System design document for Gasqueue

Version: 2.0

Date 29-05-2016

Author :Kevin,Petros,Long,Eric

1 Introduction

1.1 Design goals

Creating loosely coupled components to be reused in other system and reuse for other platforms with different views. E.g model should be able to be reused, controller should be able to handle different views. The code should also be testable.

1.2 Definitions, acronyms and abbreviations

- Android - mobile operating system for smartphones
- Activity- class handling the view for every window in an Android application
- Bar - the user who handles the orders
- Customer - the user who makes an order
- ClientID - unique ID used to track the user of the application
- GUI - Graphical User Interface
- Product -the item a customer can order.
- Order- A list of products made by the customer.
- Queue- a queue based chronologically on the orders.
- Firebase - database platform

2 System design

2.1 Overview

The application uses an modified MVC model suited for the Android design principle. Instead of using the Activity class as the main controller, other controller class are made that connects the activity with the model and database. The reasoning behind the design is to keep the components from being dependent on the android Activity. This allows for change of platform. E.g removing the android view and implementing Swing.

2.1.1 Unique Identifiers
Every user will have a clientID based on the users smartphone Android ID. The clientID will be used to track the user. E.g tracking the order made by the customer through his/her clientID.

2.1.2 Event paths
User inputs will be handled by the following chain: View->Activity->Controller class->Model
The activities will also be able to retrieve information from the database through Adapters.

- All updates in model will be through setters
- Sending data to the database will be handled by the controllers
- Retrieving data from the database will be handled by event listeners

2.2 Software decomposition

2.2.1 General
Activity, contains all GUI related classes
Model, contains the OO-models
Controller, contains all controller classes
Service, contains all service related content. E.g api to access database

2.2.2 Decomposition into subsystems
A data base is used to store and transfer data between the customer and the host. Firebase is used as the database platform. Firebase will be implemented through following database interface.

//Database interface
```
public interface IDatabaseManager<T> {

   public void saveMap(String address, Map<String, Map<String, Integer>> map);

   public void saveStringList(String address, List<String> list);

   public void sendObject(String address, Object object);

   public T getReference();

   public void setValue(String address, Object object);

   public T createChildReference(String childReference);
```

```
    public boolean checkCode(String barCode);

}
```

2.2.3 Layering

See figure 1.

2.2.4 Dependency analysis

Dependencies in the figure below.

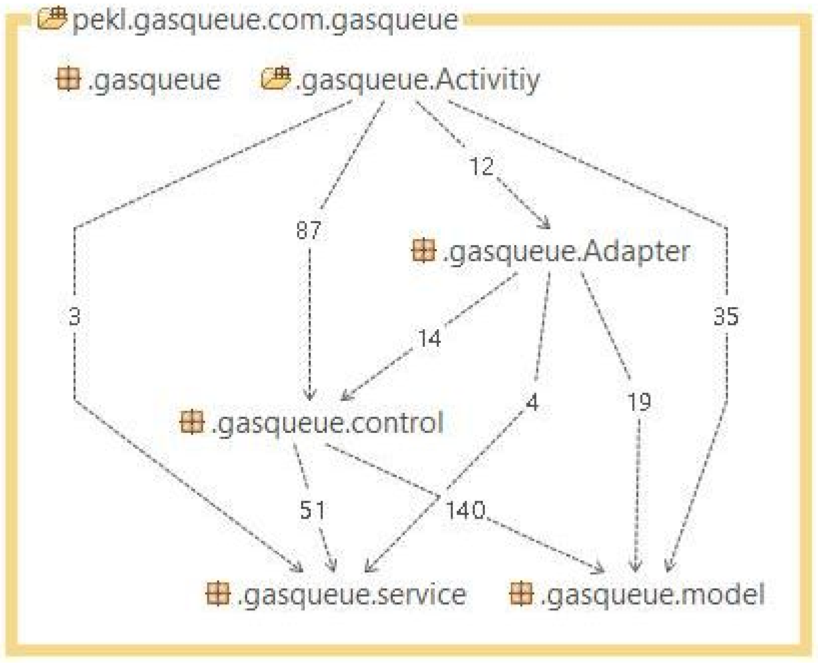
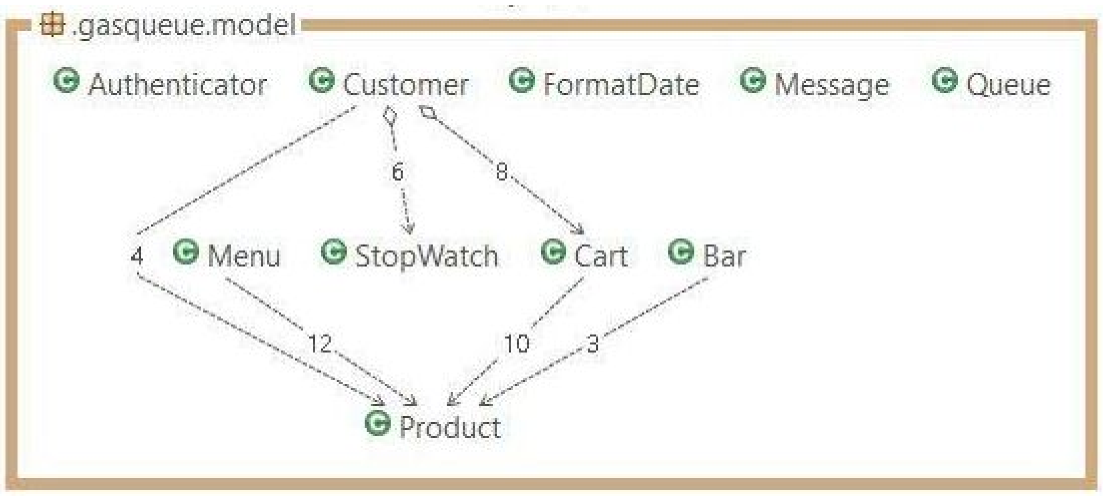


figure 1

- Activity, android specific classes that handles the view
- Control, use case controller classes
- Service, classes for database service
- Model, the model classes
- Adapter, android specific classes to update listviews.



2.3 Concurrency issues
Every user is notified every time the queue is updated.

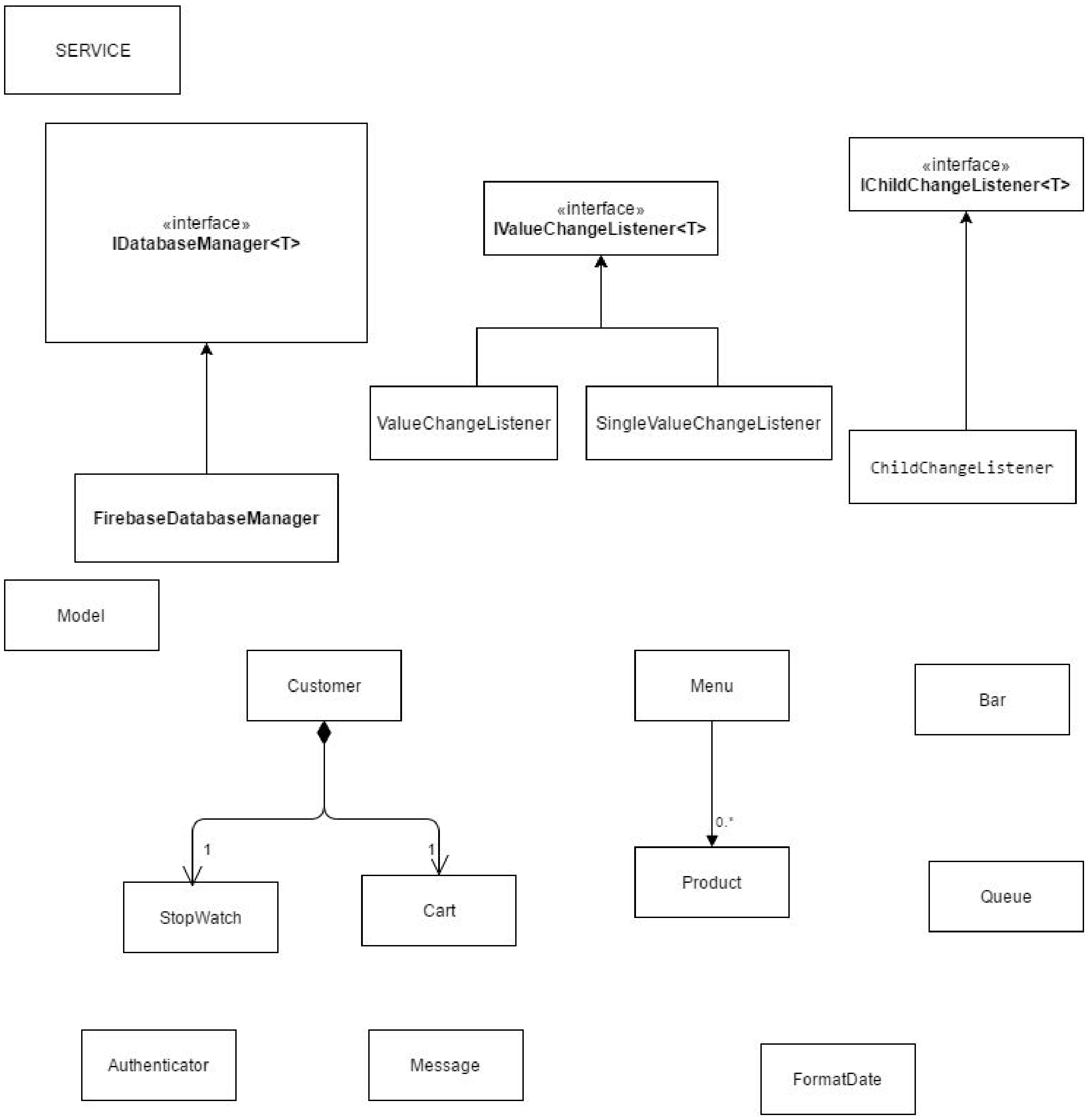
2.4 Persistent data management
All persistent data will be stored on Firebase. All data can be accessed and altered through the database API.

2.5 Access control and security
Client ID is based on the smartphone Android ID which can be manipulated by the user if using right tools.

2.6 Boundary conditions
The application can be run on a desktop using Android Studio IDE, which builds the application automatically using gradle. Otherwise it requires the Android Operating system to run.

3 References

APPENDIX
UML


SERVICE
«interface»
IDatabaseManager<T>
«interface»
IValueChangeListener<T>
«interface»
IChildChangeListener<T>
FirebaseDatabaseManager
ValueChangeListener
SingleValueChangeListener
ChildChangeListener
Model
Customer
Menu
Bar
1
1
0.*
StopWatch
Cart
Product
Queue
Authenticator
Message
FormatDate