

FMWAY IŞIK

Object Design Document

Version 1.0

15.05.2020

Ersin Çebi
Emre Can Rua
Umutcan Neşet Ölçer
Çağatay Demircan

Prepared for
SOFT3102



IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

Table of Contents

1.	Introduction.....	3
1.1	Object Design Trade-offs.....	3
1.2	Interface Documentation Guidelines.....	4
1.3	Definitions, Acronyms, and Abbreviations.....	4
2.	Packages.....	5
2.1	Models.....	5
2.2	Operations.....	5
2.3	Libs.....	6
2.4	Layout.....	7
2.5	Menu.....	8
3.	Class Interfaces.....	9
3.1	Class Diagram.....	9
3.2	Database Diagram.....	10
3.3	Method Explanations.....	10
4.	References.....	20

OBJECT DESIGN DOCUMENT

1. Introduction

This Object Design Document (ODD) defines the object-level design of the app to be developed. It is based on the initial concept of the MVVM architecture, proposed in Android App development strategy.

1.1 Object Design Trade-offs

FMWAY Işık is a transportation application exclusively for Işık University students. In FMWAY Işık, there are several trade-offs which editing trip as admin, adding/editing trip as driver, sign up to use the functions, rating and upload issues to admins.

Drivers can add their own trip and save it to the system. However, when an edition on a trip is done by admin, it is automatically approved by system and started to show in trip list without any approval. So, we can say that, admin has the over control on everything in the trip system. Users can add a trip, but admins can alter or even delete it without ask the users.

Drivers also can add or edit their own trips without any control. They can define their trip information, and simply add or delete them with a click. However, in case of report on a trip information by any user, the trip will be deleted immediately by admins and that user may be banned from the system.

In the FMWAY Işık system, signing up to the system is a must for joining a trip. All users are signed up as passengers in the beginning and can only join trips, not create one. The users who upload their license and get approved by the system, are defined as drivers, who can both join and create trips

Another trade-off is rating. Every user can rate their experience on the trip with the other person, such as driver can rate the passenger as “+” or “-“, and the passenger can rate the driver in an numeric rate system.

Uploading issue is another trade-off of FMWAY Işık system. Every registered user can send their issue to the system admins to tell his/her problems. Every admin can check report messages.

Other trade-offs we have made are:

- Buy vs Build : Our software project is built in android studio, and we are coding and developing the code together with our team.
- Memory space vs response time : Response time focused to be not take a very long time, and more memory space can be used for faster response time.
- Internet Connection Required: The project need internet connection in order to complete the application task such as login, add-delete-update information on tables and database.

1.2 Interface Documentation Guidelines

There are a list of rules we have applied in our developer team so we can understand and develop the code written (individually) together easier, and make it more readable.

These rules are:

- Buttons are labeled as “<ActivityName>Button” for onClick events
- Classes are named with function name for instance; AddTripActivity, SignUpActivity etc.
- Methods are named with verb phrases, fields, and parameters with noun phrases.
- Start and End of certain functions are shown in comment lines.
- Since we worked with Parse, every pushes and posts to Parse Server are came with some callback functions including exceptions.
- The functions and the classes will be written based on object oriented development, not hard-coding.

Login page is the main page for all of the users. After the login, with related to user roles, different kind of main pages are served to the users. With respect to user roles, different kind of functions will be served to the users, such as, add trip for driver, join trip for passenger and so on.

FMWAY Işık has simple log-in and sign-up views. In Sign-in page, system asks to new users to enter the needed information. In login page, there are just e-mail and password field.

1.3 Definitions, Acronyms, and Abbreviations

MVVM- Model View ViewModel

Admin: Actor of FMWay Işık System.

Database: The collection of large data set/stack.

Feedback: The answer of the system for the operation.

GitHub: A web-based hosting service, mostly used for computer code.

Passenger: Actor of FMWay-Işık System.

Driver: Actor of FMWay-Işık System.

Trips: The specific travels that drivers create and passengers attend to.

Parse Server: Open source Backend-as-a-Service(BaaS) framework initially developed by Facebook.

Java 8: A programming language.

Template: A guide of making patterns.

Android Studio: An Integrated Development Environment (IDE) developed by IntelliJ.

2. Packages

FMWAY Işık is designed and implemented as an Android application. There are a lot of Android system packages, but there are 5 designed and implemented packages by us which are models, operations, libs, layout and menu.

Our project classes include Java,XML,Parse Query commands. XML is used for creating and designing the “view” part of the application. Java classes and activities are used for developing the “UI”, which is the “controller” side of this application. Java Android Library is used on the application development. The “model” side is the database, which is imported in the project, “build.gradle Scripts”. PARSE server database system is used for the project and the tables of the project exists on the PARSE Server “www.back4app.com”.

2.1 Models

In this package, the system object classes are stored.

- Chat.java: Chat model
- Trip.java: Trip model
- User.java: User models(as admin, passenger, driver and upload)

2.2 Operations

In this package, all operations of the application are stored.

For Admin:

- AddAdminActivity.java: Admin can add another admin with this method, since no one can be signed up as admin.
- AddDriverActivity.java: Admin can add a user whose role is driver. This method is a backup plan, the main driver add method is to approveToBeDriverActivity.
- AdminActivity: Admin main page activity which shows the main functions of the admin.
- DriverLicenseListActivity: This method shows the uploaded licenses from the drivers which are stored in the database.
- EditTripAdminActivity: Admin can edit all trips that drivers created.
- IssueListActivity: This method shows the uploaded issues from the users which are stored in the database.
- ListTripActivityAdmin: This method shows all the active trips that is saved in the database.
- ListUserActivityAdmin: This method shows all users that is saved in the database.
- PostActivityIssueList: This method is the adapter of the issues.
- PostActivityLicenseList: This method is the adapter of the licenses.

- TripDetailsAdminActivity: This method shows the clicked trip's information and provides edit or delete options.
- UserDetailsAdminActivity: This method shows the clicked user's information.

For Driver:

- AddTripActivity: Driver can add trip to the system.
- DriverActivity: Driver main page activity which shows the main functions of the driver.
- EditTripDriverActivity: Driver can alter his/her trip information.
- ListMyTripsDriverActivity: This method shows the active trips that driver created.
- TripDetailsDriverActivity: This method shows the clicked trip's information and provides edit or delete options.

For Passenger

- driverFeedback: Passenger can rate the driver.
- ListTripActivityPassenger: This method shows all the active trips that is saved in the database.
- PassengerActivity: Passenger main page activity which shows the main functions of the passenger.
- TripDetailsPassengerActivity: This method shows the clicked trip's information and provides join option.
- UploadActivity: Passenger can upload his/her issue to notify the admin.
- UploadLicenceActivity: Passenger can upload his/her license to apply to be driver.

CommonActivities:

- ForgotPasswordActivity: User can change its password via e-mail.
- ParseStarterClass: Database connection.
- Payment: Deposit money to the owned account.
- Profile: This method shows profile information.
- SignUpActivityMain: Sign up method.
- SignUpLoginActivity: Login method.
- tripChat: Chat method for the participants of the specific trip.
- tripPage: Joined trip method.

2.3 Libs

In this package, common variables are defined

- Common.Java

2.4 Layout

All layouts that are used by activities are stored in this package.

- activity_addadmin.xml
- activity_adddriver.xml
- activity_admin.xml
- activity_driver.xml
- activity_driverlicenselist.xml
- activity_edittrip.xml
- activity_home_page.xml
- activity_passenger.xml
- activity_payment.xml
- activity_profile.xml
- activity_trip_chat.xml
- activity_trip_page.xml
- activity_tripdetailsadmin.xml
- activity_tripdetailspassenger.xml
- activity_userdetailsadmin.xml
- addtrip-activity.xml
- chat_customview.xml
- customview_driverlicence.xml
- customview_listusers.xml
- driver_feedback.xml
- forgot_password_activity.xml
- listmytrips_activity.xml
- listtrips_activity.xml
- listuser_activity.xml
- signup_activity.main.xml
- signup_login_activity.xml
- upload_activity.xml
- upload_licence_activity.xml

2.5 Menu

This package contains all menu layouts for all user types.

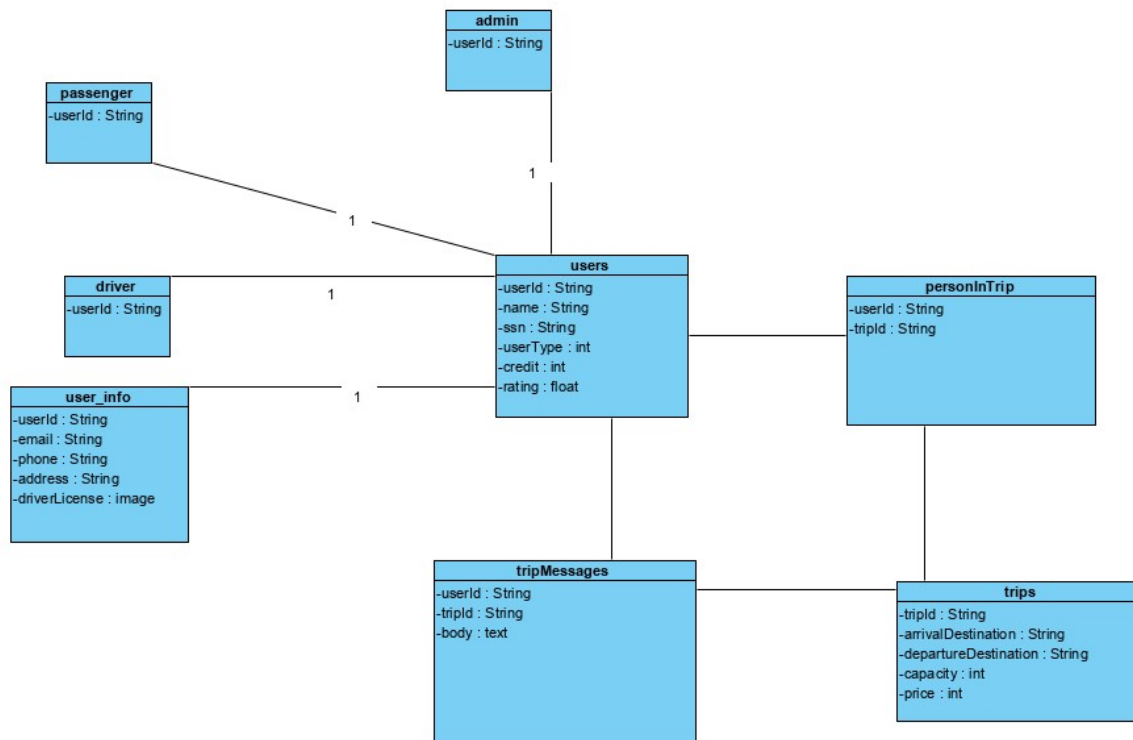
- menu_admin.xml: Admin's menu layout.
- menu_not_logged_in.xml: Menu layout for users' that not logged in.
- menu_user.xml: Menu layout for all registered and logged in users.

3. Class Interfaces

3.1 Class Diagram



3.2 Database Diagram



3.3 JavaDoc Export

Class TripDetailsAdminActivity

Modifier and Type

Method and Description

void	download() searches for the trip details on database and fills the fields
protected void	onCreate (android.os.Bundle savedInstanceState) admin trip detail page activity constructor also makes button handling of edit and delete trip
boolean	onCreateOptionsMenu (android.view.Menu menu) menu option creator handler
boolean	onOptionsItemSelected (android.view.MenuItem item) logout button activity

Class UserDetailsAdminActivity

void [banUser](#)(com.parse.ParseUser object)
this class helps to ban a user on system its created separately because for testing

protected void [onCreate](#)(android.os.Bundle savedInstanceState)
admin user detail page activity constructor also makes button handling for blocking users

boolean [onCreateOptionsMenu](#)(android.view.Menu menu)

boolean [onOptionsItemSelected](#)(android.view.MenuItem item)
logout button activity

Class ForgotPasswordActivity

protected void [onCreate](#)(android.os.Bundle savedInstanceState)
activity constructor

Class listMyTrips

void [getUserTripList](#)()
searches for the trip details on database and fills the fields

void [listUserLists](#)(java.lang.String tripId)
takes each trip id that user joined and add int list

protected void [onCreate](#)(android.os.Bundle savedInstanceState)
persons trip listing page activity constructor also takes userID by intent and makes button handling for opening required trip

boolean [onCreateOptionsMenu](#)(android.view.Menu menu)
menu option creator handler

protected void [onNewIntent](#)(android.content.Intent intent)

boolean [onOptionsItemSelected](#)(android.view.MenuItem item)
logout button activity

Class ParseStarterClass

void [onCreate](#)()
activity constructor initializes the parse for all app

Class ParseStarterClass

void [onCreate](#)()
activity constructor initializes the parse for all app

Class Payment

- protected void [onCreate](#)(android.os.Bundle savedInstanceState)
activity constructor also takes the balance value from the database and sets on the screen for user
- void [pay](#)()
button handling for user balance according to filled fields takes the amount of money and loads the users credit for trips payments also makes error handling it self

Class Profile

- void [balance](#)(android.view.View view)
button action for opening payment activity
- void [calculateRating](#)(java.lang.String tripID)
find and calculate all the rating
- void [getUser](#)()
Fills users info into page
- void [getUserRating](#)()
find all trips created by users
- void [myJoinedTrips](#)(android.view.View view)
button action for opening payment activity
- protected void [onCreate](#)(android.os.Bundle savedInstanceState)
- boolean [onCreateOptionsMenu](#)(android.view.Menu menu)
menu option creator handler
- boolean [onOptionsItemSelected](#)(android.view.MenuItem item)
logout button activity
- void [openSettings](#)(android.view.View view)
button action for opening settings activity

Class profileSettings

void	<u>chooseImage</u> (android.view.View view) open file choose file dialog
void	<u>editUser</u> (com.parse.ParseObject object, com.parse.ParseFile parseFile, java.lang.String name, java.lang.String surname, java.lang.String phone) this class helps to edit profile editing on database its created separately because for testing
void	<u>getUser</u> () Fills users info into page
protected void	<u>onActivityResult</u> (int requestCode, int resultCode, android.content.Intent data)
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) persons trip listing page activity constructor also takes userID by intent and makes button handling for opening required trip
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
protected void	<u>onNewIntent</u> (android.content.Intent intent)
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity
void	<u>updateProfile</u> (android.view.View view) when activated takes filled form info and updates the database for desired user

Class SignUpActivityMain

protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) activity constructor
void	<u>signUp</u> (android.view.View view) records the new user to the database according to given information in the fields and redirects to the activity according to user role
void	<u>userSignUp</u> (com.parse.ParseUser user, java.lang.String name, java.lang.String surname, java.lang.String email, java.lang.String phone, java.lang.String username, java.lang.String password) this class helps to create user on database its created separately because for testing every user is given passenger role as default

Class SignUpLoginActivity

void	<u>forgotPassword</u> (android.view.View view) button handler for password save for user
void	<u>login</u> (android.view.View view) login direction handler after user logs in their info finds the user role and redirect required page activity
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) activity constructor
void	<u>signUp_orientation</u> (android.view.View view) button handler for sign up user

Class tripChat

void	<u>listMessages</u> (java.lang.String takeTripId) message listing handler for the trip chat
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) activity constructor
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity
void	<u>saveMessageToDatabase</u> (com.parse.ParseObject parseObject, java.lang.String messageFromUser)
void	<u>sendMessage</u> (android.view.View view) button handler for sending entered message

Class tripPage

void	<u>deleteFromTripArray()</u> searches trip by tripID than deletes the userID from Passenger array
void	<u>deleteFromUserArray()</u> searches user by userID than deletes the tripID from Trip array
int	<u>findIndexOf</u> (java.lang.String item) takes the search key returns the index of item
void	<u>getTrip()</u> Fills trips info into page
void	<u>leaveTrip</u> (android.view.View view) allow people to leave trip if the trips date haven't currently started
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) admin trip detail page activity constructor also makes button handling for opening trip edit activity and trip delete
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity
void	<u>rateTheTrip</u> (android.view.View view) redirects to chat page of trip
void	<u>sendMessage</u> (android.view.View view) redirects to chat page of trip
void	<u>setJSONArrayToNull()</u>

Class AddTripActivity

void	<u>addTrip</u> (android.view.View view) button handler for adding trip info entered by driver also makes error e-handling for entered info
void	<u>addTripToDb</u> (com.parse.ParseObject object, java.lang.String date, java.lang.String time, java.lang.String from, java.lang.String destination, int capacity, int price, java.lang.String uid) this class helps to create trip creation on database its created separately because for testing
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) driver add trip page activity constructor also takes userID by intent and makes button handling date and time selectors Defines from and destination places as spinner
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity

Class DriverActivity

void	<u>addTrip</u> (android.view.View view) button action for opening add trip activity
void	<u>listMyTrips</u> (android.view.View view) button action for opening list created by owned trip activity
void	<u>listTrips</u> (android.view.View view) button action for opening list trip activity
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) driver activity page activity constructor also takes userID by intent
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity
void	<u>openProfile</u> (android.view.View view) button action for opening driver profile page

Class EditTripDriverActivity

void	<u>download()</u> searches for the trip details on database and fills the fields
void	<u>editTrip</u> (android.view.View view) button handling for submitting the changes on fields for trip
void	<u>editTripOnDb</u> (com.parse.ParseObject object, java.lang.String date, java.lang.String time, java.lang.String from, java.lang.String destination, int capacity, int price) this class helps to edit trip editing on database its created separately because for testing
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) driver edit trip page activity constructor also takes userID and objectID by intent and makes button handling date and time selector
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity

Class ListMyTripsDriverActivity

void	<u>download()</u> searches for the trip details on database and fills the fields
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) driver trip listing page activity constructor also takes userID by intent and makes button handling for opening required trip
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
protected void	<u>onNewIntent</u> (android.content.Intent intent)
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity

Class TripDetailsDriverActivity

void	<u>download()</u> searches for the trip details on database and fills the fields
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) admin trip detail page activity constructor also makes button handling for opening trip edit activity and trip delete
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity

Class driverFeedback

void	<u>addRatingToDb</u> (com.parse.ParseObject object, float qualityValue, java.lang.String comment) this class helps to create rating on a trip on database its created separately because for testing
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) before call this activity this activity requires two parameter via intent
void	<u>sendFeedback</u> (android.view.View view)

Class ListTripActivityPassenger

void	<u>download()</u> searches for the trip details on database and fills the fields
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) passenger trip listing page activity constructor
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
protected void	<u>onNewIntent</u> (android.content.Intent intent)
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity

Class PassengerActivity

void	<u>applyDriver</u> (android.view.View view) button action for opening driver application page
void	<u>listTrips</u> (android.view.View view) button action for opening list trip activity
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) passenger add trip page activity constructor also takes userID by intent
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity
void	<u>openProfile</u> (android.view.View view) button action for opening driver profile page
void	<u>uploadYourIssue</u> (android.view.View view) button action for opening support page

Class TripDetailsPassengerActivity

void	<u>download</u> () searches for the trip details on database and fills the fields
void	<u>joinTrip</u> ()
protected void	<u>onCreate</u> (android.os.Bundle savedInstanceState) passenger add trip page activity constructor also takes objectID by intent
boolean	<u>onCreateOptionsMenu</u> (android.view.Menu menu) menu option creator handler
boolean	<u>onOptionsItemSelected</u> (android.view.MenuItem item) logout button activity
boolean	<u>userCheck</u> ()

Class UploadActivity

void [chooseImage](#)(android.view.View view)
open file choose file dialog

protected void [onActivityResult](#)(int requestCode, int resultCode, android.content.Intent data)

protected void [onCreate](#)(android.os.Bundle savedInstanceState)
passenger issue page activity constructor

void [upload](#)(android.view.View view)
button handling for photo and send issue

Class UploadLicenceActivity

void [chooseImage](#)(android.view.View view)

protected void [onActivityResult](#)(int requestCode, int resultCode, android.content.Intent data)

protected void [onCreate](#)(android.os.Bundle savedInstanceState)
passenger license upload page activity constructor

void [upload](#)(android.view.View view)
button handling for license photo

4. References

1. Bruegge B. & Dutoit A.H.. (2010). Object-Oriented Software Engineering Using UML, Patterns, and Java, Prentice Hall, 3rd ed.