

به نام خدا

برای انتخاب مدرسه باید ایکن زیر را انتخاب کرد:



ایکن شروع / پایان سفر را انتخاب کرد

در رویداد کلیک این تابع تمام عملیات انتخاب اتفاق می افتد.

اولین if که در تابع کلیک شروع / پایان سفر اجرا می شود، بررسی می کند اگر سایز آرایه سرویس ها برابر با سفر باشد یعنی برای راننده سرویس ست نشده است.

```
if (MyApplication.totalSchool.size() == 0) {
    UserManager.setStateHeader("برای شما سرویسی اختصاص داده نشده است");
    txtCurrentStat.setText(UserManager.getStateHeader());
    AlertDialog errorDialog = new AlertDialog(context, MainActivity.this, dialogText: "برای شما سرویس تعیین نشده است");
    errorDialog.show();
    errorDialog.setOnDismissListener(new DialogInterface.OnDismissListener() {
        @Override
        public void onDismiss(DialogInterface dialog) {
        }
    });
} else {
```

همچنین در این if دایالوگی نمایش می دهد و می گوید که برای شما سرویسی تعیین نشده است،

به صورت کلی برای state ها یعنی وضعیت اینکه راننده در سفر رفت است یا برگشت و کدوم مدرسه را انتخاب کرده است در share prefernce ما 5 تا فیلد تعریف کردیم، این به این خاطر است که تا اگر برنامه بسته شد وضعیت از دست نرود و بداند که الان در سفر رفت بود یا برگشت و برای کدوم مدرسه

فیلد های مربوط به معلوم بودن وضعیت در share prefernce :

```

public static String getStateHeader() {
    return MyApplication.securePreferences.getString("StateHeader");
}

public static void setStateHeader(String state) {
    MyApplication.securePreferences.put("StateHeader", state);
}

public static String getSchoolCode() {
    return MyApplication.securePreferences.getString("schoolCode");
}

public static void setSchoolCode(String state) {
    MyApplication.securePreferences.put("schoolCode", state);
}

public static String getStateStart() {
    return MyApplication.securePreferences.getString("stateStart");
}

public static void setStateStart(String state) {
    MyApplication.securePreferences.put("stateStart", state);
}

public static String getStateReturn() {
    return MyApplication.securePreferences.getString("stateReturn");
}

public static void setStateReturn(String state) {
    MyApplication.securePreferences.put("stateReturn", state);
}

public static String getStateGo() {
    return MyApplication.securePreferences.getString("stateGo");
}

public static void setStateGo(String state) {
    MyApplication.securePreferences.put("stateGo", state);
}

```

این ها تابع هایی هستند که در کلاس UserManager من تعریف کردم برای نگه داشتن وضعیت سفر

```

UserManager.setStateHeader(" سفری آغاز نشده است ");
UserManager.setSchoolCode("-1");
UserManager.setStateReturn("0");
UserManager.setStateGo("0");
UserManager.setStateStart("0");

```

در هنگامی که برای اولین بار دستگاه وارد می شود من فیلد هایی که تعریف کردم را با مقدار های بالا ست می کنم. StateHeader برای این هست که در صفحه اصلی بالای صفحه بنویسد که الان در چه وضعیتی هست. SchoolCode کد مدرسه را می گیرد برای ارسال. StateReturn وضعیت برگشت را نگه می دارد. StateGo وضعیت رفتن را نگه میدارد. StateStart وضعیت شروع را نگه می دارد.

بس تا اینجا وقتی برای اولین بار برنامه بالا میاد اون بالای برنامه می نویسه سفری آغاز نشده است .

توی اون if اول میبینیم که text بالا از userManager.getStateHeader() خونه و ست می کند.

Else همین if گفته اگر تعداد سرویس هایی که داری مساوی یکی است. یک دایالوگی را نمایش بده که به صورت زیر است



روند انتخاب در این دایالوگ به این صورت است که ابتدا باید مدرسه یعنی همون دکمه وسط انتخاب شود. سپس مسیر رفت یا برگشت انتخاب شود و در نهایت شروع انتخاب شود. اما محدودیت هایی که دارد قبل از انتخاب مدرسه نباید مسیر رفت یا برگشت و همچنین شروع یا پایان انتخاب شود. قبل از انتخاب مدرسه و مسیر نمی توان شروع را انتخاب کرد . به محض اینکه شروع انتخاب شود دیگر نمی شود چیزی را انتخاب کرد تا وقتی پایان زده شود. چنانچه بعد از انتخاب مدرسه و مسیر دوباره مدرسه را انتخاب کند کلا همه انتخاب از اول شروع می شود. با انتخاب شروع تمام فیلد های share preference آپدیت می شود. هنگامی که مسیر رفت انتخاب شود مسیر برگشت در صورت فعال بودن غیر فعال می شود. بعد از انتخاب پایان تمام فلگ های کنترلی و مقادیر share preference به حالت اولیه بر می گردد. در هر انتخاب رنگ باتن ها تغییر می کند. هنگامی که دایالوگ باز می شود تمام باتن ها بر اساس فیلد های ذخیره شده انتخاب و رنگ مورد نظر را می گیرند. این محدودیت ها باعث شده این قسمت کمی پیچیده شود. که به ترتیب در مورد هر if پایین تر توضیح می دهیم

```
final SelectDialog selectDialog = new SelectDialog(  
    context: MainActivity.this,  
    MyApplication.totalSchool.get(0).getOrganization().getDescript(),  
    MyApplication.totalSchool.get(0).getOrganization().getCode());  
  
selectDialog.show();  
selectDialog.layoutSchool1.setVisibility(View.VISIBLE);  
selectDialog.layoutSchool2.setVisibility(View.GONE);  
selectDialog.layoutSchool3.setVisibility(View.GONE);
```

در این قسمت ما دایالوگ را show می کنیم. و اسم مدرسه که قبلا در MyApplication داشتیم را میفرستیم تا در باتن وسط نمایش دهد و همچنین کد مدرسه را هم میفرستیم تا در tag باتن قرار دهد بعدا برای مقایسه با share prefrence نیاز می شود. در این قسمت حداکثر سه تا سرویس می تواند وجود داشته باشد ولی چون ما همون اول بررسی کردیم که یک سرویس بیشتر نداریم بقیه لایه ها را gone میکنیم.

```
if (userManager.getSchoolCode().equals(selectDialog.btnSchoolName1.getTag())) {
    selectDialog.btnSchoolName1.setBackgroundResource(R.drawable.selector1);
    selectDialog.stateBtnName = 1;
} else if (userManager.getSchoolCode().equals("-1")) {
    selectDialog.btnSchoolName1.setBackgroundResource(R.drawable.selector);
    selectDialog.stateBtnName = 0;
}

if (userManager.getStateReturn().equals("1")) {
    selectDialog.btnGo1.setBackgroundResource(R.drawable.selector);
    selectDialog.stateGO = 0;
    selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector1);
    selectDialog.stateReturn = 1;
} else {
    selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector);
    selectDialog.stateReturn = 0;
}

if (userManager.getStateGo().equals("1")) {
    selectDialog.btnGo1.setBackgroundResource(R.drawable.selector1);
    selectDialog.stateGO = 1;
    selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector);
    selectDialog.stateReturn = 0;
} else {
    selectDialog.btnGo1.setBackgroundResource(R.drawable.selector);
    selectDialog.stateGO = 0;
}
```

در اینجا با فرض اینکه کاربر برنامه را بسته و دوباره باز کرده اطلاعات باید از share prefrence خوانده بشه if اول بررسی میکند که کدام مدرسه و کدام باتن بزرگ که در صفحه هست باید انتخاب شود و فلگ مربوط را مقدار می دهد و بکگراند باتن را هم رنگی می کند

if دوم بررسی می کند که آیا در حالت برگشت بود که از برنامه خارج شد یا حالت رفت و با توجه به اون باتن رفت را رنگی و فلگ مربوطه را ست می کند

if سوم وضعیت رفت را چک می کند و در صورت انتخاب بوده و از برنامه خارج شدیم دوباره به همان وضعیت بر می گرداند

```
if (userManager.getStateStart().equals("1")) {
    selectDialog.btnStart1.setBackgroundResource(R.drawable.selector1);
    selectDialog.stateStart = 1;
}
```

این if که خیلی مهم هست بررسی میکند که کاربر دکمه خروج را زده است و بعد از برنامه خارج شده یا نه توجه کنید که همه فلگ ها که به صورت local هستند در صورت زدن دکمه شروع در share prefrence ذخیره می شوند


```

selectDialog.btnSchoolName1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (userManager.getStateStart().equals("1")) {
        } else {
            if (selectDialog.stateBtnName == 0) {
                //Log.i("BAC", ""+selectDialog.btnSchoolName1.getBackground());
                selectDialog.btnSchoolName1.setBackgroundResource(R.drawable.selector1);
                Toast.makeText(context: MainActivity.this, text: "برگشت را انتخاب و گزینه شروع را بزنید",
                selectDialog.stateBtnName = 1;
            } else {
                selectDialog.btnSchoolName1.setBackgroundResource(R.drawable.selector);
                selectDialog.btnStart1.setBackgroundResource(R.drawable.selector);
                selectDialog.btnEnd1.setBackgroundResource(R.drawable.selector);
                selectDialog.btnGo1.setBackgroundResource(R.drawable.selector);
                selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector);
                // Toast.makeText(MainActivity.this, "برگشت را انتخاب و گزینه شروع را بزنید",
                selectDialog.stateBtnName = 0;
                selectDialog.stateReturn = 0;
                selectDialog.stateGO = 0;
            }
        }
    }
});

```

این click listener هست که برای باتن بزرگ که وسط دایالوگ هست و اسم مدرسه روی آن هست نوشته شده است. در صورت انتخاب این دکمه اگر شروع انتخاب نشده باشد چون اگر شروع انتخاب شده باشد تا وقتی بایان زده نشده است نمی توان تغییری داد برای همین با یک if همون اول کار این مورد بررسی می شود در غیر اینصورت اگر قبلا باتن انتخاب نشده باشد انتخاب می شود و رنگش تغییر می کند و فلگ مربوط به این که stateBtnName هست ست می شود و پیغامی جهت اینکه راهنمایی داده می شود مبنی بر اینکه شما باید مسیر رفت و برگشت را انتخاب و سپس شروع را بزنید. اگر این باتن قبلا انتخاب شده بود و دوباره روش کلیک شد تمام باتن های رفت و برگشت و شروع و بایان و state های آن ها به حالت عادی بر میگردد انگار دایالوگ دوباره شروع شده است.

```

selectDialog.btnGo1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (userManager.getStateStart().equals("1")) {
        } else {
            if (selectDialog.stateBtnName == 0) {
                Toast.makeText(context: MainActivity.this, text: "لطفا ابتدا مدرسه را انتخاب کنید",
            } else {
                if (selectDialog.stateGO == 0) {
                    selectDialog.btnGo1.setBackgroundResource(R.drawable.selector1);
                    selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector);
                    selectDialog.stateReturn = 0;
                    selectDialog.stateGO = 1;
                } else {
                    selectDialog.btnGo1.setBackgroundResource(R.drawable.selector);
                    selectDialog.stateGO = 0;
                }
            }
        }
    }
});

```

حال باید کاربر باتن های رفت یا برگشت را انتخاب کند ابتدا مثل دفعه قبل بررسی می شود که شروع انتخاب نشده باشد. چون اگر شروع انتخاب شده باشد دیگر نمیتوان باتن ها را انتخاب کرد و باید تا زدن پایان صبر کرد با فرض اینکه شروع انتخاب نشده ادامه می دهیم اینجا بررسی می شود که آیا ابتدا مدرسه انتخاب شده است یا نه اگر انتخاب نبود پیامی مبنی بر اینکه ابتدا مدرسه را انتخاب کنید می دهد و اگر مدرسه انتخاب شده بود بررسی می شود که آیا دکمه رفتن از قبل انتخاب شده یا الان می خواهد انتخاب شود اگر از قبل انتخاب بوده با stateGo صفر و رنگش به حالت قبل برگردد و اگر از قبل انتخاب نبود باید برگشت انتخابش رو از دست بدهد و رفت رنگش تغییر و stateGO هم یک شود و stateReturn صفر شود

```
selectDialog.btnReturn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (userManager.getStateStart().equals("1")) {
        } else {
            if (selectDialog.stateBtnName == 0) {
                Toast.makeText(context: MainActivity.this, text: "لطفا ابتدا مدرسه را انتخاب کنید",
            } else {
                if (selectDialog.stateReturn == 0) {
                    selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector1);
                    selectDialog.btnGo1.setBackgroundResource(R.drawable.selector);
                    selectDialog.stateGO = 0;
                    selectDialog.stateReturn = 1;
                } else {
                    selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector);
                    selectDialog.stateReturn = 0;
                }
            }
        }
    }
});
```

در اینجا باتن برگشت هست که شبیه به کلیک باتن رفت می باشد .

```
selectDialog.btnStart1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (userManager.getStateStart().equals("1")) {
        } else {
            if (selectDialog.stateBtnName == 0) {
                Toast.makeText(context: MainActivity.this, text: "لطفا ابتدا مدرسه را انتخاب کنید", Toast.LENGTH_SHORT).s
            } else {
                if (selectDialog.stateReturn == 0 && selectDialog.stateGO == 0) {
                    Toast.makeText(context: MainActivity.this, text: "لطفا مسیر رفت یا برگشت را انتخاب کنید", Toast.LENGTH_
                } else {
                    if (selectDialog.stateReturn == 1) {
                        selectDialog.stateStart = 1;
                        userManager.setStateReturn("1");
                        userManager.setStateGo("0");
                        userManager.setStateHeader("مسیر برگشت به سمت خانه شروع شده است");
                        userManager.setStateStart("1");
                        userManager.setSchoolCode(MyApplication.totalSchool.get(0).getOrganization().getCode());
                        selectDialog.dismiss();
                        txtCurrentStat.setText("مسیر برگشت به سمت خانه شروع شده است");
                        Toast.makeText(context: MainActivity.this, text: "مسیر برگشت به سمت خانه شروع شده است", Toast.LENGTH_
                    } else {
                        selectDialog.stateStart = 1;
                        userManager.setStateReturn("0");
                        userManager.setStateGo("1");
                        userManager.setStateHeader("مسیر رفت به سمت مدرسه شروع شده است");
                        userManager.setStateStart("1");
                        userManager.setSchoolCode(MyApplication.totalSchool.get(0).getOrganization().getCode());
                        selectDialog.dismiss();
                        txtCurrentStat.setText("مسیر رفت به سمت مدرسه شروع شده است");
                    }
                }
            }
        }
    }
});
```

این رویداد انتخاب دکمه شروع می باشد که خیلی مهم هست ابتدا بررسی می شود که در حالتی باشد که قبلا شروع انتخاب نشده باشد. سپس ابتدا چک می کند تا باتن مدرسه انتخاب شده باشد در غیر اینصورت پیام مناسب می دهد سپس چک می کند دکمه رفت و برگشت انتخاب شده باشد و در غیر اینصورت پیام مناسب سپس چک میکند اگر رفت انتخاب شده بود stateStart را برابر یک میگذارد و شروع میکند به ثبت تغییرات در share prefrence و دایالوگ را dismiss میکند و نمایش مناسب در قسمت بالای صفحه اصلی می دهد اگر رفت انتخاب شده باشد متناسب با حالت رفت share prefrence را ثبت می کند و دایالوگ را میبندد.

```
if (userManager.getStateStart().equals("1")) {
    selectDialog.btnEnd1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            selectDialog.stateBtnName = 0;
            selectDialog.stateGO = 0;
            selectDialog.stateReturn = 0;
            selectDialog.stateStart = 0;
            selectDialog.stateEnd = 0;
            selectDialog.tagCode = "-0";

            userManager.setStateHeader("سفری آغاز نشده است");
            userManager.setSchoolCode("-1");
            userManager.setStateReturn("0");
            userManager.setStateGo("0");
            userManager.setStateStart("0");

            txtCurrentStat.setText("سفری آغاز نشده است");

            selectDialog.btnGo1.setBackgroundResource(R.drawable.selector);
            selectDialog.btnReturn1.setBackgroundResource(R.drawable.selector);
            selectDialog.btnStart1.setBackgroundResource(R.drawable.selector);
            selectDialog.btnEnd1.setBackgroundResource(R.drawable.selector);
            selectDialog.btnSchoolName1.setBackgroundResource(R.drawable.selector);

        }
    });
}
```

اما با انتخاب گزینه پایان و در حالتی که شروع انتخاب شده باشد همه فلگ ها و همه فیلد های share prefrence به حالت اولیه و تمام باتن ها رنگ اولیه را به خود می گیرند.