Course: Product Development

Bachelor of Information and Communication

Click here to enter text.

April 6, 2017

Group P3

**TECHNICAL DOCUMENTATION**

– about mobile app E-market.

****

BACHELOR´S REPORT | ABSTRACT   
TURKU UNIVERSITY OF APPLIED SCIENCES

Click here to enter text.:

Instructor(s): Poppy Skarli

PHAN HONG DUC ,TRINH TRAN BINH, NGUYEN XUAN HUNG,

, THAPA AJIT BAHADUR, DO HIEU

\_

**ABSTRACT**:

This document gives the user a better understanding of the structure of this application, an application that aggregates information with early development time is information about the product used daily and the seller are the students who are studying(studied) around Turku .

The technical method of this application is to use XML to build a Layouts of App(frontend) and Java for building product features(backend) and using a database (PHP MySQL) on a server called APIs for small size of App .

KEYWORDS:

Technical documentation, UML, Use Case, product development, mobile app

contentS

[List of Abbreviations (OR) Symbols 5](#_Toc479334247)

[1 OVERVIEW 6](#_Toc479334248)

[1.1 Description of Organization Status 6](#_Toc479334249)

[1.1.1 Primitive description 6](#_Toc479334250)

[1.1.2 List of related subjects 7](#_Toc479334251)

[1.1.3 Application Development Problem 7](#_Toc479334252)

[1.1.4 Priority aspects 8](#_Toc479334253)

[1.2 Solution Description and Selection 8](#_Toc479334254)

[1.3 Method 10](#_Toc479334255)

[2 BUSINESS MODELING 13](#_Toc479334256)

[2.1 UML/Use case Diagram Explain about the Features of App 13](#_Toc479334257)

[2.2 Identify Business Worker and Business Entity 14](#_Toc479334258)

[2.2.1 Interaction with the app 14](#_Toc479334259)

[2.2.2 Server and database management 16](#_Toc479334260)

[2.2.3 Ads Management 18](#_Toc479334261)

[3 PROTOTYPE OF PRODUCT 20](#_Toc479334262)

[4 DATABASE ILLUSTRATION 24](#_Toc479334263)

[4.1 Database show case 24](#_Toc479334264)

[4.2 Main Code (database ) 28](#_Toc479334265)

[5 TEST CASE 32](#_Toc479334266)

[6 CONCLUSION 37](#_Toc479334267)

[References 38](#_Toc479334268)

appendices

None

pictures

[Picture 1. Greetings Screen 20](#_Toc479287842)

[Picture 2. Homepage interface 21](#_Toc479287843)

[Picture 3. Furniture catalog 22](#_Toc479287844)

[Picture 4. Checkout interface 23](#_Toc479287845)

figures

[Figure 1. Screen map of E-market app. 7](#_Toc479287833)

[Figure 2. Processes of retrieving database from server. 12](#_Toc479287834)

[Figure 3. Use case diagram showing relation between app and customer, admin 13](#_Toc479287835)

[Figure 4. Use case diagram showing how customer can use with the app. 14](#_Toc479287836)

[Figure 5. UML Interaction between each role 16](#_Toc479287837)

[Figure 6. Use case Server and Database Management 18](#_Toc479287838)

[Figure 7. Use case Ads Management 19](#_Toc479287839)

[Figure 8. Database version 1 24](#_Toc479287840)

[Figure 9. Database version 2. 25](#_Toc479287841)

tables

[Table 1. Subjects 7](#_Toc479287821)

[Table 2. Details and solutions for each problem 7](#_Toc479287822)

[Table 3. Application requirement 8](#_Toc479287823)

[Table 4. Binding application development 9](#_Toc479287824)

[Table 5. Solution selection 9](#_Toc479287825)

[Table 6. Interaction of Business workers 14](#_Toc479287826)

[Table 7. Management by Business Workers 16](#_Toc479287827)

[Table 8. Ads Management by Business Workers 18](#_Toc479287828)

[Table 9. Device Specific Test 32](#_Toc479287829)

[Table 10. Network Specific Check 33](#_Toc479287830)

[Table 11. User Interface Check 34](#_Toc479287831)

[Table 12. App Specific Check 35](#_Toc479287832)

# List of Abbreviations (OR) Symbols

None

# OVERVIEW

## **Description of Organization Status**

### **Primitive description**

Nowadays, the need of information gathering is raising in various aspects, such as education, politics and shopping information. Realizing the problem, an application whose purpose is to gather all information about purchases had come to life called e-Market. e-Market is a mobile application that uploads and stays updated almost everything related to daily demands such as clothes, household appliances, etc. to help users reach their need, and therefore can choose their best option.

* Selection
* Product searching
* Shopping cart (Collection of picked items)
* Display related products
* Language switching (English/Finnish/German/French, etc.)
* Notification (notice users when there’s a change on some items)

When the app is open, a stand-by screen is displayed. It will then check Internet connectivity and if the app cannot connect to the Internet, it notices the users to stay update with the newest information of items. Next, users will see the list of categorized items and can then proceed to detail information by clicking at the images of items. They will also see a fixed number of items that relates to the chosen item so that they can easily compare items. Users can pick the items in many ways, by choosing the images which are already divided into categories or by searching for the keyword of them. Finally, users click “+” to add items to the shopping cart, and when they do not wish an item, they can easily flick it away or click [ - ] to remove it from the shopping cart.

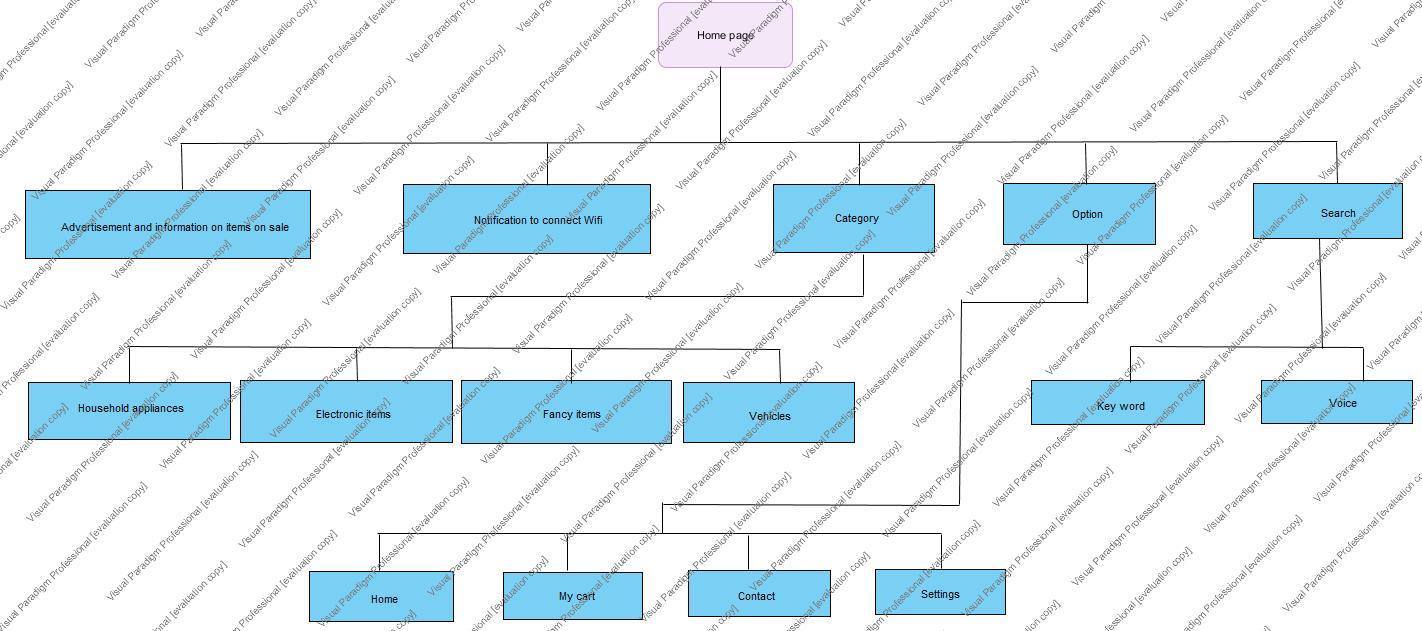


Figure 1. Screen map of E-market app.

### **List of related subjects**

Table 1. Subjects

|  |  |  |
| --- | --- | --- |
| Name | Representative | Role |
| Administrator | Mobile app | Manage the app |
| Provider | Supermarket, sellers | Selling |
| Customer | Anyone | Buying |
| Advertiser | Companies that want to advertise on the app | Provide ads and giving revenue |

### **Application Development Problem**

Table 2. Details and solutions for each problem

|  |  |
| --- | --- |
| Problem | Gather information and application development |
| Affected subject | Administrator |
| Effect of the problem | * Difficult in collecting data * Time-consuming * Difficult to advertise to customers. * Difficult to find sponsors. * Time is limited so the app may not perform perfectly. |
| Suggested solution | Limit the number of items stored on server |

### **Priority aspects**

*Nonpriority task*

The app is not required to:

* Distinguish between guess and member.
* Require user’s authentication.
* Support many languages (Only English will be used during the first stage).
* Support online payments.

*Priority task*

An application with basic functions:

* Search for items.
* See detail of items.
* Connect the application to API server.

## **Solution Description and Selection**

Table 3. Application requirement

|  |  |
| --- | --- |
| R1 | Can run on Android with basic interactions (scroll, back, home…) |
| R2 | Works with internet |
| R3 | Does not crash often |
| R4 | Has a static database |
| R5 | Has feedback feature |
| R6 | Does not crash except in extreme cases Only has minor bugs |
| R7 | Available on Google Play |
| R8 | Uses APIs Server for database |
| R9 | Has advance features |
| R10 | Available on App store, Window store |
| R11 | Implement a bot |

Absolute requirements: R1-R5

Essential requirements: R6-R8

Desire requirements: R9-R11

Table 4. Binding application development

|  |  |
| --- | --- |
| Y1 | Overall cost does not exceed 300€ |
| Y2 | Monthly cost does not exceed 20€ |
| Y3 | Development time is 4 months |

Table 5. Solution selection

|  |  |  |  |
| --- | --- | --- | --- |
|  | Plan A | Plan B | Plan C |
| R1 – R5 | Yes | Yes | Yes |
| R6 | No | Yes | Yes |
| R7 | No | Yes | Yes |
| R8 | No | No | Yes |
| R9 | No | No | Yes |
| R10 | No | No | Yes |
| R11 | No | No | No |
| Y1 | 50€ | 100€ | 200€ |
| Y2 | 0€ | 10€ | 25€ |
| Y3 | 2 months | 4 months | 7 months |

Based on the requirement, development bindings and our team situation, we decided to follow plan B.

## **Method**

Our app is built using 5 steps:

* Create static database
* Front end: XML
* Back end: Java
* Upload “real” database to server
* Take data from webserver APIs through web service, which responds to clients’ devices with JSon type.

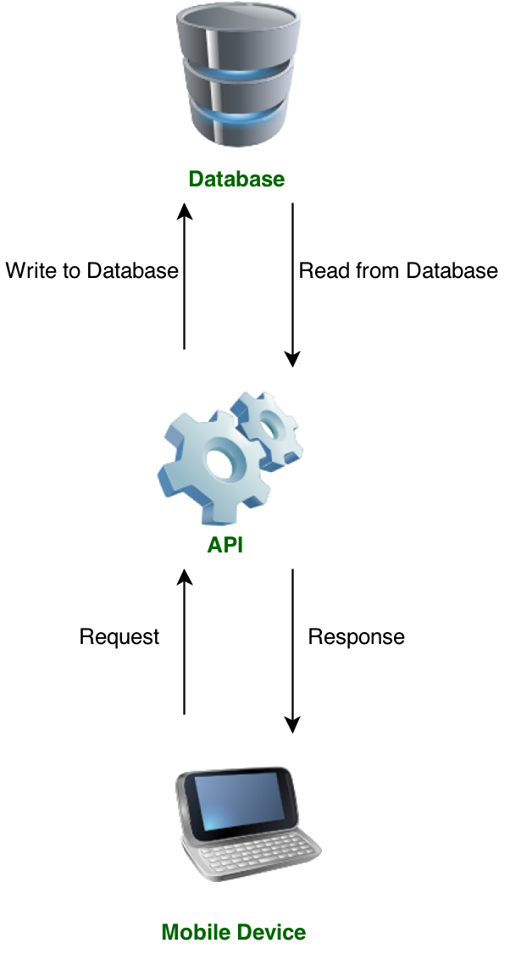


Figure 2. Processes of retrieving database from server.

# BUSINESS MODELING

## **UML/Use case Diagram Explain about the Features of App**

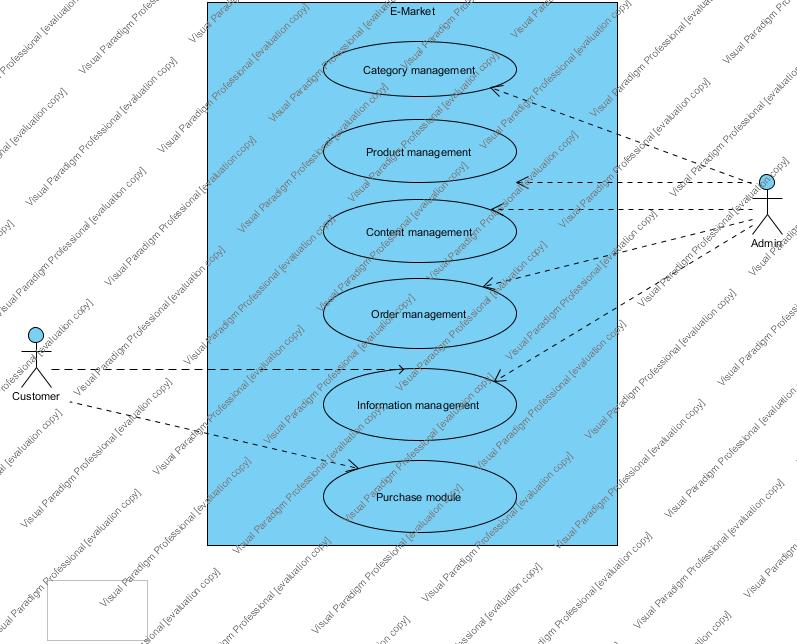


Figure 3. Use case diagram showing relation between app and customer, admin

Admin manages the app by updating, categorising the product type and collecting short information of customer. Customer can also manage some information by own and add selling items in category section.

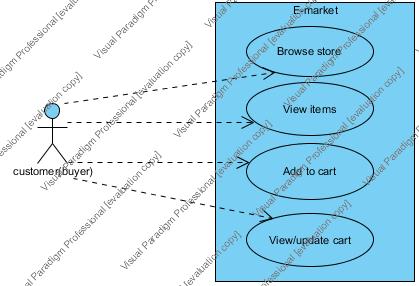


Figure 4. Use case diagram showing how customer can use with the app.

Buyer can select items per category by browsing app. Interested items can be added in the cart.

## **Identify Business Worker and Business Entity**

### Interaction with the app

Table 6. Interaction of Business workers

|  |  |
| --- | --- |
| Business Workers | Roles |
| Administrators | Maintain and update the app  Supervise the app |
| Advertisers | Sponsor the app |
| Users | Accept and follow user agreement  Provide information of users and items |

Table 3. Interaction by Business Entities (cont.)

|  |  |
| --- | --- |
| Business Entities | Roles |
| User interface | Act as a medium for users to interact with the app |
| Items | Trading objects between users |

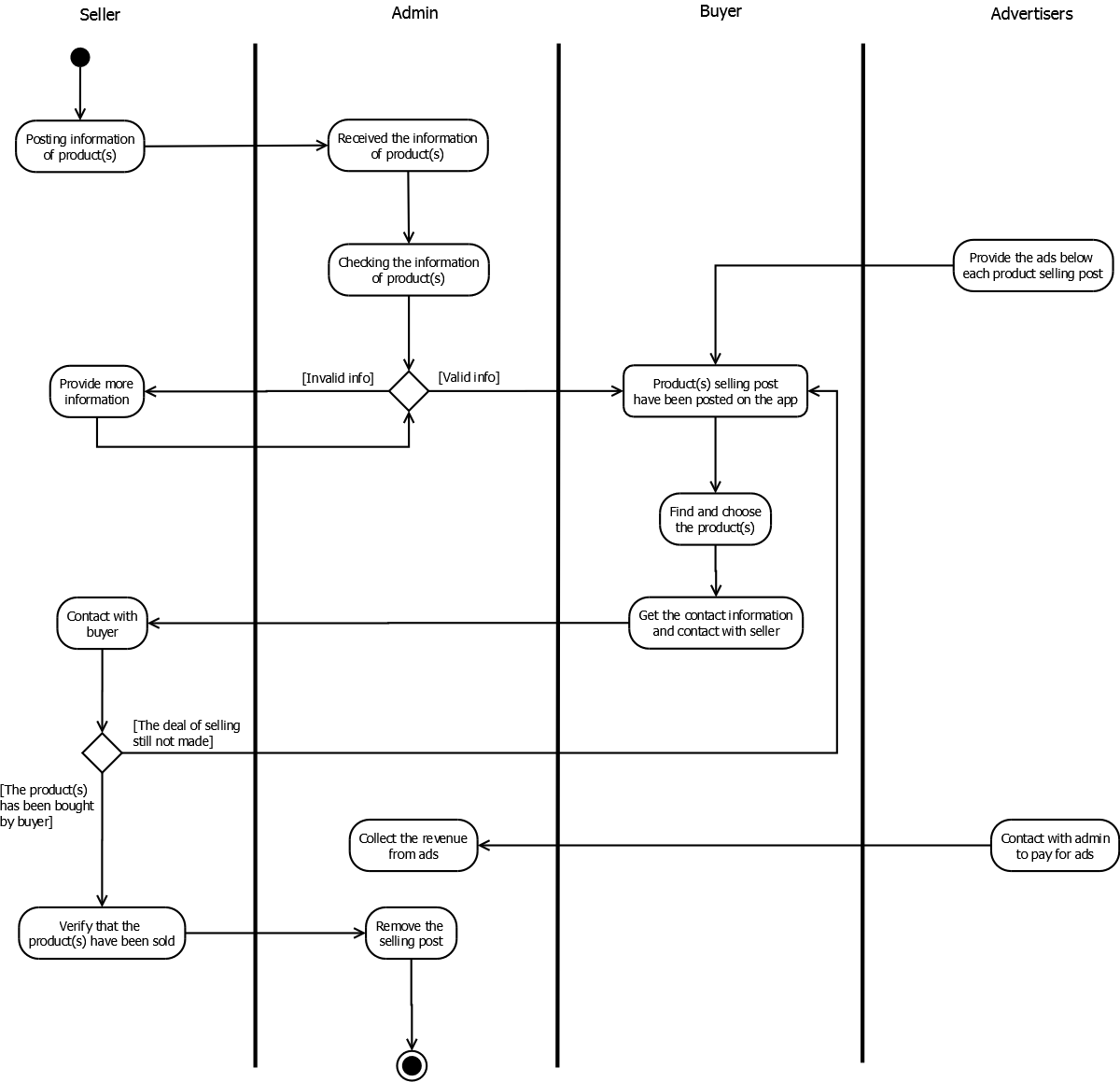


Figure 5. UML Interaction between each role

### **Server and database management**

Table 7. Management by Business Workers

|  |  |
| --- | --- |
| Business Workers | Roles |
| Administators | Observe the server  Customize the service |
| Users | Submit requests for item information, user profile…  Enrich the database |

Table 4. Management by Business Entities (cont.)

|  |  |
| --- | --- |
| Business Entities | Roles |
| Server | Contain the database |
| Service | Automatically respond to requests |

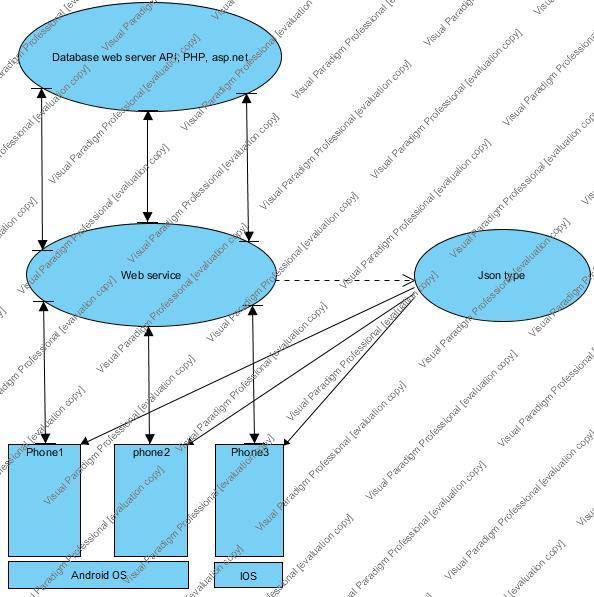


Figure 6. Use case Server and Database Management

### **Ads Management**

Table 8. Ads Management by Business Workers

|  |  |
| --- | --- |
| Business Workers | Roles |
| Advertisers | Provide ads |
| Administrators | Insert ads into the app |
| Users | View ads |

Table 5. Ads Management by Business Entities (cont.)

|  |  |
| --- | --- |
| Business Entities | Roles |
| Ads | Advertise for sponsors and earn money for the app |

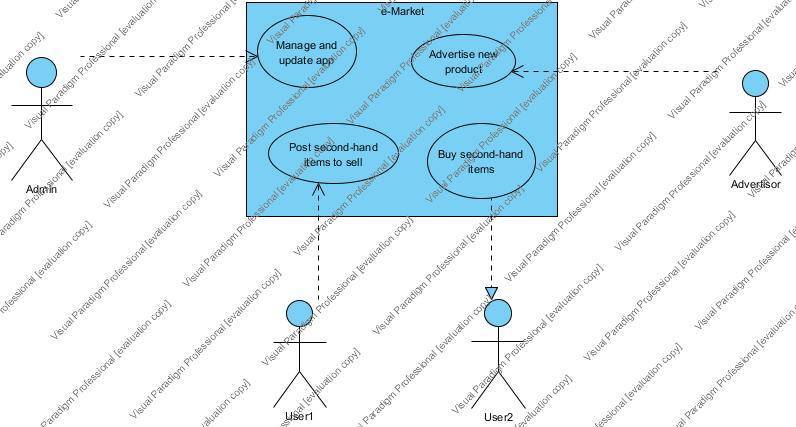
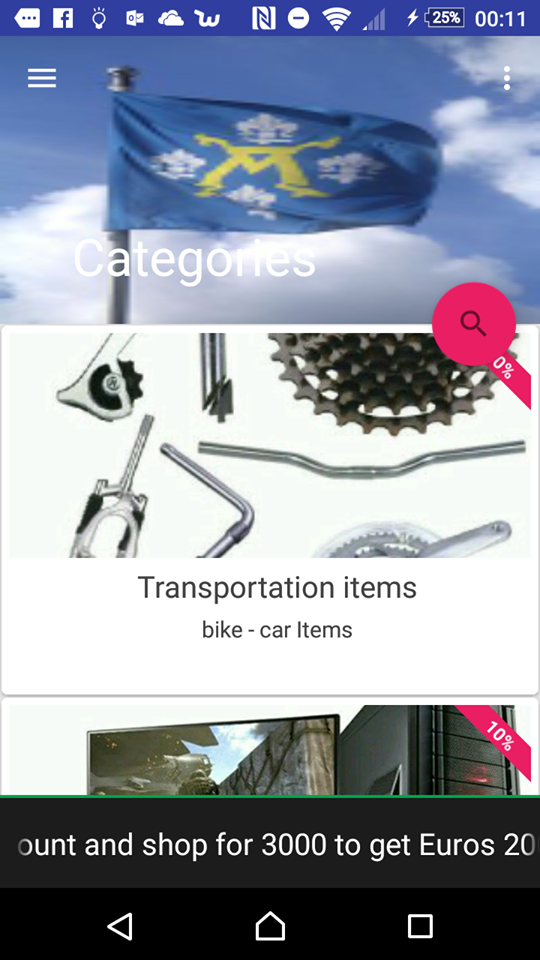


Figure 7. Use case Ads Management

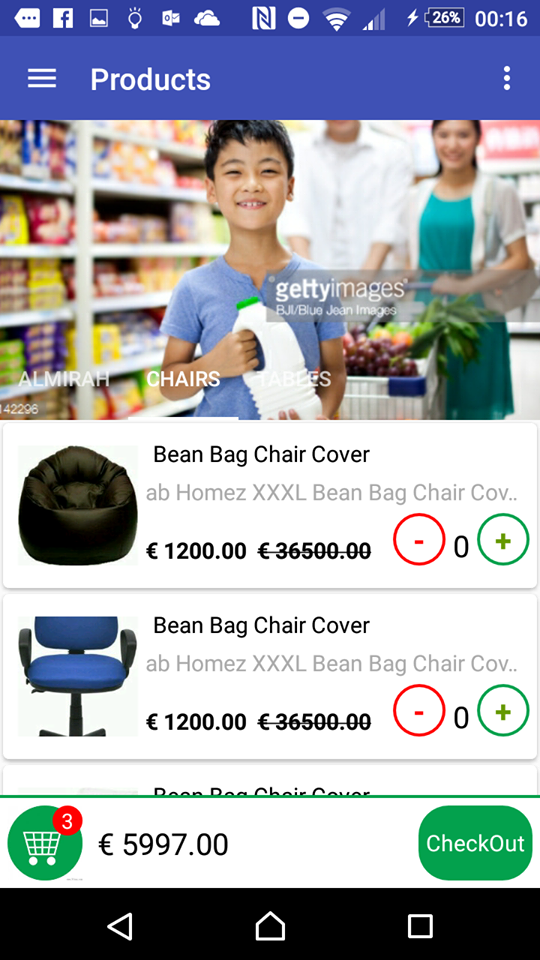
# PROTOTYPE OF PRODUCT



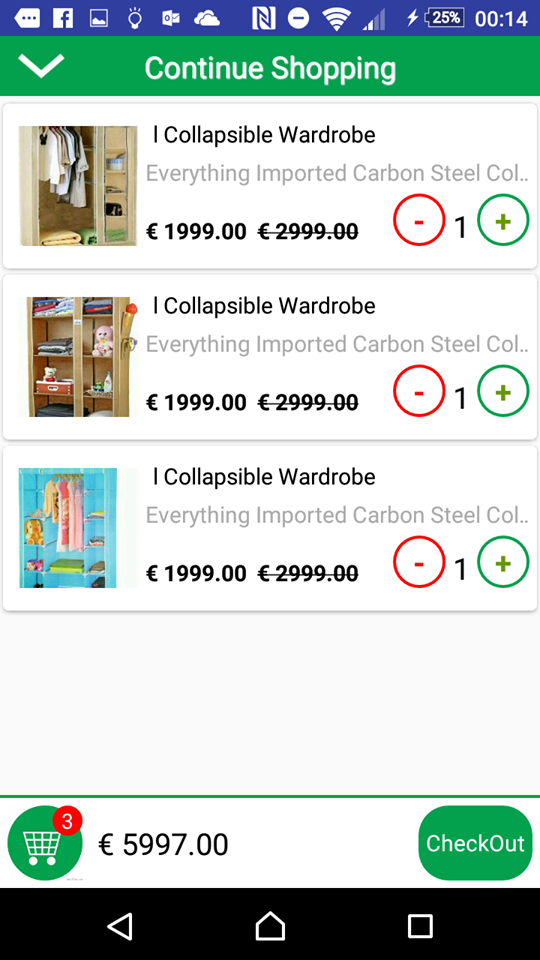
Picture 1. Greetings Screen



Picture 2. Homepage interface



Picture 3. Furniture catalog



Picture 4. Checkout interface

# DATABASE ILLUSTRATION

## **Database show case**



Figure 8. Database version 1



Figure 9. Database version 2.

*Terminology*

THUONGHIEU: Brand

MATH: Brand code

TENTH: Brand name

HINHTH: Brand image (logo)

LOAIKHUYENMAI: Promotion type

MALOAIKM: Promotion type code

TENLOAIKM: Promotion name

CHITIETKHUYENMAI: Promotion detail

MASP: Product code

NGAYBATDAUKM: Promotion start date

NGAYKETTHUCKM: Promotion end date

PHANTRAMKM: Promotion percentage

LOAISANPHAM: Product type

MALOAISP: Product type code

TENLOAISP: Product type name

MALOAI\_CHA: Parent brand type code

GIOHANG: Shopping cart

SOLUONG: Quantity

SANPHAM: Product

TENSP: Product name

GIA: Price

HINHLON: Big image

HINHNHO: Small image

THONGTIN: Information

CHITIETDATHANG: Order detail

MADH: Order code

DONDATHANG: Order receipt

MANV: Employee code

TRANGTHAI: Status

NGAYDAT: Order date

NGAYGIAO: Delivery date

MOTA: Description

MONGMUON: Cart

DANHGIA: Evaluate

NOIDUNG: Content

SOSAO: Number of stars (Quality evaluation)

NHANVIEN: Employee

TENNV: Employee’s name

TENDN: Employer’s name

MATKHAU: Password

DIACHI: Address

NGAYSINH: Date of Birth

SODT: Phone number

GIOITINH: Sex

CMND: Personal ID

MALOAINV: Employee’s class code

LOAINHANVIEN: Employee’s class

TENLOAI: Class name

BINHLUAN: Comments

MABL: Comment code

NGAYVIETBINHLUAN: Comment date

## **Main Code (database )**

DROP DATABASE IF EXISTS DBLAZADA;

CREATE DATABASE DBLAZADA;

USE DBLAZADA;

CREATE TABLE LOAISANPHAM(

MALOAISP INT IDENTITY(1,1),

TENLOAISP VARCHAR(200),

MALOAI\_CHA INT,

CONSTRAINT LOAISANPHAM\_MALOAISP PRIMARY KEY (MALOAISP)

);

CREATE TABLE THUONGHIEU(

MATH INT IDENTITY(1,1),

TENTH VARCHAR(200),

HINHTH TEXT,

CONSTRAINT THUONGHIEU\_MATH PRIMARY KEY (MATH)

);

CREATE TABLE SANPHAM(

MASP INT IDENTITY(1,1),

TENSP VARCHAR(200),

GIA DECIMAL,

HINHLON TEXT,

HINHNHO TEXT,

THONGTIN TEXT,

SOLUONG INT,

MALOAISP INT,

MATH INT,

CONSTRAINT SANPHAM\_MASP PRIMARY KEY (MASP),

CONSTRAINT SANPHAM\_MALOAISP FOREIGN KEY (MALOAISP) REFERENCES LOAISANPHAM(MALOAISP),

CONSTRAINT SANPHAM\_MATH FOREIGN KEY (MATH) REFERENCES THUONGHIEU(MATH)

);

CREATE TABLE LOAINHANVIEN(

MALOAINV INT IDENTITY(1,1),

TENLOAI VARCHAR (20),

CONSTRAINT LOAINHANVIEN\_MALOAINV PRIMARY KEY (MALOAINV)

);

CREATE TABLE NHANVIEN(

MANV INT IDENTITY(1,1),

TENNV VARCHAR (100),

TENDN VARCHAR (20),

MATKHAU VARCHAR (20),

DIACHI TEXT ,

NGAYSINH VARCHAR (10),

SODT VARCHAR (12),

GIOITINH VARCHAR(3),

CMND VARCHAR (20),

MALOAINV INT ,

CONSTRAINT NHANVIEN\_MANV PRIMARY KEY (MANV),

CONSTRAINT NHANVIEN\_MALOAINV FOREIGN KEY (MALOAINV) REFERENCES LOAINHANVIEN(MALOAINV)

);

CREATE TABLE DONDATHANG(

MADH INT IDENTITY(1,1),

MANV INT,

TRANGTHAI VARCHAR(3),

NGAYDAT VARCHAR (10),

NGAYGIAO VARCHAR(10),

MOTA TEXT,

CONSTRAINT DONDATHANG\_MADH PRIMARY KEY (MADH)

);

CREATE TABLE CHITIETDATHANG(

MADH INT ,

MASP INT,

SOLUONG INT,

CONSTRAINT CHITIETDATHANG\_MADH\_MASP PRIMARY KEY (MADH,MASP),

CONSTRAINT CHITIETDATHANG\_MADH FOREIGN KEY (MADH)REFERENCES DONDATHANG(MADH),

CONSTRAINT CHITIETDATHANG\_MASP FOREIGN

KEY (MASP) REFERENCES SANPHAM(MASP)

);

CREATE TABLE LOAIKHUYENMAI(

MALOAIKM INT IDENTITY(1,1),

TENLOAIKM VARCHAR(200),

CONSTRAINT LOAIKHUYENMAI\_MALOAIKM PRIMARY KEY (MALOAIKM)

);

CREATE TABLE CHITIETKHUYENMAI(

MALOAIKM INT ,

MASP INT,

NGAYBATDAUKM VARCHAR (10),

NGAYKETTHUCKM VARCHAR (10),

PHANTRAMKM VARCHAR(1),

CONSTRAINT CHITIETKHUYEMAI\_MALOAIKM\_MASP PRIMARY KEY (MALOAIKM,MASP),

CONSTRAINT CHITIETKHUYEMAI\_MALOAIKM FOREIGN KEY (MALOAIKM) REFERENCES LOAIKHUYENMAI(MALOAIKM),

CONSTRAINT CHITIETKHUYEMAI\_MASP FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP)

);

CREATE TABLE BINHLUAN(

MABL INT IDENTITY(1,1),

MANV INT,

MASP INT,

NOIDUNG TEXT,

NGAYBINHLUAN VARCHAR (10),

CONSTRAINT BINHLUAN\_MABL PRIMARY KEY (MABL),

CONSTRAINT BINHLUAN\_MANV FOREIGN KEY (MANV) REFERENCES NHANVIEN (MANV),

CONSTRAINT BINHLUAN\_MASP FOREIGN KEY (MASP) REFERENCES SANPHAM (MASP)

);

CREATE TABLE DANHGIA(

MASP INT,

MANV INT,

NOIDUNG TEXT,

SOSAO VARCHAR(1),

CONSTRAINT DANHGIA\_MASP\_MANV PRIMARY KEY (MANV,MASP),

CONSTRAINT DANHGIA\_MASP FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP),

CONSTRAINT DANHGIA\_MANV FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV)

);

CREATE TABLE MONGMUON(

MASP INT,

MANV INT,

CONSTRAINT MONGMUON\_MASP\_MANV PRIMARY KEY (MANV,MASP),

CONSTRAINT MONGMUON\_MASP FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP),

CONSTRAINT MONGMUON\_MANV FOREIGN KEY (MANV) REFERENCES NHANVIEN(MANV)

);

# TEST CASE

Table 9. Device Specific Test

|  |  |  |  |
| --- | --- | --- | --- |
| # | Decription | OK or NOT | Remark |
| 1 | Can the app be installed on the device? | OK |  |
| 2 | Does the app behave as designed/desired if there is an incoming call? | OK |  |
| 3 | Does the app behave as designed/desired if there is an incoming SMS? | OK |  |
| 4 | Does the app behave as designed/desired if the charger is connected? | OK |  |
| 5 | Does the app behave as designed/desired if the charger is disconnected? | OK |  |
| 6 | Does the app behave as designed/desired if the device goes to sleeping mode | OK |  |
| 7 | Does the app behave as designed/desired if the device resumes from sleeping mode | OK |  |
| 8 | Does the app behave as designed/desired if the device resumes from lock screen? | OK |  |
| 9 | Does the app behave as designed/desired if the device is tilted? | OK |  |
| 10 | Does the app behave as designed/desired if the device is shaken? | OK |  |
| 11 | Does the app behave as designed/desired if a local message is coming from another app (think of: calendar reminders, to-do task etc.). | OK |  |
| 12 | Does the app behave as designed/desired if a push message is coming from another app (think of: twitter mentions, whatsapp message, word feud invitation, etc.). | OK |  |
| 13 | Is the functionality of all the buttons or keys on the device defined for this app? | OK |  |
| 14 | Verify that buttons or keys which have no defined function have no unexpected behavior on the app when activating. | OK |  |
| 15 | In case there’s a true “home” button available on the device, does the home button get the user back to the home screen of the device? | OK |  |
| 16 | Does the app behave as designed/desired if the “Battery low” message is pushed | OK |  |
| 17 | Does the app behave as designed/desired if the sound on the device is turned off? | OK |  |
| 18 | Does the app behave as designed/desired if the device is in airplane mode? | OK |  |
| 19 | Can the app be de-installed from the device? | OK |  |
| 20 | Does the application function as expected after re-installation? | OK |  |
| 21 | Can the app switch to different apps on the device through multitasking as designed/desired? | OK |  |
| 22 | Are all touch screen positions (buttons) working when a screen protector is used? | OK |  |

Table 10. Network Specific Check

|  |  |  |  |
| --- | --- | --- | --- |
| # | Decription | OK or NOT | Remark |
| 23 | Does the app behave according to specification if connected to the internet through Wi-Fi? | OK |  |
| 24 | Does the app behave according to specification if connected to the internet through 3G? | OK |  |
| 25 | Does the app behave according to specification of the app is out of network reach? | OK |  |
| 26 | Does the app resume working when it gets back into network reach from outside reach of the network? | OK |  |
| 27 | Does the app still work correctly when tethering or otherwise connected to another device | OK |  |
| 28 | What happens if the app switches between networks (Wi-Fi, 3G, 2G) | OK |  |

Table 11. User Interface Check

|  |  |  |  |
| --- | --- | --- | --- |
| # | Decription | OK or NOT | Remark |
| 29 | To keep controls as unobtrusive as possible for instance by fading them out if they are not used for a while. | OK |  |
| 30 | The main function of the app should be apparent immediately. It should speak for itself. | OK |  |
| 31 | In an app, the user should not be able to store files locally, outside the app sandbox. | OK |  |
| 32 | In an app, the user should not be exposed to the permissions of a specific file | OK |  |
| 33 | If performance is slow, indicate a progress status icon (“Loading…”), preferably with specific message. | OK |  |
| 34 | Do not use standard buttons for other functions then that they are normally used for | OK |  |
| 35 | The app should respond to all changes in device orientation, as per the design |  |  |
| 36 | If the app is stopped at an unexpected time, user data should be saved locally and available at start-up. |  |  |
| 37 | Keyboard adjusts to expected input (for instance numbers/letters when expected). |  |  |

This checklist above is based on the recommendations of Apple and some other experts. This checklist is not a substitute for a usability test, to get a good feeling of the user experience of an app a user experience test is always the most reliable method. User interface checks toelichten.

Table 12. App Specific Check

|  |  |  |  |
| --- | --- | --- | --- |
| # | Decription | OK or NOT | Remark |
| 1 | Stability check: if the app has a list (for instance of pictures) in it, try scrolling through it at high speed. | OK | There’s animation which makes the app feel clunky |
| 2 | Stability check: if the app has a list (for instance of pictures) in it, try scrolling to before the first picture or behind the last picture. | OK |  |
| 3 | The app does not interfere with other apps when in background/multitasking mode (using GPS, playing music, etc.). | OK |  |
| 4 | Verify most common gestures used to control the app. | OK |  |
| 5 | What happens if you select different options at the same time (undesired multi-touch, for example – select two contacts from the phone book at the same time). | OK |  |

# CONCLUSION

To sum up, the E-Market App has gone through a lot of processes like making database (fake-database) and building prototype of App and testing step and so on, in order to make sure this App that can work stably before releasing to Google Play.

After the time for building the App, we have a prototype of E-Market Mobile App with basic features. In the next step, collection database from supermarket and sellers and upload it to a server are building. Finally, the purpose of the App is that collect all information from many fields like items/goods from supermarkets/sellers ( the App is in the time of developing some more advanced features that are enough to meet the actual needs for shopping ) , or education or events/news around Turku city.

# References

Heading of appendix

None

Heading of appendix

None