website: <https://redis.io/>

Twilio - Official website: <https://www.twilio.com/>

sjwt - Official website: <https://github.com/dgrijalva/jwt-go>

Stripe - Official website: <https://stripe.com/>

Flutter - Official website: <https://flutter.dev/>

Google Maps Flutter - Official website: https://pub.dev/packages/google_maps_flutter

Provider - Official website: <https://pub.dev/packages/provider>

encoding/json - Official website: <https://golang.org/pkg/encoding/json/>

bcrypt - Official website: <https://pkg.go.dev/golang.org/x/crypto/bcrypt>