THE UNIVERSITY OF HONG KONG

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Project 2017-2018

Mobile Web Application - Electronic Payment System

Progress Report for CAES

Supervisor: Dr. W. H. Lam

Name: ZHU Zicong

UID: 3035142132

Curriculum: BEng(Computer Engineering)

Summary

**Since the last few decades, smart phone has become necessary for everyone in the daily life, instead of just tools for communications. Due to the rapid development of mobile system, the mobile applications made people’s life more convenient such as map applications and delivering applications. Among the large quantities of mobile applications, electronic payment applications have become an important part of the application family. It has become significant for developing and improving electronic payment applications with great qualities.**

**In this project, a mobile web application for electronic payment will be designed and implemented. This electronic payment application will be focusing on providing users with convenient and safe electronic payment experiences. Multiple kinds of functions will be implemented in this application including QR code and face recognition, which have been widely applied in mobile application fields to simply the procedures of payment.**

**In the design and development of this project, a RESTful structure will be used with android front-end application, Java back-end server, MySQL database and Google Cloud Message pushing service. Furthermore, face recognition parts will be implemented by Tensorflow in Artificial Neural Network field, a method called Eigenface will be referenced.**

**A convenient and safe electronic payment systems will be developed as the goal of this project. Compared to other existing applications, it is focusing on a brand-new system architecture, better payment experiences among users. However, there’re quite a lot limitation in the final product, like safety method and capacity of the application, which shall be improved for further considerations.**

Table of Content

[Summary 2](#_Toc507823453)

[List of Figures & Tables 5](#_Toc507823454)

[List of Symbols 5](#_Toc507823455)

[1. Introduction 6](#_Toc507823456)

[1.1. Backgroud 6](#_Toc507823457)

[1.2. Objective 6](#_Toc507823458)

[1.3. Project Scope and Limitation 6](#_Toc507823459)

[1.4. Project Status 7](#_Toc507823460)

[Till this progress report submitted, the structure of the electronic payment system has already been built. According to the following [Table. 1.3.1] with specific description, most of the functions have already been implemented. However, the GCM(Google Cloud Messaging) and Face Recognition haven’t been developed yet. 7](#_Toc507823461)

[2. Analysis of Problem 8](#_Toc507823462)

[2.1. Variety of Modules 8](#_Toc507823463)

[2.2. Security Problem 8](#_Toc507823464)

[3. Design & Construction 9](#_Toc507823465)

[3.1. Overall System Architecture 9](#_Toc507823466)

[3.1.1. System Overview 9](#_Toc507823467)

[3.1.2. Front-end Application 9](#_Toc507823468)

[3.1.3. Server Module 9](#_Toc507823469)

[3.1.4. Database Module 11](#_Toc507823470)

[3.1.5. GCM 12](#_Toc507823471)

[3.2. GUI Flow 12](#_Toc507823472)

[3.3. Data Flow 12](#_Toc507823473)

[3.3.1. Data flow chart of Login 13](#_Toc507823474)

[3.3.2. Data flow chart of Registration 13](#_Toc507823475)

[3.3.3. Data flow chart of Payment\_byScan 13](#_Toc507823476)

[3.3.4. Data flow chart of Payment\_byQRCode 14](#_Toc507823477)

[3.3.5. Data flow chart of Payment\_byTransfer 14](#_Toc507823478)

[3.3.6. Data flow chart of Gathering\_byScan 14](#_Toc507823479)

[3.3.7. Data flow chart of Gathering\_byQRCode 15](#_Toc507823480)

[4. Further Development 16](#_Toc507823481)

[4.1. UI design 16](#_Toc507823482)

[4.2. Implementation of Geolocation by WIFI 16](#_Toc507823483)

[4.3. Implementation of GCM 16](#_Toc507823484)

[5. Conclusions 17](#_Toc507823485)

List of Figures & Tables

[Table. 1.3.1 Finished Function List 7](#_Toc507823424)

[Figure. 3.1.1.1 Overall System Architecture 9](#_Toc507823425)

[Figure. 3.1.3.1 Flow Diagram of Server 10](#_Toc507823426)

[Table. 3.1.4.2 Table of fyp\_user 11](#_Toc507823427)

[Table. 3.1.4.2 Table of fyp\_trans 11](#_Toc507823428)

[Figure. 3.2.1 GUI Flow Chart 12](#_Toc507823429)

[Figure. 3.3.1 DFD\_Login 13](#_Toc507823430)

[Figure. 3.3.2 DFD\_Registration 13](#_Toc507823431)

[Figure. 3.3.3 DFD\_Payment\_byScan 13](#_Toc507823432)

[Figure. 3.3.4 DFD\_Payment\_byQRCode 14](#_Toc507823433)

List of Symbols

|  |  |
| --- | --- |
| App | Application |
| GCM | Google Clouding Messaging |
| DFD | Data Flow Diagram |
| SQL | Structured Query Language |
| JSON | JavaScript Object Notation |
| POS | Point of Sale |
| FYP | Free Yeah Payment |
| UI | User Interface |
| IP | Internet Protocol |
| MAC | Medium Access Control |
| JDBC | Java DataBase Connectivity |
| REST | REpresentational State Transfer |
| URL | Uniform Resource Locator |

1. **Introduction**
   1. **Backgroud**

The electronic payment is generally defined as “Users send payment orders to bank systems directly or indirectly via electronic devices, to achieve currency payment and fund transfer”. However, in the recent decades while the smart mobile phones are being developed rapidly, a great number of mobile applications for electronic payment are carried out like Alipay, WeChat Pay and PayPal, which are developed as professional commercial applications and shows significant influences on users’ daily lives.

* 1. **Objective**

In this project named “Mobile Web Application - Mobile Electronic Payment System”, an Android mobile application called “Free Yeah Payment” (denoted as FYP in the following) will be developed.

FYP applications are developed based on ideas of “Better electronic payment experiences on interactions within users”. In other words, bank is not a considerable role in this project, all functions are designed for interactions between users and users.

As a professional application system, it is necessary to implement the following components: mobile application, main server and back-end database. And platforms, structures and languages for each parts, connections and communication methods between them shall be considered.

Particularly speaking, besides all the basic functions like login and registration, electronic payment methods are the most important parts to be considered. Three kinds of methods for electronic payment will be developed in this project: payment by directly transfer, payment by QR code and payment by Face Recognition. Function of payment by directly transfer is added as a backup choice for users, which can make the whole application more completed and professional. Function of payment by QR code is implemented as the main method in FYP, which satisfies the payment needs in different scenarios. In other words, any two users can use QR code in serval ways to achieve the goal of payment. As for the last function of payment by Face Recognition, this is considered as the new feature in electronic payment systems. In this feature, the final goal is to achieve “Payment by scanning faces on portable(mobile) devices.” Till now, there’re still many works to be done in this field, even Alipay does not support face recognition on mobile devices.

* 1. **Project Scope and Limitation**

This FYP project will be fully developed on a personal computer and a mobile phone, which means the main server and database will be deployed on one single laptop. Therefore, user capacity supported will be a great limitation in this environment. Meanwhile, as for the front-end application deployed on the mobile phone, the final product will not support for other operation systems, but only android, even more it may carry out unpredictable issues like GUI layout missing and function support on mobile phone of other brand different from the testing machine (a Huawei and MI phone are tested in this project).

According to the basic ideas for function design in this project, it is assumed that all the functions are designed for interactions within users. Therefore, as it was mentioned before, the role of bank is ignored in this project.

As for the safety, some simple security methods are implemented here like user’s IP checking and QR code valid number. However, compared to formal commercial electronic applications, security methods implemented in FYP have great limitations, which worth further improvements.

For the Face Recognition feature which has not be fully implemented yet, due to the hardware and Eigenface method used, a limited number of users may be supported. The capacity details will be derived once this feature finished.

* 1. **Project Status**

Till this progress report submitted, the structure of the electronic payment system has already been built. According to the following [Table. 1.3.1] with specific description, most of the functions have already been implemented. However, the GCM (Google Cloud Messaging) and Face Recognition haven’t been developed yet.

|  |  |
| --- | --- |
| **Function Name** | **Description** |
| **Login** | **Users login with necessary information** |
| **Registration** | **Users register with necessary information** |
| **Payment\_byScan** | **Users scan a QR Code to pay. Transfer value input is required** |
| **Payment\_byQRCode** | **Users generate a QR Code which is able to be scanned by others. Transfer value input is required before generating** |
| **Payment\_byAccount** | **Users directly type in target and value to transfer** |
| **Gathering\_byScan** | **Users scan a QR Code to gathering. Transfer value is defined in the QR Code** |
| **Gathering\_byQRCode** | **Users generate a QR Code which is able to be scanned by others to transfer** |

Table. 1.3.1 Finished Function List

1. **Design & Construction**
   1. **Overall System Architecture**
      1. **System Overview**

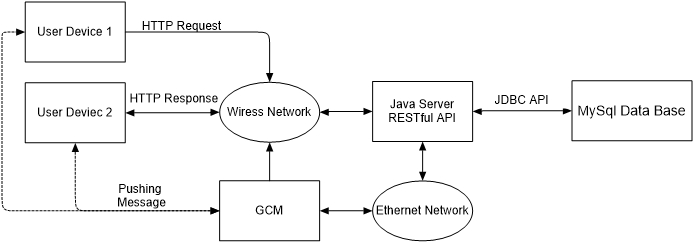
****

Figure. 3.1.1.1 Overall System Architecture

[Figure. 3.1.1.1] shows the overall system architecture of FYP project. There are 4 main modules in the whole system: Front-end application, main server, database, and GCM. The front-end application is designed for end-users, who are going to use this electronic payment service. Meanwhile the front-end application is deployed in users’ end-devices, which could be portable. Both main server and database are deployed in one single PC, therefore, the hardware issue could be a limitation for system capacity and performance. According to [Figure. 3.1.1.1], the users send HTTP request from the mobile device to the main server via wireless network (WIFI mostly). And the main server communicates with MySQL database through JDBC (Java Database Connectivity) API. As for the GCM, which has not been fully deployed yet, main server will send request to GCM, which is in the cloud supported by Google, via ethernet network. GCM will push the requested messages to specific users via wireless network.

Technical details and justification of engineering choices will be discussed in the following prats.

###here

* + 1. **Front-end Application**

In this project, the front-end application is developed in Android, which is the most popular operating systems for smart phones, and Android Studio is used for development as the IDE.

Front-end application takes the responsibility of interaction with different users; therefore, principles of UI design shall be considered here like convenience, precise and be friendly to users.

* + 1. **Server Module**

In my design of this server module, a kind of special approach is considered here: RESTful API structure. REST stands for the full term of Representational State Transfer, which describes an interaction mode between client and server and RESTful API means API in REST style structure. In other words, with more generally definition, REST structure uses URL for locating resources and HTTP action for behaviors.

In fact, RESTful API is usually used in web applications because it utilizes both URL and HTTP, which are mostly used for web. However, in my mind, it could valuable implementing a RESTful API for an Android application, which means the server design is well structured and easy for modification, compared with communication by socket. For example, you may have to design your own protocol for socket communication. Moreover, with RESTful API structure, it is easy to transplant this application to a new platform of web browser.

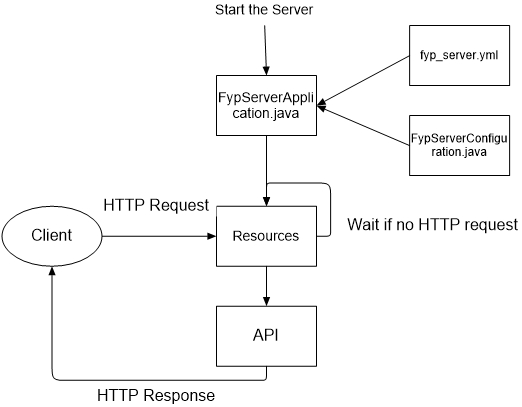
****

Figure. 3.1.3.1 Flow Diagram of Server

* Run file FypServerApplication.java to start the server
* File fyp\_server.yml defines the port of the server
* File FYPServerConfiguration.java defines the other initial configurations
* Resource package contains all the resource files (like Login.java)
* When there is a HTTP request from the client, the server will locate the resource file according to the HTTP request with URL. If there is no request, just waiting for the activation.
* Once the resource file is located, call the API files to response. In other words, API files define the format of response
* The response is sent back to client by HTTP
  + 1. **Database Module**
       1. **Overview of Database**

MySQL is deployed here. And according to the other flow designs, 2 tables a created: table fyp\_user for user information and table fyp\_trans for transaction information.

* + - 1. **Table of fyp\_user**

|  |  |  |
| --- | --- | --- |
| **Field/Attribute** | **Data Type** | **Description** |
| **user\_id** | **int, not null** | **Primary key, auto increment** |
| **user\_name** | **varchar(10), not null** | **Set to be unique** |
| **user\_password** | **varchar(20), not null** |  |
| **user\_emailAddr** | **varchar(20)** | **user’s email address** |
| **user\_balance** | **int** | **user’s current balance** |
| **user\_createTime** | **timestamp, not null** | **this account’s created time** |
| **user\_lastModifiedTime** | **timestamp, not null** |  |
| **user\_valid** | **char, not null** | **1 for valid, 0 for invalid** |
| **user\_bankAccount** | **varchar(20)** | **user’s bank account** |
| **user\_ip** | **varchar(16)** | **user’s last login IP address** |
| **user\_qrValidNum** | **varchar(6)** | **6-bit random string for current QR Code generated by this user** |
| **user\_qrValue** | **int** | **value information shall be contained be the current QR Code generated by this user** |

Table. 3.1.4.2 Table of fyp\_user

* + - 1. **Table of fyp\_trans**

|  |  |  |
| --- | --- | --- |
| **Field/Attribute** | **Data Type** | **Description** |
| **trans\_id** | **int, not null** | **Primary key, auto increment** |
| **trans\_fromID** | **int, not null** | **user id of payment side** |
| **trans\_toID** | **int, not null** | **user id of gathering side** |
| **trans\_fromName** | **varchar(10), not null** | **user name of payment side** |
| **trans\_toName** | **varchar(10), not null** | **user name of gathering side** |
| **trans\_fromBalance** | **int, not null** | **user’s balance of payment side** |
| **trans\_toBalance** | **int, not null** | **user’s balance of gathering side** |
| **trans\_value** | **int, not null** | **value of this transaction** |
| **trans\_creatTime** | **timestamp, not null** |  |

Table. 3.1.4.2 Table of fyp\_trans

* + - 1. **JDBC**

JDBC stands for Java DataBase Connectivity, which is a Java API used to execute SQL statement for retrieving or updating data from MySQL database.

* + 1. **GCM**

GCM stands for Google Clouding Messaging, which is used for pushing a message to the end-user device. Because the server is built in request-response mode, therefore, server cannot send a message to the user if user doesn’t give a request. GCM takes the responsibility of pushing a message to the end-user when transaction happened or finished.

* 1. **GUI Flow**

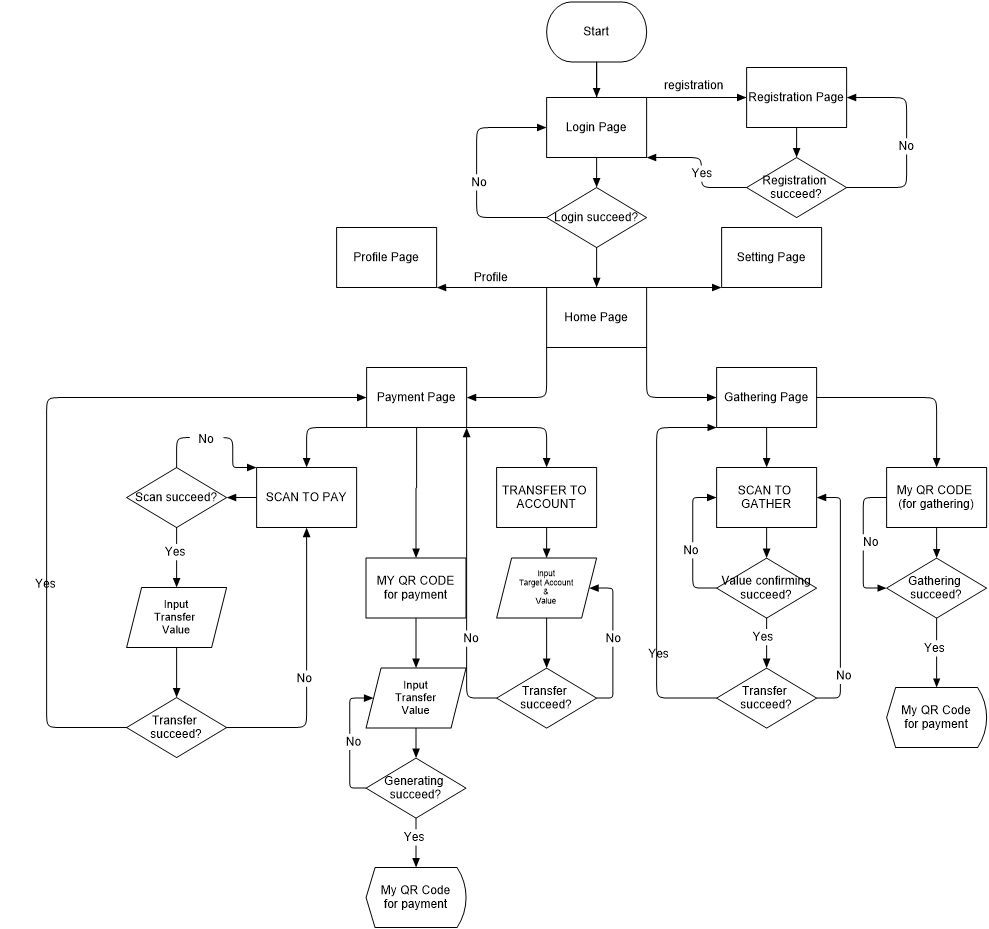
****

Figure. 3.2.1 GUI Flow Chart

* 1. **Data Flow**

The following data flow charts corresponds to the [Table 1.3.1 Project Scenario Definition]

* + 1. **Data flow chart of Login**

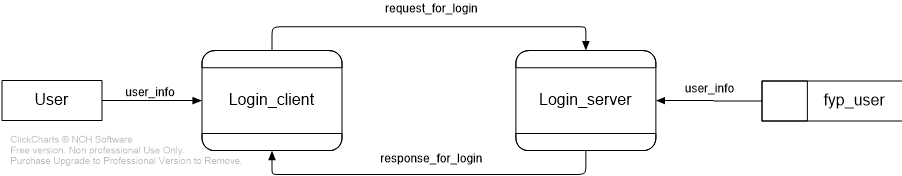
****

Figure. 3.3.1 DFD\_Login

* + 1. **Data flow chart of Registration**

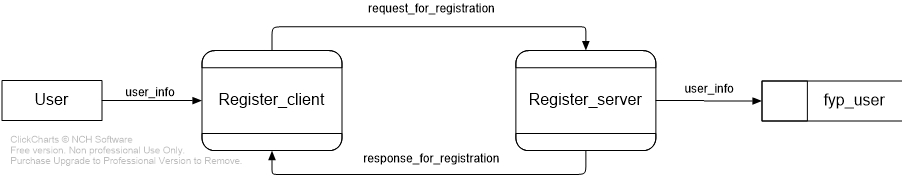
****

Figure. 3.3.2 DFD\_Registration

* + 1. **Data flow chart of Payment\_byScan**

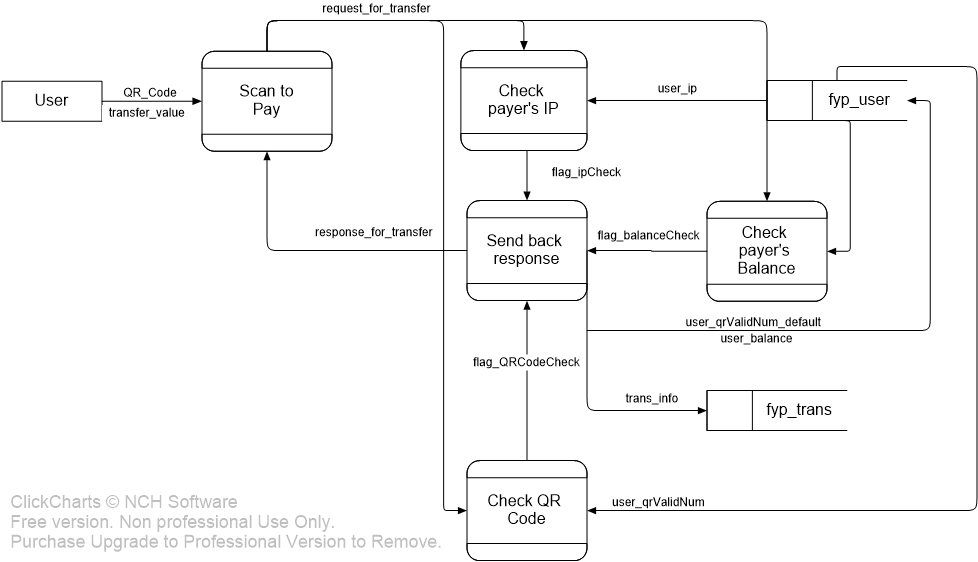
****

Figure. 3.3.3 DFD\_Payment\_byScan

* + 1. **Data flow chart of Payment\_byQRCode**

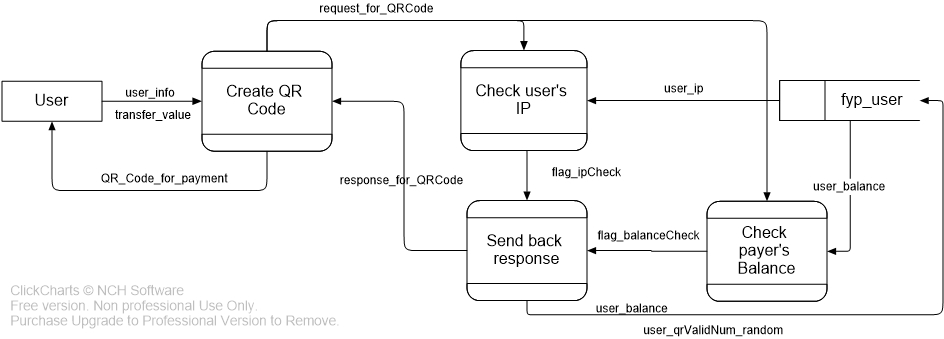
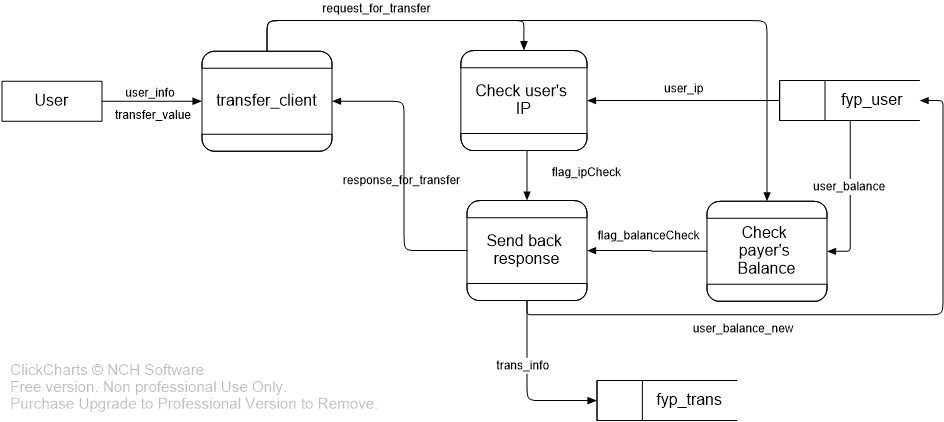
****

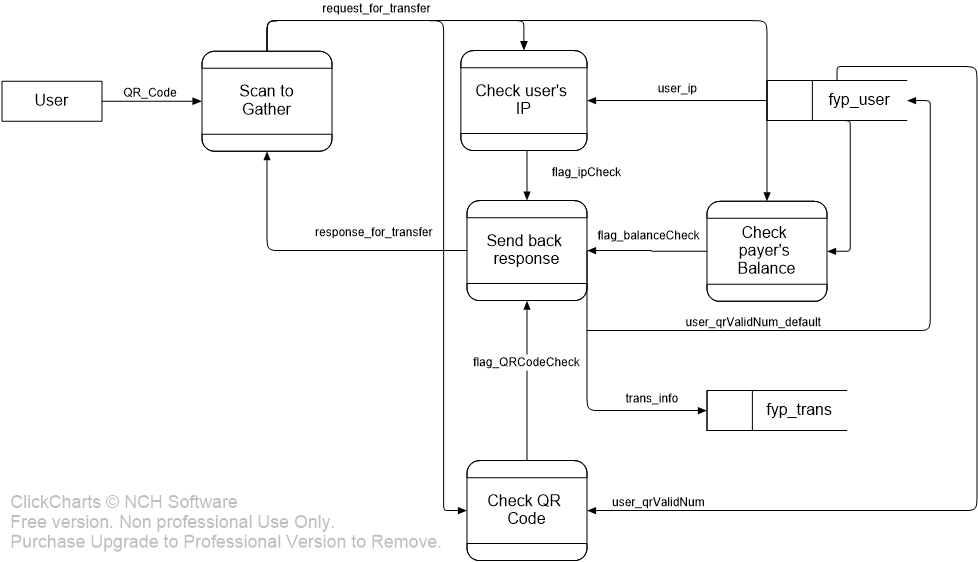
Figure. 3.3.4 DFD\_Payment\_byQRCode

* + 1. **Data flow chart of Payment\_byTransfer**

****

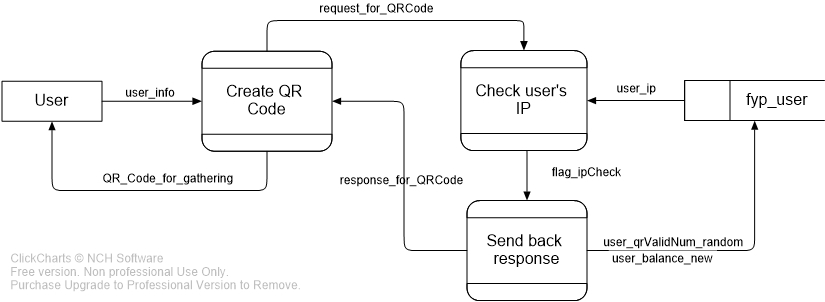
*Figure. 3.3.5 DFD\_Payment\_byTransfer*

* + 1. **Data flow chart of Gathering\_byScan**

****

*Figure. 3.3.6 DFD\_Gathering\_byScan*

* + 1. **Data flow chart of Gathering\_byQRCode**

****

*Figure. 3.3.7 DFD\_Gathering\_byQRCode*

1. **Further Development**
   1. **UI design**

In this stage of project, the UI is really simple and seems to be a prototype. Therefore, one part of the further development of this application is to improve the UI design of the front-end application. The ideal UI shall be not only beautiful, which is attractive to new users, but also friendly to users.

* 1. **Implementation of Geolocation by WIFI**

Electronic payment can be considered as a method of electronic business. As far as I’m considering, this idea of geolocation by WIFI can be applied to giving the users some shop recommendations while they are nearby. The specific location of the user is calculated by WIFI signals around the user. Compared to geolocation by GPS, geolocation by WIFI has some advantages like no need to change the current WIFI networks. As time goes by, there’re more and more access points built around, which helps geolocation by WIFI method becoming more fast and accurate.

* 1. **Implementation of GCM**

GCM, the Google Clouding Messaging is a service implemented for sending a message to user even though there is no request from the user device. This is useful because our server is designed in mode of request-response. Sometimes we need unidirectionally send message to the user, therefore the GCM is need. However, the technical details of GCM will be considered later.

1. **Conclusions**

This project called “Mobile Web Application - Electronic Payment System”, which is focusing on the system architecture design and practical implementation. The final goal is to design and develop a full functional electronic payment system with capabilities of convenience, safety and user friendly. Multiple technical tools and methods are involved in this project like Android programming, RESTful API developing for a Java server, database technology, cloud message pushing system and so on. Furthermore, in the later stage of this project, it’s necessary to improve for a better UI, a better security method and the implementation of geolocation by WIFI.