FMWAY IŞIK

Test Design Document

Version 1.0

15.05.2020

Ersin Çebi

Emre Can Rua

Umutcan Neşet Ölçer

Çağatay Demircan

Prepared for

SOFT3102



**Table of Contents**

[1. Introduction 3](#_Toc515931331)

[1.1 Objectives 3](#_Toc515931332)

[1.2 Testing Strategy 3](#_Toc515931333)

[1.3 Scope 4](#_Toc515931334)

[2. Test 4](#_Toc515931335)

[2.1 Modules 4](#_Toc515931336)

[2.2 Control Procedures 4](#_Toc515931337)

[2.3 Feature To Be Tested 5](#_Toc515931338)

[2.4 Feature Not To Be Tested 5](#_Toc515931339)

2.5 Test Explanations……………………………………………………………5

[3. References 7](#_Toc515931346)

**TEST DESIGN DOCUMENT**

# Introduction

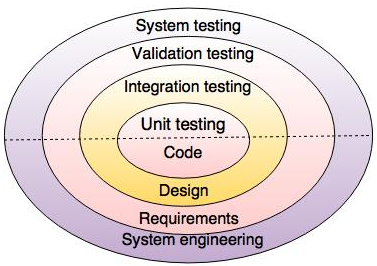
This Test Design Document defines the test that applied on the app develop process . It is based on the initial concept of the MVVM architecture, proposed in Android App development strategy.

## Objectives

* [Finding defects](http://tryqa.com/what-is-defect-or-bugs-or-faults-in-software-testing/) which may get created by the programmer while developing the software.
* Gaining confidence in and providing information about the level of [quality](http://tryqa.com/what-is-software-quality/).
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product.

## Testing Strategy

A strategy of software testing is shown in the context of spiral.



**Unit testing:**  
 Unit testing starts at the centre and each unit is implemented in source code.  
  
**Integration testing:**  
 An integration testing focuses on the construction and design of the software.  
  
**Validation testing:**  
 Check all the requirements like functional, behavioral and performance requirement are validate against the construction software.  
  
**System testing:**  
 System testing confirms all system elements and performance are tested entirely.

Until now we made unit testing for all methods in project.

## Scope

Software Testing is an investigation conducted to provide stakeholders with information about the quality of a particular product or service under test. In other words, software testing is a process of verification and validation. So, we try to prevent any feature defect in project for consumer good. In our testing we test every unit testable component but of course, repair and maintenance of application software is never ending cycle.

# Test

Like we mentioned before in testing phases we firstly use unit testing on components, we haven’t started integration testing. We only test the intractive parts of application, we did not test the design part of the app, so we are all developers at the end.

## Modules

For Module Testing, designing a [Test Case](https://www.guru99.com/test-case.html) is an important segment. While designing test cases for a module test, a tester has to take two things into consideration.

* Specification for the module
* The module's source code
* Probability of identifying errors or bugs on smaller chunks of program becomes higher
* Multiple modules can be tested simultaneously and hence supports parallel testing
* Complexity of testing can be easily managed
* Non-incremental method- all modules are tested independently. First, it combines all modules and then test the whole program
* Incremental method- each module is tested first and then gradually incremented to the tested collection. It does a step wise retesting
* Incremental Testing, there are two approaches – Top down and Bottom Up testing
* To execute the module with the selected data, it requires a **driver** for supplying the test data, monitoring the execution and capturing the results

## Control Procedures

Control Procedures is a set of activities for ensuring quality in software products. Software Quality Control is limited to the Review/Testing phases of the [Software Development Life Cycle](http://softwaretestingfundamentals.com/software-development-life-cycle/) and the goal is to ensure that the products meet specifications/requirements.

* Requirement Review
* Design Review
* Code Review
* Deployment Plan Review
* Test Plan Review
* Test Cases Review

These specified procedures and outlined requirements lead to the idea of Verification and Validation and software testing. It also refers to the ability for software to perform well in unforeseeable scenarios and to keep a relatively low defect rate. We try to apply all of the steps above on our project.

## Feature To Be Tested

In testing part we unit test every model, a cording to their feature. First lets talk about the variable naming we did.

On testing class we name the parameter we send with parameter’#’ means if we send more than one parameter it goes like parameter1, parameter2.. and so on. After defining part we create a setup() function, in this function we declare the method that we want to test, if requares we send the parameters in it. Then we create a function called tearDown(), this function does what and how we named it. After testing we use this function the clear memory. The class declaration set to null in this function and it no longer usable after you call this function.

In test function firs we use name of that actual function we want test than add a ‘test’ than we add the expected outcome usually we expect true, For example let say the function we test is getTripId() by using that name getTripId\_Test\_expectParameter1() it s expects parameter1 because it is a class testing.

Inside the test function firs we call setup() than we define two variable ‘expected’ the type of the parameter could change a cording to requirement of class, than we define ‘actual’ this means the returning value. After here we chose the required function for testing it might be assertEquals(expect,actual); this takes two parameter and gives you whether they are equal or not or we might check if the expected is true by calling assertTrue(actual); than we call tearDown to terminate test.

The test design is up to tester you can create the test by asking the function ‘what it wants, what it wants to do?’.

For explaining the subject like we mentioned we test every software component that interacts with user by saying this; every insert, update, class methods.

## Feature Not To Be Tested

## We did not test the design parts, database listing functions.

## Test Explanations

* **commonsUnitTest**

In this test, we checked the common variable of definitions method, which is date.

* **chatParseDefinitionsTest**

In this test, we checked the variables of chatParseDefinitions object, which are classNameKey(String), objectIDKey(String), createdAtKey(String) ,tripIDKey(String), personIDKeyString) and messageKey(String).

* **ChatTest**

In this test, we controlled the function of chat object. The checked parameters are tripID(String), personID(String) and message(String).

* **TripParseDefinitionsTest**

In this test, we checked the variables of TripParseDefinitions object, which are classNameKey(String), objectIDKey(String), tripCreatedByKey(String), dateKey(String), timeKey(String), fromKey(String), destinationKey(String), capacityKey(String) and priceKey(String).

* **TripTest**

In this test, we controlled the function of trip object. The checked parameters are objectId(String), date(String), time(String), from(String), destination(String), capacity(int), price(int).

* **RatingParseDefinitionsTest**

In this test, we checked the variables of RatingParseDefinitions object, which are objectIDKey(String), createdAtKey(String), userIDKey(String), tripIDKey(String), commentKey(String) and ratingValueKey(String).

* **IssuesParseDefinitionsTest**

In this test, we checked the variables of IssuesParseDefinitions object, which are className(String), imageKey(String), uploadKey(String), filename(String) and userNameKey(String).

* **LicenseParseDefinitionsTest**

In this test, we checked the variables of LicenseParseDefinitions object, which are className(String), imageKey(String), uploadKey(String), filename(String) and userNameKey(String).

* **UserParseDefinitionsTest**

In this test, we check the variables of LicenseParseDefinitions object, which are objectIDKey(String), createdAtKey(String), nameKey(String), surnameKey(String), usernameKey(String), emailKey(String), phoneKey(String) and userTypeKey(String).

* **UserTypesTest**

In this test, we check the user’s type, which can be admin, passenger or driver. The controlled variable is userType(String).

* **AddAdminActivityTest**

In this test, we control the function of AddAdminActivity. The checked parameters are name(String), surname(String), email(String), phone(int), username(String), password(String).

* **AddDriverActivityTest**

In this test, we control the function of AddDriverActivity. The checked parameters are name(String), surname(String), email(String), phone(int), username(String), password(String).

* **EditTripAdminActivityTest**

In this test, we control the function of EditTripAdminActivity. The checked parameters are date(String), time(String), from(String), destination(String), capacity(int), price(int).

* **UserDetailsAdminActivityTest**

In this test, we control the selected user’s userId. The checked variable is userID(String).

* **SignUpActivityMainTest**

In this test, we control sign up function of signUpActivityMain. The checked parameters are name(String), surname(String), email(String), phone(int), username(String), password(String).

* **tripChatTest**

In this test, we checked the send message function of tripChat. The checked variables are specified trip object and messageText(String).

* **AddTripActivityTest**

In this test, we check the function of addTripActivity. The checked parameters are date(String), time(String), from(String), destination(String), capacity(int), price(int) and uid(String).

* **EditTripDriverActivityTest**

In this test, we check the function of editTripDriverActivity. The checked parameters are date(String), time(String), from(String), destination(String), capacity(int), price(int).

* **DriverFeedbackTest**

In this test, we check the function of driverFeedback. The checked parameters are rating(float), and comment(String).

# References

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