

UDC 004:656.614.2 (045)

¹L. M. Oleschenko,
²A. Y. Shrol

TECHNOLOGY OF REGISTRATION OF PASSENGERS BY THE DRIVER OF BUS IN INTERCITY CONNECTIONS

¹Department of System Programming and Specialized Computer Systems, ²Department of Computer Engineering, National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine

E-mails: ¹olm-86@mail.ru, ²Redlayn7@gmail.com

Abstract—The technology of reserving the places in the bus of intercity connections by a driver is described in the article and a software product is developed for mobile devices and tablets on the base of the operating system of Android. For creation the application program Android Studio and Android SDK were utilized from API version 16. Registration of passengers takes place by REST-requests. Requests which allow to authorize and register a driver in the system, to get information about routes and begin them, report about taking the place and displacement are realized.

Index Terms—APRS; client-server technologies; database; registration of passengers; transport vehicle; Android; REST-request; intertown passenger transportations; quality of service for passengers.

I. INTRODUCTION. STATEMENT OF THE PROBLEM

Control system of modern motor transport enterprises (MTE) constitute a dynamic information system (IS) with the branches of feed-back. To build links of communication between moving participants, there is an urgent need to build fast and reliable data transmission channels. Within cities typically is used system of packet data exchanging on the basis of national mobile operators. Modern services of taxi duplicate information rarely by official VHF of communication channel. A situation looks differently for a ferryman who serves intertown routes. Unfortunately, at the time of writing the article batch communications in cellular networks works on majority of territory of Ukraine only in theory [1].

In previous articles proposedes deployment of a backup VHF channel of transceiver the data, the location of the head local station to address the challenges of communication MTE Chernihiv and parts of the Kyiv region [2]. From the point of view of the author, it looks promising and it would also except sending a voice message on the basis of the proposed equipment transfer packet data. During the study of intercity passenger transport of Chernihiv numerous conflicts between passengers and drivers because of the lack of support information to drivers of vehicles and controllers had identified [1] – [3]. In obedience to the current legislation of Ukraine, on an intertown route transportation of stand-up passengers is forbidden. In the regions of Ukraine bus tickets can be bought in a cashdesk, using the Internet (in the automated system the "General electronic ticket")

[1]. However there are always «random» passengers who have no possibility to buy a ticket, especially in rural settlements, placed between the cities of region. As a result of that a driver does not register such passengers into proper concerned places and the sale of tickets is conducted. The relevant occupied space is the ticket. Therefore, to improve the quality of passenger services in Ukraine, the MTE needs to use modern information technology (it) for the registration of "random" passengers in a moving vehicle in the on-line mode. Developed solutions for the driver must have a convenient user interface to avoid emergency situations during transmission of data to the server. The aim of the article is the analysis developed on the basis of the Android operating system software product for driver, which allows to register random passengers in minimal time in on-line mode.

II. USAGE OF TECHNOLOGY FOR APRS DATA TRANSMISSION OF MTE

Currently, for long-distance bus passenger transport there are a large number of IT technologies to automate telephony, to implement IP-PBX, to develop websites for on-line ticket booking, connection of GPS monitoring, the settings of the GPS trackers etc. To control the quality of passenger service on buses CCTV systems should be installed. However, for long-distance transportation the registration system is not sufficiently developed for the "random" passengers who occupy seats, their embarkation and debarkation, which makes it impossible for adequate ticket sales from the bus station to the appropriate places in the bus. Considering equipment VHF communication channel, there are several solutions,

such as: X.25 (AX.25), Code-5, ALE, SIM-31, WSJT, D-STAR, APRS. Previous studies highlighted the budgetary system of communication between driver and dispatcher MTE using the APRSDroid application for VHF radio. Network of automatic transmission of digital data radio APRS is the most adapted for implementation in MTE [4]. In APRS such parts of the system are highlighted: the base station and portable, in Figs 1 and 2 is given their block diagram.

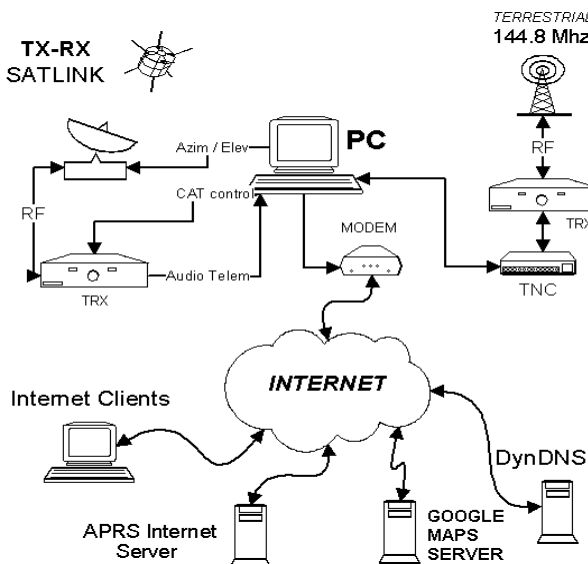


Fig. 1. Scheme of using the technology of APRS

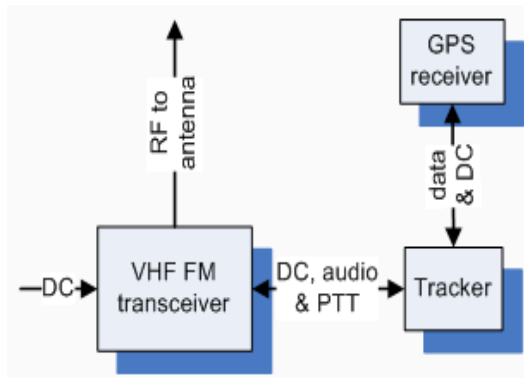


Fig. 2. Block diagram of the portable station

The APRS technology allows you to track vehicle movement on a map in the on-line and to exchange short messages (Fig. 3). The disadvantage of this system is awkward interface for the driver when transmitting data, no automatic reservation on the server. Existing app APRSDroid allows you to exchange packet data through the radio channel according to the APRS Protocol and transport Protocol of TCP. For data transmission on VHF channel APRSDroid has a free API. This allows you to tailor the application to the requirements of the MTE and to create your own GUI for devices with operating system (OS) Android.



Fig. 3. Sending of reporting from APRSDroid

III. APPLICATION FOR DRIVER'S INTERACTION WITH THE DATABASE OF THE MTE ON-LINE

To improve the quality of passenger service it is required an automated system of tickets booking by the driver of the vehicle. For the purpose of joint the use of radio and cellular coverage, the authors developed an application for mobile devices and tablets based on Android OS. It is used Android Studio and Android SDK with API version 16, which provides compatibility in more than 90 % of all available Android – powered devices. Development of a software product is according to the server that holds all the data about routes, drivers, TC. The interaction with the server happens via a REST-request (Fig. 4) [5] – [7].

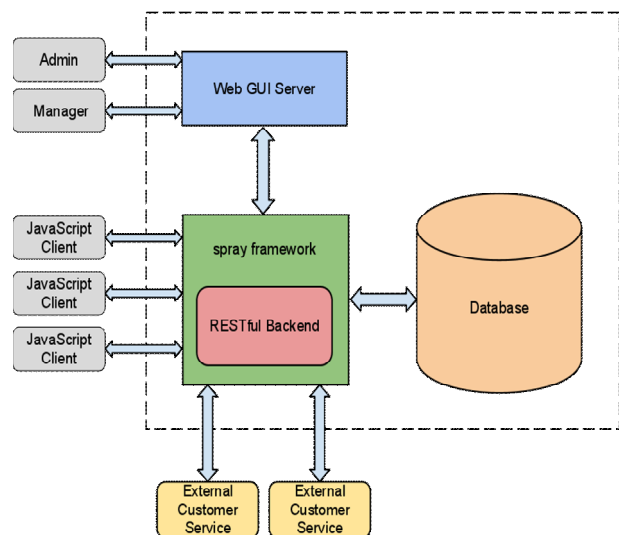


Fig. 4. Chart of client-server co-operation

To create a back-end layer it is used an implementation REST in the form of the JAX-RS and implementation Jersey2.2 [8], as a layer to access the database (DB) DAO pattern is used. The database connection implements through a data source on the server that is indexed via JNDI. Implemented request

allows to confirm and register the driver in the system, get information about routes, start them, to report about the boarding and disembarkation of passengers. To create GUI standard methods were used for the Android platform, namely the interface description in the file *.xml (Fig. 5).

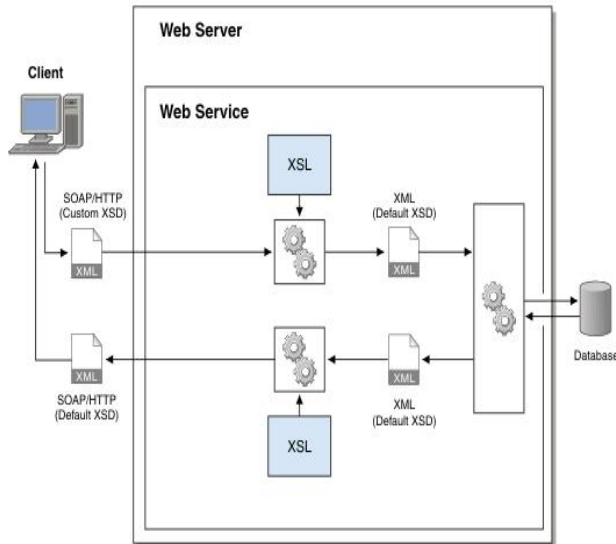


Fig. 5. Creation of the exchange file XML

The software product has multiple main windows: the start window, the window of authorization and registration, the window to select the route and edit information about places in the transport.

The launch window contains a button of authorization and registration, which will open the windows of the authorization and registration in accordance.

The authorization window contains fields for entering your login and password that were specified when registering the driver and the sign-in button. When you click on the login button username and password are checked. If fields are missing data, the program informs you. In case of successful filling of forms, the program does the REST – authorization request and waits for a response from the server. If the server finds data that has been entered in the field, the transition occurs to the main part of the program. The registration window contains fields for entering your login and password that will be required for future login to the app, the fields of first and last name, a button to confirm registration. When you click on the registration button all the fields are checked for matching and confirmation of the password to avoid errors in the password. If fields contains different passwords, the program informs you. In case of successful filling of forms, the program does the REST – the registration request and waits for a response from the server (Fig. 6).

If the server successfully performs the operation, it reports about it and a transition to the main part of

the program takes place. During the opening of the main part of the application it is checked whether the driver has already the active route and thus opens the window for creating or displaying the route. Also in the main window there is a sidebar where the user can open a window to display or create a route, edit account, information about the software product and the exit from account.

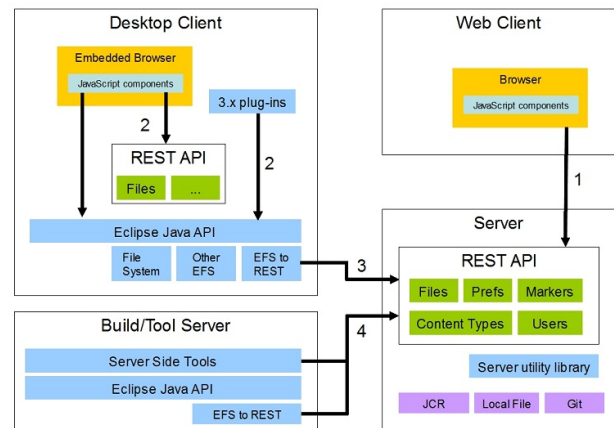


Fig. 6. Structure of the REST request for the client to communicate with the server

The window of creation of a route contains drop-down lists to select the departure point, destination point, route selection, button of the beginning of the route. After selecting waypoints, the program sends a request to the server to search existing routes with the same parameters for the start and end and display this information in drop-down list routes. After selecting the parameters and pressing the “Start route” data is send to the server and if send successfully, a window opens with a demonstration of the route.

In the “Start route” the start and end point of the route and scheme of the vehicle interior are displayed. While opening this window, the application sends a request to BusManager, which receives information about the route points of the route and information about occupied seats.

While driving vehicle and making boarding passengers the driver should note the place that he or she took and in order to do it the driver can just press the corresponding button on the screen. At this point, the program sends a query like "http://bus-nikichxp.rhcloud.com/api/route/addclient?token=9b68140b-ef2a-4711-94e6-3da5dded5793&seatnumber=2", the server adds to the database information indicating that the seat # 2 is occupied and sends back the response about the success of the process. As a result, the button indicating the occupied seat will change color from green to red which means that the place is busy. If the driver dropped off the passenger, he presses the

corresponding button, data is send to the server, and if successful, the button becomes green, showing a spare place.

For creating a schema the class Generator, and adapter NumberedAdapter are responsible. In the first class analysing of the number of seats in the vehicle take place. This data is transmitted in Numbered Adapter, which creates the appropriate number of buttons that correspond to the number of seats.

Generator is responsible for this fragment code:

```
public View onCreateView(LayoutInflater inflater,
    ViewGroup container, Bundle savedInstanceState) {
    View v =
inflater.inflate(R.layout.activity_recycler_view,
    container, false);
    SharedPreferences tok =
getActivity().getSharedPreferences("token", 0);
    String token = tok.getString("token", null);
    SharedPreferences settings =
getActivity().getSharedPreferences("route", 0);
    int seats = settings.getInt("seats", 0);
    if (seats>21){
        column = 4;
    } else column = 3;
    RecyclerView recyclerView =
(RecyclerView) v.findViewById(R.id.recycler_view);
    recyclerView.addItemDecoration(new
MarginDecoration(getActivity()));
    recyclerView.setHasFixedSize(true);
    //recyclerView.setLayoutManager(new
GridLayoutManager(getActivity(), column));
    //recyclerView.setAdapter(new
NumberedAdapter(seats));
    recyclerView.setLayoutManager(new
GridLayoutManager(getActivity(), 3));
    recyclerView.setAdapter(new
NumberedAdapter(getActivity(), 19, token));
    return v; }
```

For creation of buttons in NumberedAdapter the following fragment of code is responsible:

```
@Override
public ToggleButtonHolder
onCreateViewHolder(ViewGroup parent, int viewType) {
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.
bus_scheme_item, parent, false);
    return new ToggleButtonHolder(view); }
@Override
public void onBindViewHolder(final
ToggleButtonHolder holder, final int position) {
    final String label = labels.get(position);
    holder.toggleButton.setText(label);
    holder.toggleButton.setTextOn(label);
    holder.toggleButton.setTextOff(label);
    if (iSeats.contains(position+1)){
        holder.toggleButton.setChecked(true);}
    holder.toggleButton.setOnClickListener(new
View.OnClickListener() {
```

```
@Override
public void onClick(View v) {
    if (holder.toggleButton.isChecked()){
        Toast.makeText(holder.toggleButton.getContext(),
        "activate "+(position+1),
        Toast.LENGTH_SHORT).show();
        mApi.addClient(sToken, position);
    }else{
        Toast.makeText(holder.toggleButton.getContext(),
        "deactivate "+(position+1),
        Toast.LENGTH_SHORT).show();
        mApi.removeClient(sToken, position);
    }
}); }
```

For sending the request and receiving the result from the server meets the Volley library. Because the response from the server comes as a text in JSON format, it is needed to bring its contents to the data that correspond to the structure of the program. For this purpose it was used the GSON library.

Example f a request to get directions and their processing:

```
public static final String URL
="http://bus-nikichxp.rhcloud.com/api/";
public void getRoutes(int from, int to){
    gson = new Gson();
    String URL_Request =
URL+"route/get?from="+from+"&to="+to;
    Log.d("q getRoutes", URL_Request);
    jsonObjectRequest = new
JsonObjectRequest(Request.Method.GET, URL_Request,
new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                JSONObject js = response; String s =
js.getString("error");
                Toast.makeText(ctx, "Error: "+s,
                Toast.LENGTH_SHORT).show();
            } catch (JSONException e) {
                if (e.getMessage().equals("No value for
error")){
                    RoutePack routePack =
gson.fromJson(response.toString(), RoutePack.class);
                    mGetRoutesListener.onGetRoutes(routePack);
                } else Toast.makeText(ctx, "Error:
"+e.getMessage(), Toast.LENGTH_SHORT).show(); }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError
error) {
                try {
                    Log.d("error", error.getMessage());
                } catch (NullPointerException e){
                    Toast.makeText(ctx, "Server error",
                    Toast.LENGTH_SHORT).show(); } } }
    });
    mQueue.add(jsonObjectRequest); }
```

For addition and deleting of the concerned place in DB the following methods are utilized:

```
@GET
@Path("/addclient")
/**
 * Adding client to seat */
public String addClient (@QueryParam("token")String
token, @QueryParam("seatnumber")String seatNumber)
{
try {
seatNumber.trim();
DataContainer.getRoute(token).addClient(Integer.parseInt(seatNumber));
return gson.toJson(new Error("Success"));
} catch (Exception e) {
return gson.toJson(new
Error(e.toString()+seatNumber+"|"));
} }
@GET
@Path("/deleteclient")
/**
 * When client left seat */
public String deleteClient
(@QueryParam("token")String token,
@QueryParam("seatnumber")String seatNumber) {
DataContainer.getRoute(token).removeClient(Integer.parseInt(seatNumber));
return gson.toJson(new Error ("Success"));
```

V. CONCLUSIONS

A developed software for devices with Android OS can authenticate and authorize users to prevent unauthorized access to the system. Software product BusManager enables to register random passengers and to identify the availability of seats in the on-line mode. The app allows a driver in long-distance communication to send data about occupied seats on the server of MTE during movement and has a convenient user interface. The developed application provides the possibility of its use in a cellular coverage, with no coverage, the driver connects to the

backup VHF channel. This can cause inconvenience when transferring data to the server by the driver while driving the vehicle. Therefore, the aim of future research is the creation of a compatible automated system data transmission in a cellular network and a backup VHF radio to ensure the quality of service of passengers of MTE in real time.

REFERENCES

- [1] L. M. Oleschenko, and A. O. Moshenskyy, "Computer networks of communication of participants of a passenger-transport process". *Scientific messages of USEIC*, no. 1(29), 2014, pp. 82–86. (in Ukrainian).
- [2] L. M. Oleschenko, and A. O. Moshenskyy, "Experimental research of area of coverage of VHF of radio channel is for connection of controller of motor transport enterprise with the drivers of rolling stock." *Scientific messages of USEIC*. no 3(31), 2014, pp. 47–52. (in Ukrainian).
- [3] L. M. Oleschenko, "Use of automated wireless data transmission technologies in the construction of IT for MTE". *Telecommunication and informative technologies*. no (2), 2014. pp. 67–72. (in Ukrainian).
- [4] L. M. Oleschenko, "About possibility of introduction of technology of APRS on the intertown bus routes of region". *Scientific messages of USEIC*. no. 1(35), 2015, pp. 93–96. (in Ukrainian).
- [5] T. S. Mashnin, *Web services Technology Java platform*. St. Petersburg, BHV-Peterburg, 2012, 560 p. (in Russian).
- [6] E. Thomas, C. Benjamin, P. Cesare, and B. Raj, *SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST*. Prentice Hall, Canada. 2013. 624 p.
- [7] E. Wilde, and C. Pautasso, *REST: From Research to Practice*. Springer Science & Business Media, 2011. 528 p.
- [8] Jersey 2.2 User Guide [Electronic resource] <https://jersey.java.net/>.

Received August 04, 2015

Oleschenko Lyubov. PhD. Senior teacher.

Department of System Programming and Specialized Computer Systems, National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine.

Education: Taras Shevchenko Chernihiv State Pedagogical University, Chernihiv, Ukraine (2008).

Research interests: gravitational model and information technology in transport systems.

Publications: 25.

E-mail: olm-86@mail.ru

Shrol Andriy. Bachelor.

Department of the Computing Engineering, National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine.

Education: National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine.

Research interests: programming, client-server technologies.

Publications: 1.

E-mail: Redlayn7@gmail.com

Л. М. Олещенко, А. Ю. Шроль. Технологія реєстрації пасажирів водієм автобуса міжміського сполучення
Описано технологію бронювання водієм місць в автобусі міжміського сполучення та розроблено програмний продукт для мобільних пристроїв та планшетів на базі операційної системи Android. Для створення додатку використано програму Android Studio та Android SDK з API версії 16. Реєстрація пасажирів відбувається за допомогою REST-запитів. Реалізовано запити, які дозволяють авторизувати та реєструвати водія в системі, отримувати інформацію про маршрути та розпочинати їх, повідомляти про посадку та висадку пасажирів.

Ключові слова: APRS; клієнт-серверні технології; база даних; реєстрація пасажирів; транспортний засіб; Android; REST-запити; міжміські пасажирські перевезення; якість обслуговування пасажирів.

Олещенко Любов Михайлівна. Кандидат технічних наук. Старший викладач.

Кафедра програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут», Київ, Україна.

Освіта: Чернігівський державний педагогічний університет ім. Т. Г. Шевченка, Чернігів, Україна (2008).

Напрямок наукової діяльності: гравітаційні моделі, інформаційні технології в транспортних системах.

Кількість публікацій: 25.

E-mail: olm-86@mail.ru

Шроль Андрій Юрійович. Бакалавр.

Кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут», Київ, Україна.

Освіта: Національний технічний університет України «Київський політехнічний інститут».

Напрямок наукової діяльності: програмування, клієнт-серверні технології.

Кількість публікацій: 1.

E-mail: Redlayn7@gmail.com

Л. М. Олещенко, А. Ю. Шроль. Технология регистрации пассажиров водителем автобуса междугородного сообщения

Описана технология бронирования водителем мест в автобусе междугородного сообщения и разработан программный продукт для мобильных устройств и планшетов на базе операционной системы Android. Для создания приложения использована программа Android Studio и Android SDK из API версии 16. Регистрация пассажиров происходит с помощью REST-запросов. Реализованы запросы, которые позволяют авторизовать и регистрировать водителя в системе, получать информацию о маршрутах и начинать их, сообщать о посадке и высадке пассажиров.

Ключевые слова: APRS; клиент-серверные технологии; база данных; регистрация пассажиров; транспортное средство; Android; REST-запрос; междугородные пассажирские перевозки; качество обслуживания пассажиров.

Олещенко Любовь Михайловна. Кандидат технических наук. Старший преподаватель.

Кафедра программного обеспечения компьютерных систем, Национальный технический университет Украины «Киевский политехнический институт», Киев Украина.

Образование: Черниговский государственный педагогический университет им. Т. Г. Шевченко, Чернигов, Украина (2008).

Направление научной деятельности: гравитационные модели, информационные технологии в транспортных системах.

Количество публикаций: 25.

E-mail: olm-86@mail.ru

Шроль Андрей Юриевич. Бакалавр.

Кафедра вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт», Киев, Украина.

Образование: Национальный технический университет Украины «Киевский политехнический институт».

Направление научной деятельности: программирование, клиент-серверные технологии.

Количество публикаций: 1.

E-mail: Redlayn7@gmail.com