The university of Hong Kong
Department of electronical and electronic engineering

Mobile Web Application – Electronic Payment System

Final Year Project 2017-2018

Supervisor: Dr. W. H. Lam

Name: ZHU Zicong

UID: 3035142132

Curriculum: BEng(Computer Engineering)

CONTENT

1. INTRODUCTION

1. INTRODUCTION - BACKGROUND

- **Definition of Electronic Payment**

  Users send payment orders to bank systems directly or indirectly via electronic devices, to achieve currency payment and fund transfer.

1. INTRODUCTION - BACKGROUND

- **Categories of Electronic Payment**
  - Internet
  - Telephone
  - Point of Sale (POS)
  - **Mobile Device**

1. INTRODUCTION - OBJECTIVES

- **To build an electronic payment system with features:**
  - Multi-functional
  - Safe & Stable
  - Convenient & User-friendly

- Especially focusing on **User-to-User** experience

1. INTRODUCTION – MAIN FUNCTIONS

- **3 ways to pay money**
  - Scan a QR Code
  - Provide a QR Code (like a cheque)
  - Transfer by Entering the Account

- **3 ways to receive money**
  - Scan a QR Code
  - Provide a QR Code (like an ID card)
  - Scan a Face

2. SYSTEM DESIGN

2. SYSTEM DESIGN - OVERVIEW

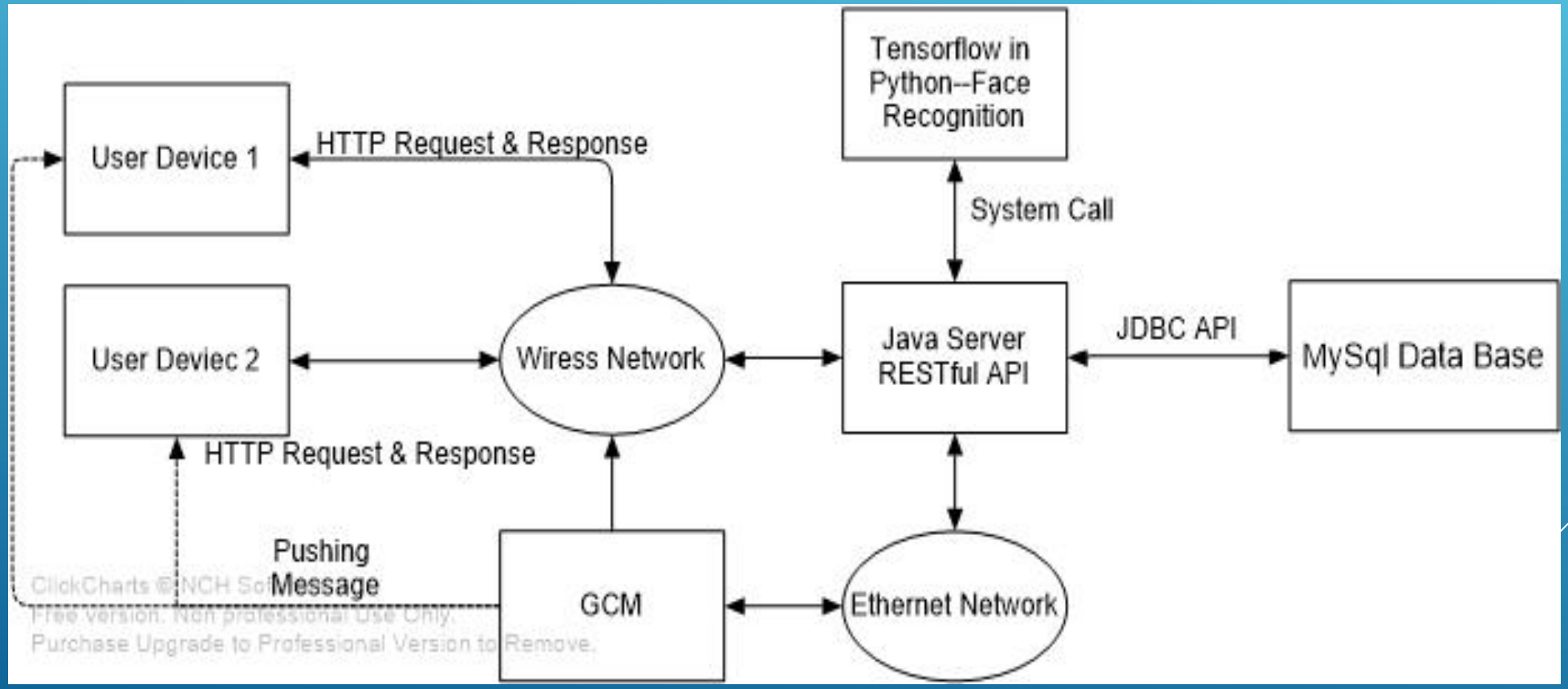
- **Android front-end application**
- **Java server with RESTful API**
- **MySQL database**
- **GCM**

Uses URL for locating resources and HTTP actions for behaviors

Google Cloud Messaging

2. SYSTEM DESIGN - OVERVIEW

- **Overall Architecture**



2. SYSTEM DESIGN

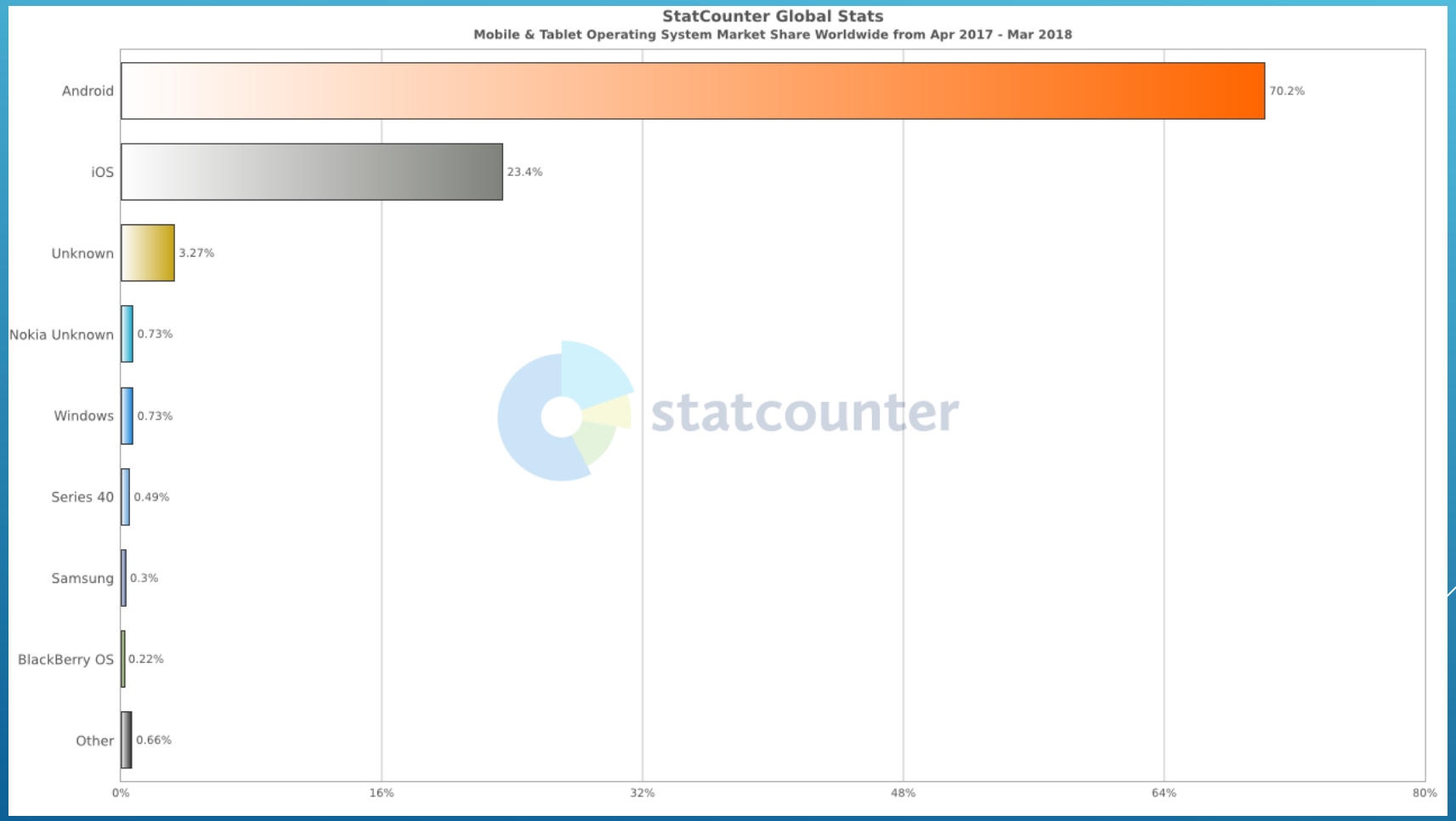
ANDROID APPLICATION

2. SYSTEM DESIGN – ANDROID APPLICATION

- **What is Android?**
  - A mobile operating system
  - Developed by Google
  - Mainly deployed on portable devices
    - Smartphones
    - Tablets
    - Wearable devices

2. SYSTEM DESIGN – ANDROID APPLICATION

- **Why Android?**

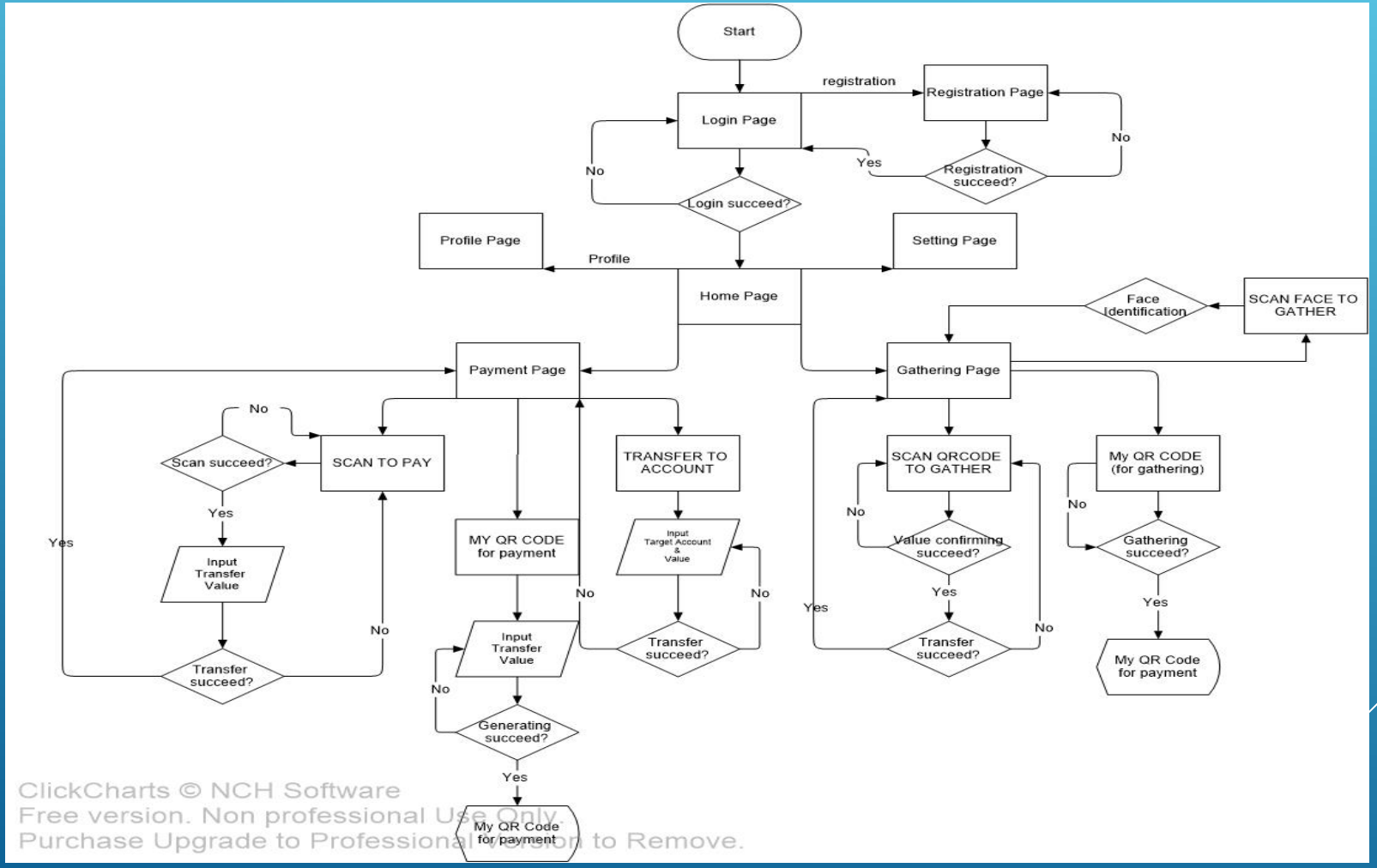


2. SYSTEM DESIGN – ANDROID APPLICATION

- **Development Environment for Android Application**
  - Programming Environment
    - Windows 10
  - IDE
    - Android Studio Ver. 3.1.1
  - Simulation
    - Android smart phones (Ver. 5.1)
    - Virtual device is supported
  - Android Version Supported
    - Ver. 4.0 +

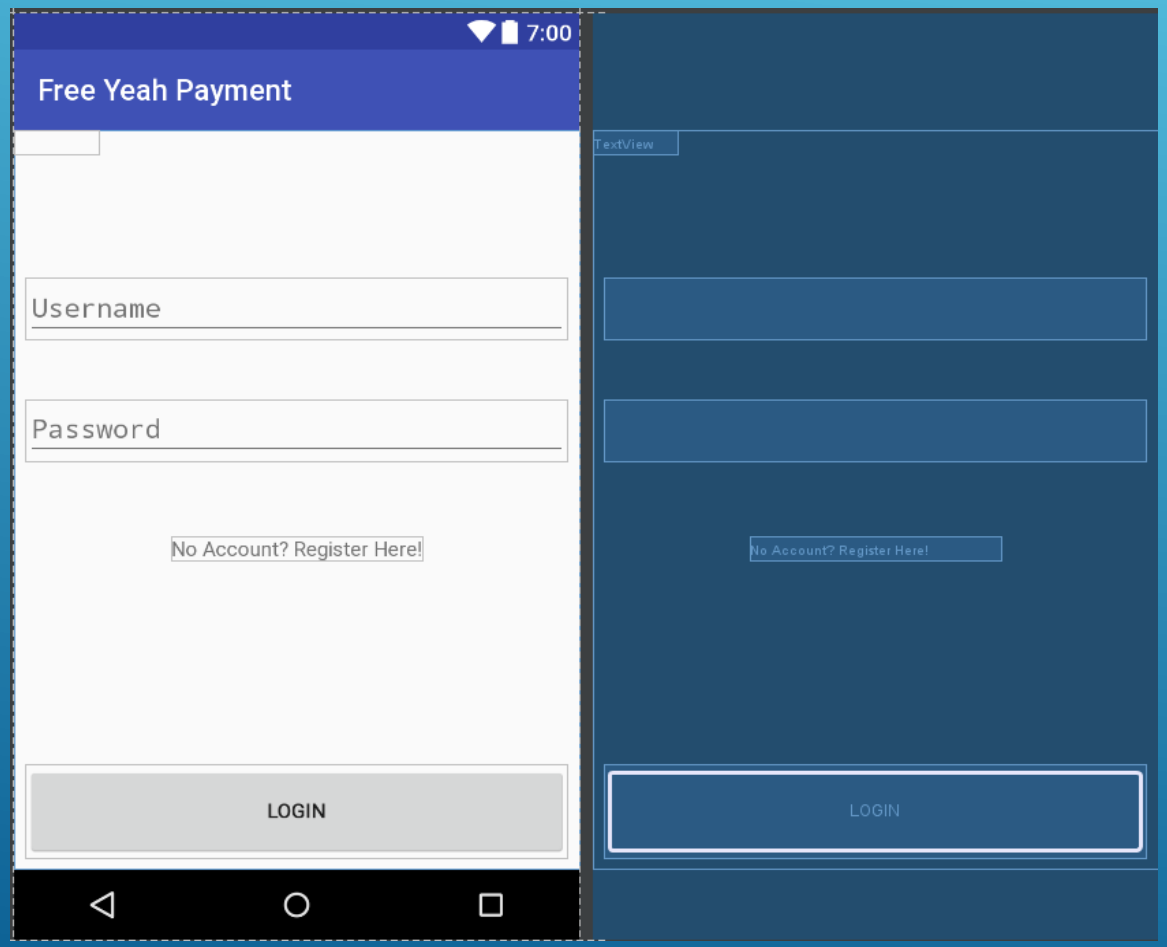
2. SYSTEM DESIGN – ANDROID APPLICATION

- **Overall User Interface Flow Diagram**



2. SYSTEM DESIGN – ANDROID APPLICATION

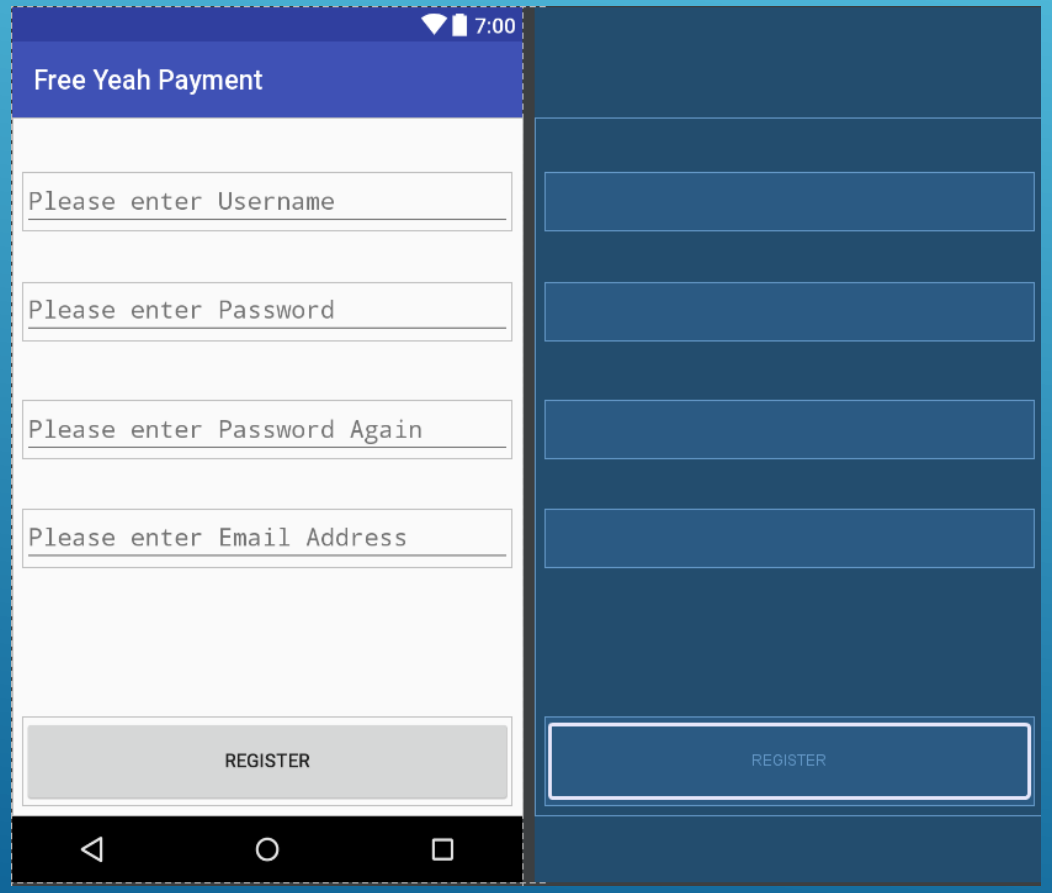
- **Login Page**



- Enter the user name
- Enter the password
- Click the button to login
- Click the text to register

2. SYSTEM DESIGN – ANDROID APPLICATION

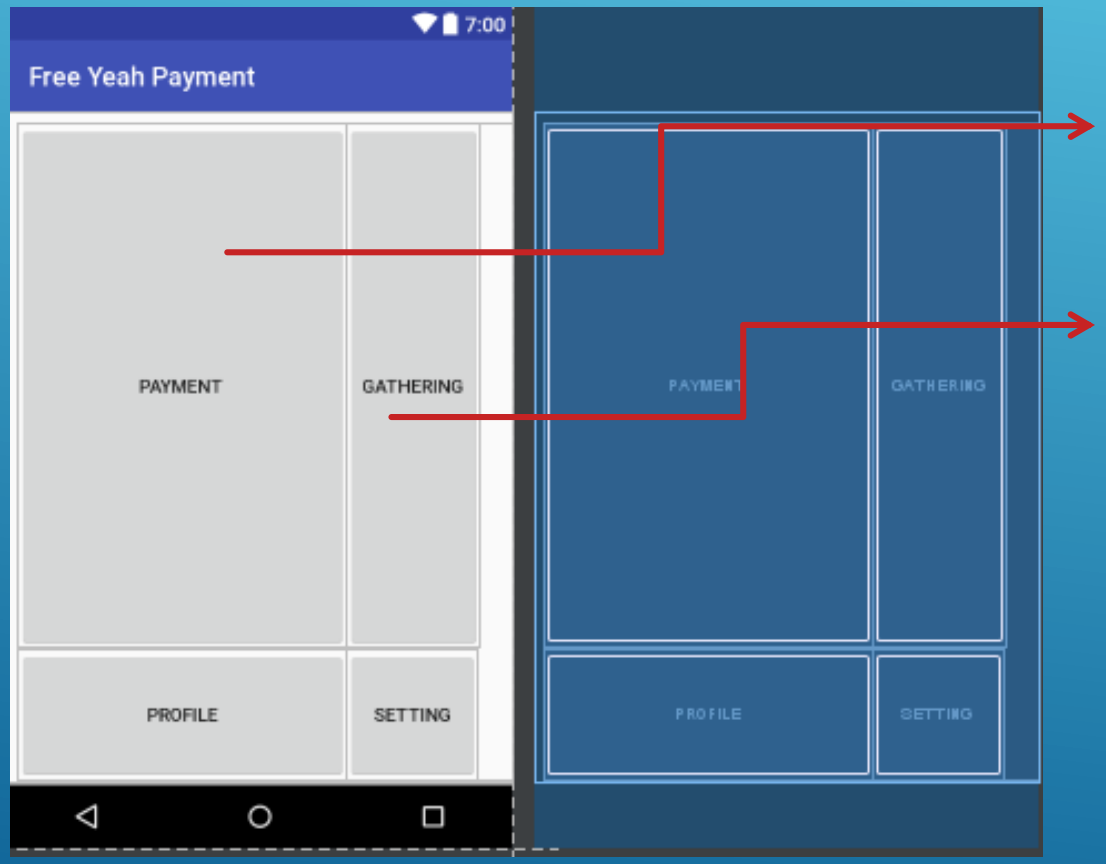
- **Register Page**



- Enter the user name
- Enter the password
- Enter the email address
- Click the button to register

2. SYSTEM DESIGN – ANDROID APPLICATION

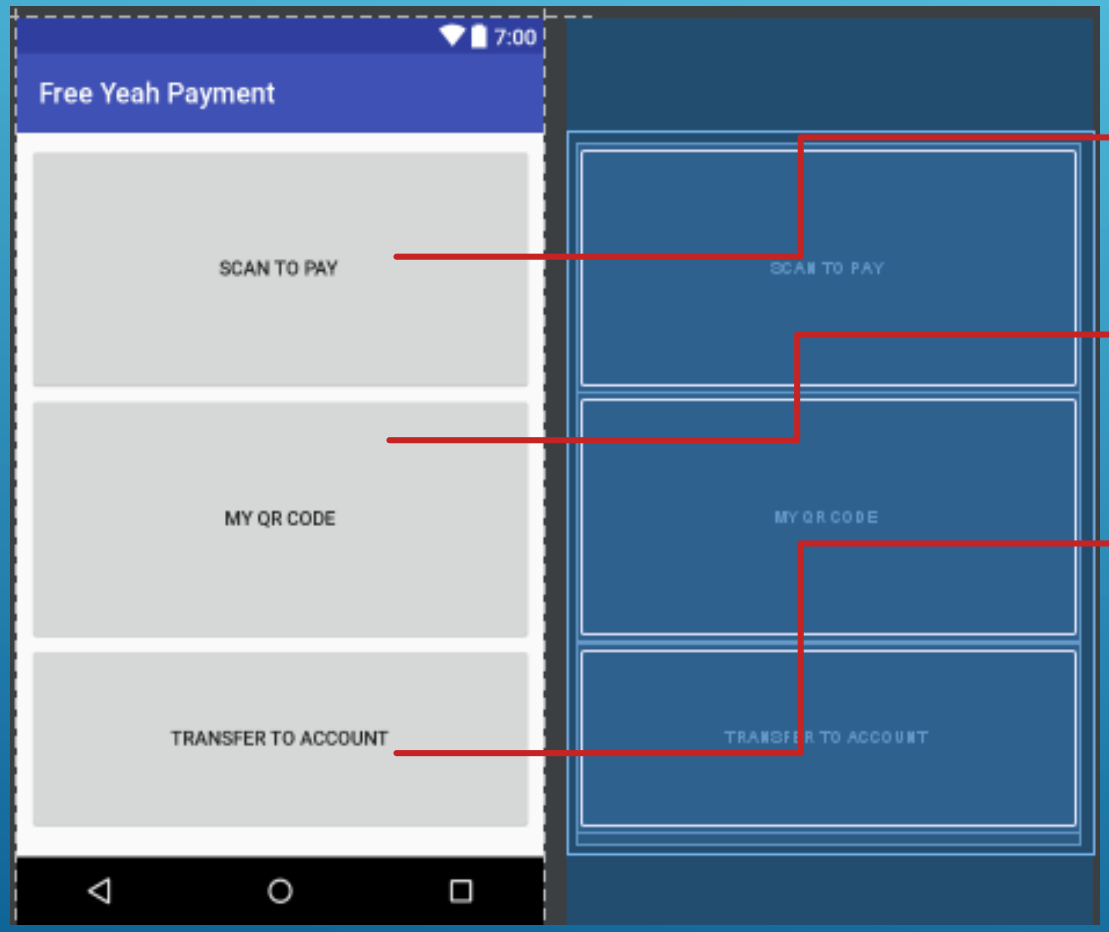
- **Home Page**



- Pay money
- Receive money

2. SYSTEM DESIGN – ANDROID APPLICATION

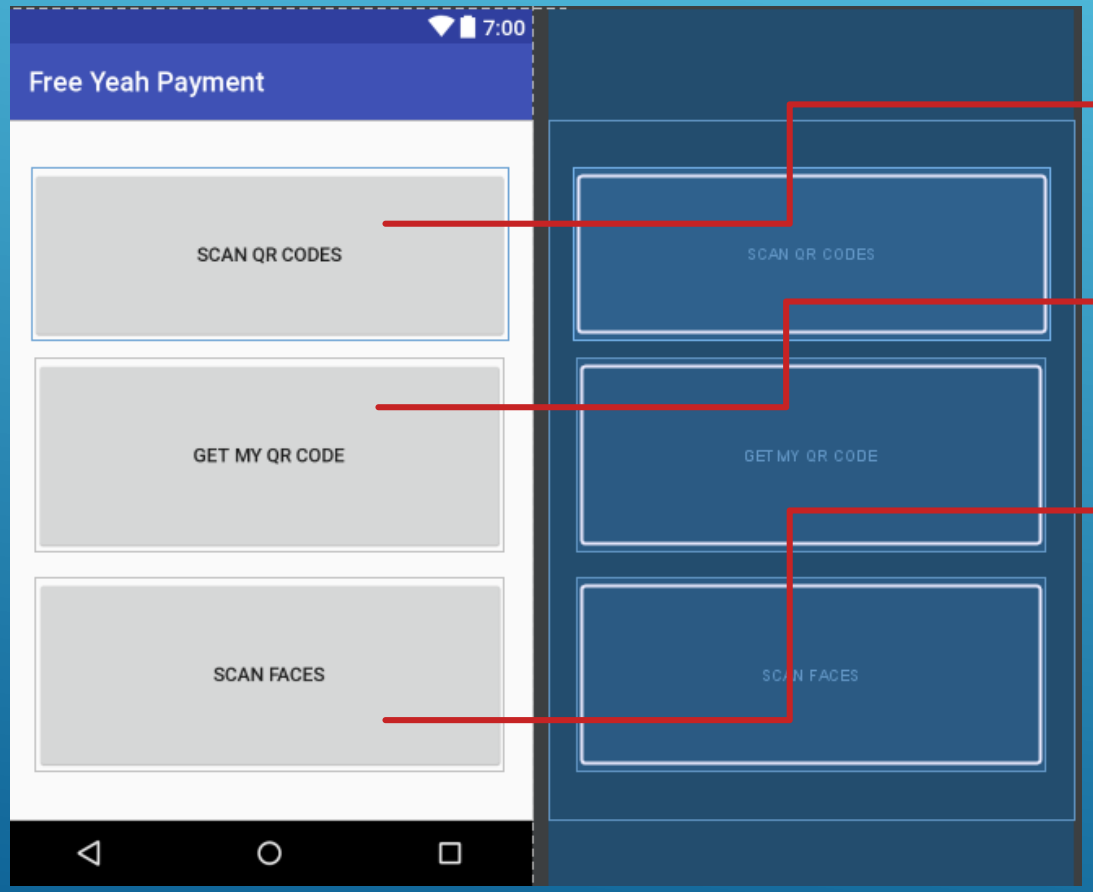
- **Payment Page**



- Scan a QR Code
- Generate a QR Code
- Pay by account

2. SYSTEM DESIGN – ANDROID APPLICATION

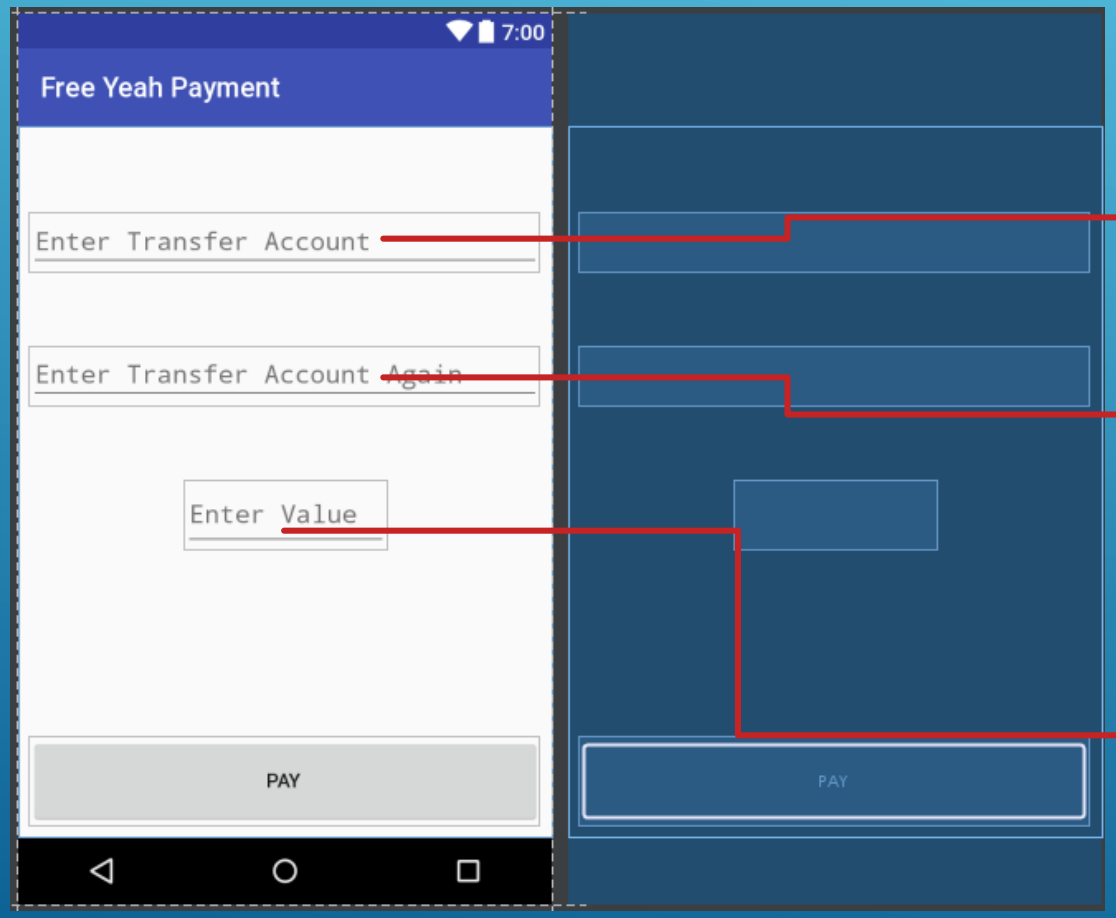
- **Gathering Page**



- Scan a QR Code
- Generate a QR Code
- Scan a Face

2. SYSTEM DESIGN – ANDROID APPLICATION

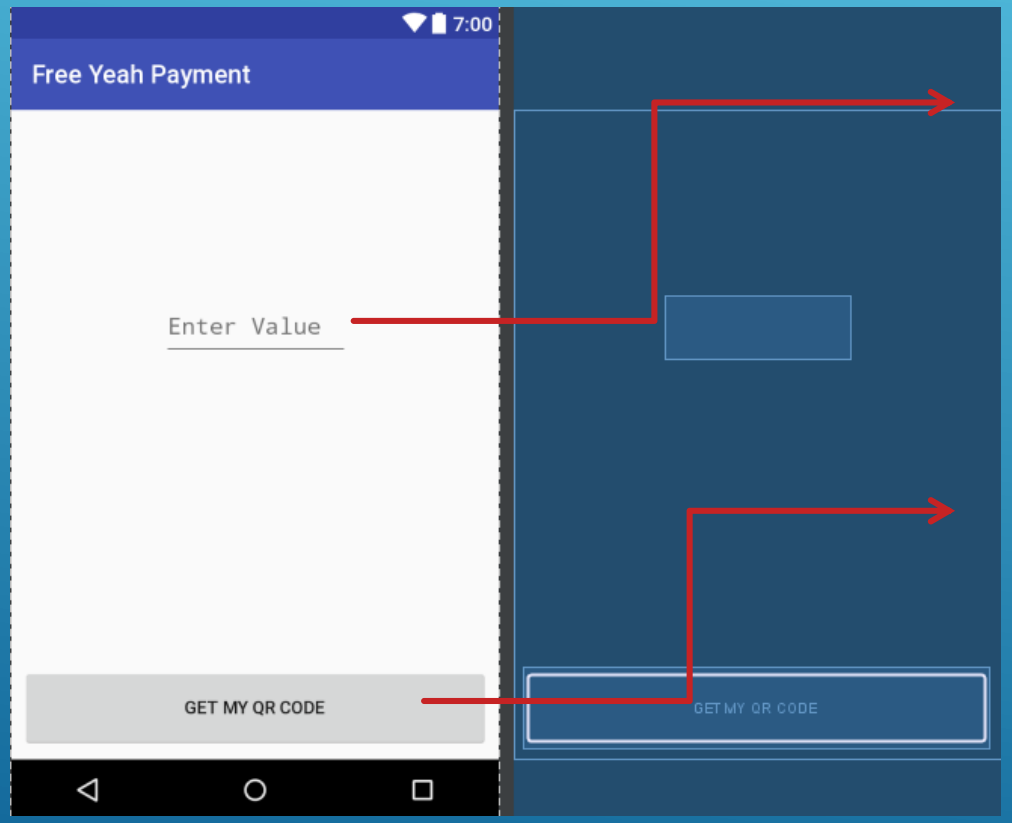
- **Payment by Account**



- Enter the target account
- Enter the target account again for check
- Enter the transfer value

2. SYSTEM DESIGN – ANDROID APPLICATION

- **Create a QR Code for Payment**



- Enter the Value for this Code
- Click the button to create

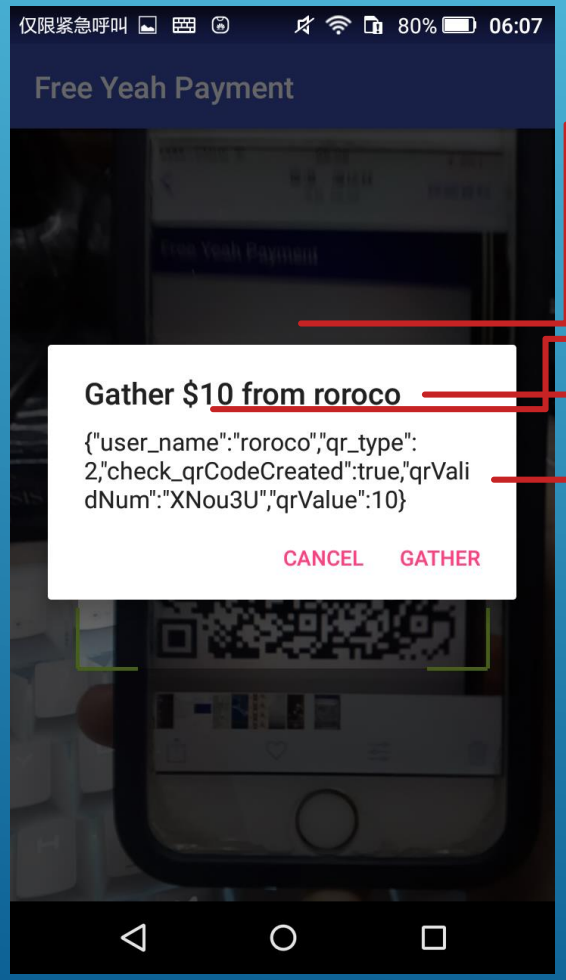
2. SYSTEM DESIGN – ANDROID APPLICATION

- **Create a QR Code for Payment**

- The QR Code
- Contain value inside

2. SYSTEM DESIGN – ANDROID APPLICATION

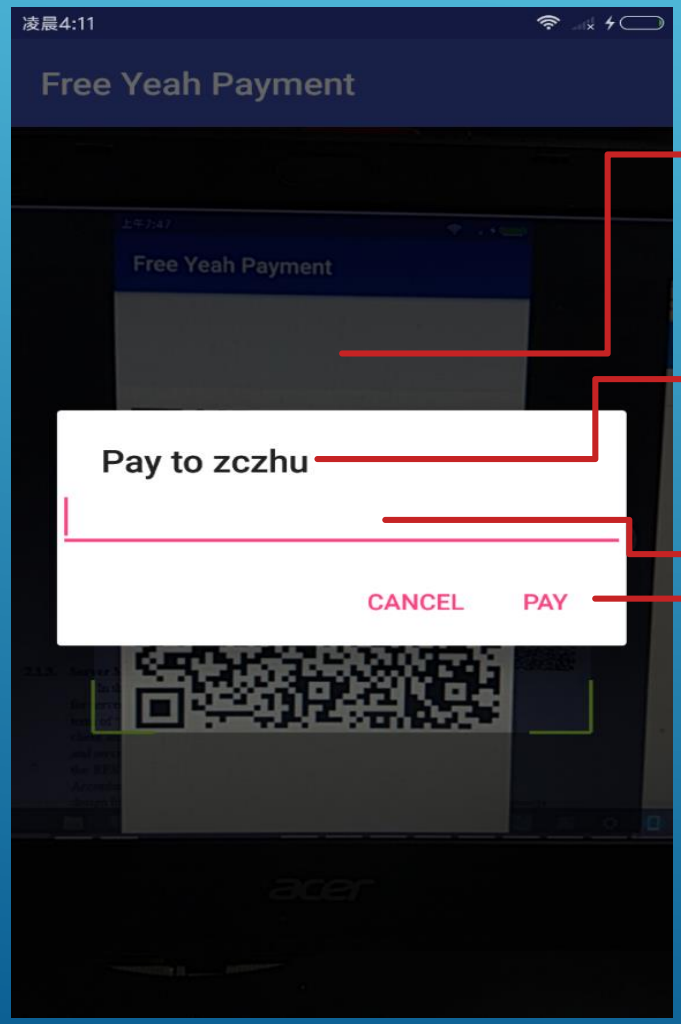
- **Gathering by Scan a QR Code**



- Scan a QR Code
- Check the target value
- Display the payer account
- Data in QR Code (only for demo here)

2. SYSTEM DESIGN – ANDROID APPLICATION

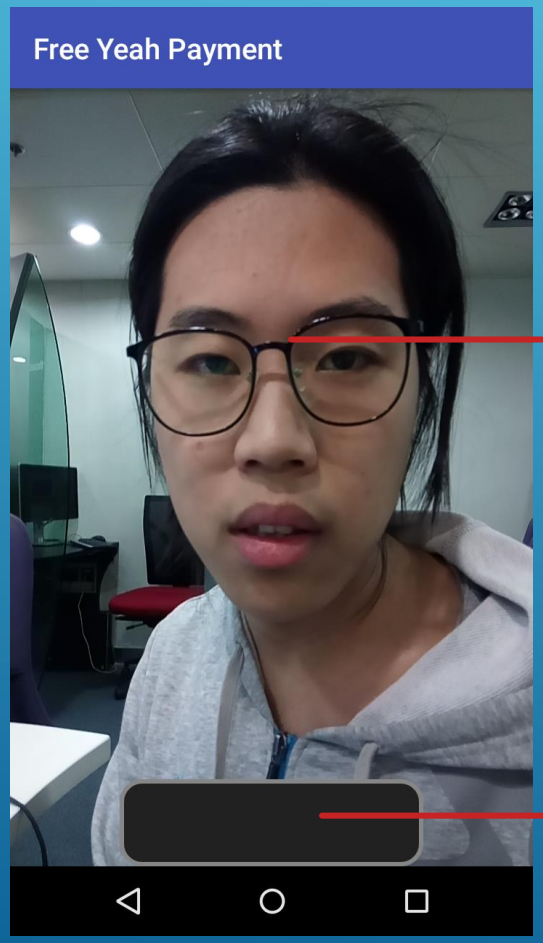
- **Payment by Scanning a QR Code**



- Scan a QR Code
- Check the target account
- Enter the value
- Confirm to pay

2. SYSTEM DESIGN – ANDROID APPLICATION

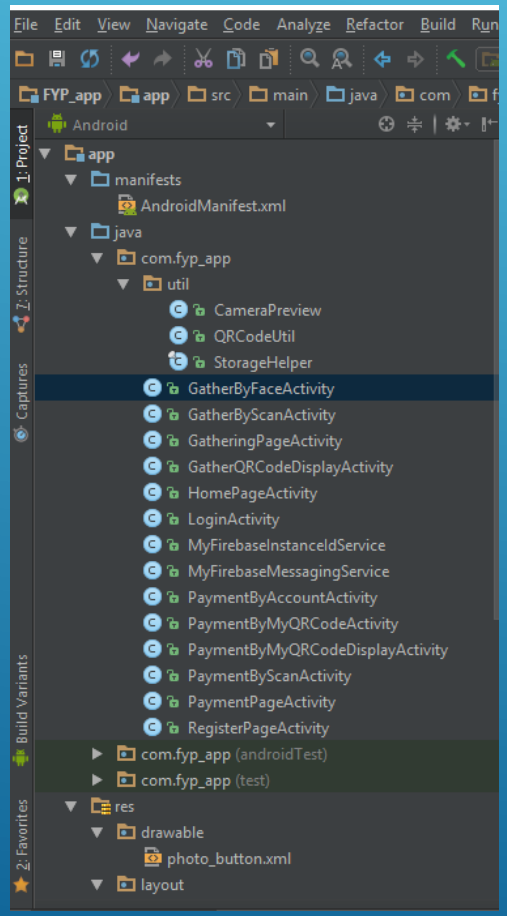
- **Gathering by Scanning a Face**

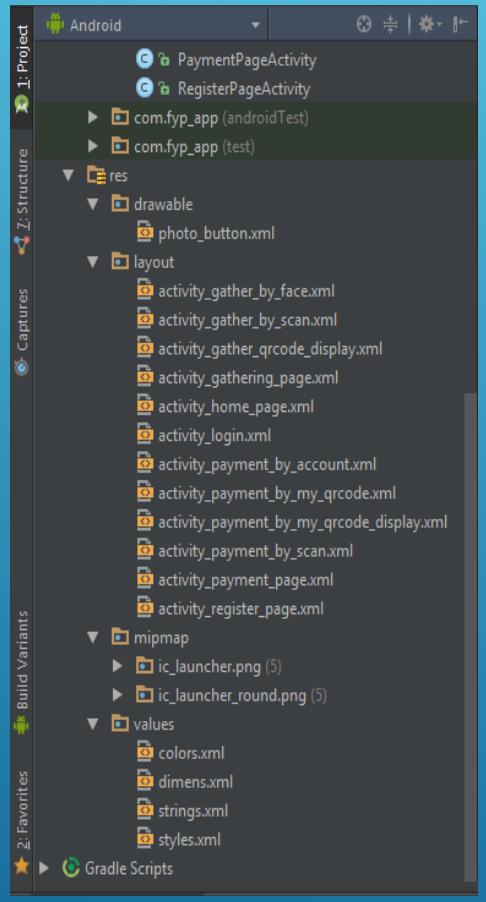


- Scan a User's Face
- Click the bar to take the photo

2. SYSTEM DESIGN – ANDROID APPLICATION

- **Programming File Structure**





2. SYSTEM DESIGN – ANDROID APPLICATION

- **Communication Method with Server**

  - **HTTP GET Request**
    - For most functions

  - **HTTP POST Request**
    - For uploading face images

2. SYSTEM DESIGN

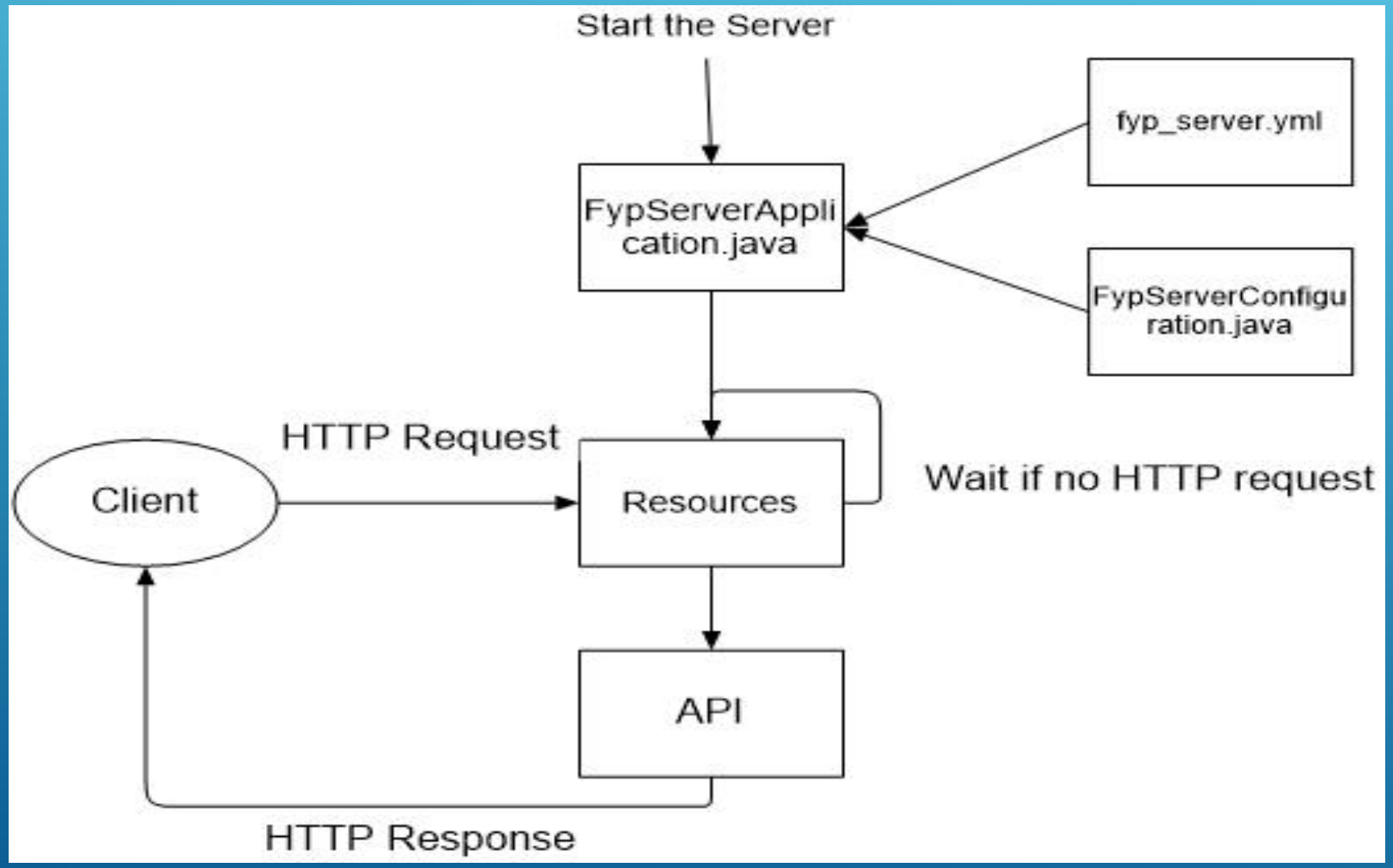
SERVER

2. SYSTEM DESIGN – SERVER

- **Development Environment for Server**
  - Programming Environment
    - Windows 10
  - IDE
    - Eclipse Ver. Mars.1 Release (4.5.1)
  - Project Management Tool
    - Maven
  - Java Version
    - Ver. 1.8.0_71

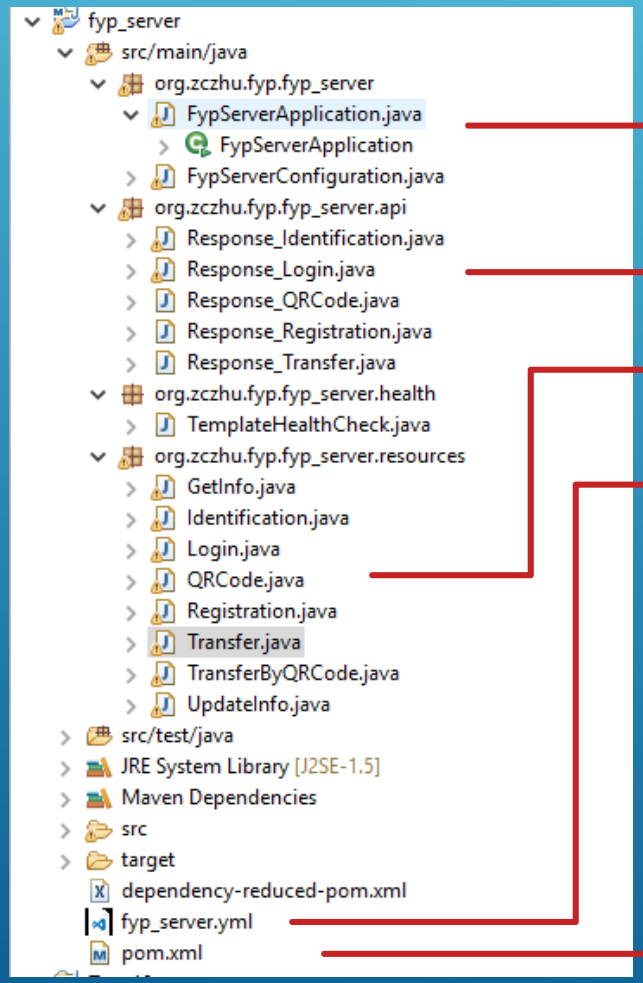
2. SYSTEM DESIGN – SERVER

- **Working Flow of the Server**



2. SYSTEM DESIGN – SERVER

- **Programming File Structure**



- Server main application
- API for response object
- Resources
- Server Port definition
- Dependencies definition

2. SYSTEM DESIGN – SERVER

- **Supported URLs**

```
GET     /getinfo/test (org.zczhu.fyp.fyp_server.resources.GetInfo)
GET     /getinfo/{username}&{type} (org.zczhu.fyp.fyp_server.resources.GetInfo)
GET     /identification/test (org.zczhu.fyp.fyp_server.resources.Identification)
POST    /identification/upload/ (org.zczhu.fyp.fyp_server.resources.Identification)
GET     /login/{username}&{password} (org.zczhu.fyp.fyp_server.resources.Login)
GET     /qrcode/{username}&{type}&{value}&{fcmToken} (org.zczhu.fyp.fyp_server.resources.QRCode)
GET     /registration/{username}&{password}&{emailaddr} (org.zczhu.fyp.fyp_server.resources.Registration)
GET     /transfer/{fromName}&{toName}&{value} (org.zczhu.fyp.fyp_server.resources.Transfer)
GET     /transferbyqrcode/{fromName}&{toName}&{value}&{type}&{qrValidNum}&{fcmToken} (org.zczhu.fyp.fyp_server.resources.TransferByQRCode)
GET     /updateinfo/{username}&{type}&{newdata} (org.zczhu.fyp.fyp_server.resources.UpdateInfo)
```

2. SYSTEM DESIGN – SERVER

- **Communication Method**
  - **With Database**
    - Java DataBase Connectivity (JDBC) Library
  - **With Android Application**
    - HTTP Response
      - JSON format
      - Defined in API files

2. SYSTEM DESIGN

DATABASE

2. SYSTEM DESIGN – DATABASE

- **Development Environment for Database**
  - Environment
    - Windows 10
  - Tool
    - MySQL Ver. 5.7.19
      - Open source, free
      - Easy to use for individual user

2. SYSTEM DESIGN – DATABASE

- **Table Structure**
  - Table fyp_user

Field	Type	Null	Key	Default	Extra
user_id	int(5)	NO	PRI	NULL	auto_increment
user_name	varchar(10)	NO	UNI		
user_password	varchar(20)	NO		NULL	
user_emailAddr	varchar(20)	NO			
user_balance	int(10)	NO		0	
user_createTime	timestamp(6)	NO		CURRENT_TIMESTAMP(6)	
user_lastModifiedTime	timestamp(6)	NO		CURRENT_TIMESTAMP(6)	on update CURRENT_TIMESTAMP(6)
user_valid	char(1)	NO		1	
user_bankAccount	varchar(20)	NO			
user_qrExpired	timestamp(6)	NO		CURRENT_TIMESTAMP(6)	
user_ip	varchar(16)	YES		0.0.0.0	
user_qrValidNum	varchar(6)	NO		xxxxxx	
user_qrValue	int(10)	NO		-1	
user_fcmToken	longtext	YES		NULL	

2. SYSTEM DESIGN – DATABASE

- **Table Structure**
  - Table fyp_trans

Field	Type	Null	Key	Default	Extra
trans_id	int(20)	NO	PRI	NULL	auto_increment
trans_fromID	int(5)	NO		0	
trans_toID	int(5)	NO		0	
trans_fromName	varchar(10)	NO		NULL	
trans_toName	varchar(10)	NO		NULL	
trans_fromBalance	int(10)	NO		0	
trans_toBalance	int(10)	NO		0	
trans_value	int(10)	NO		0	
trans_createTime	timestamp(6)	NO		CURRENT_TIMESTAMP(6)	

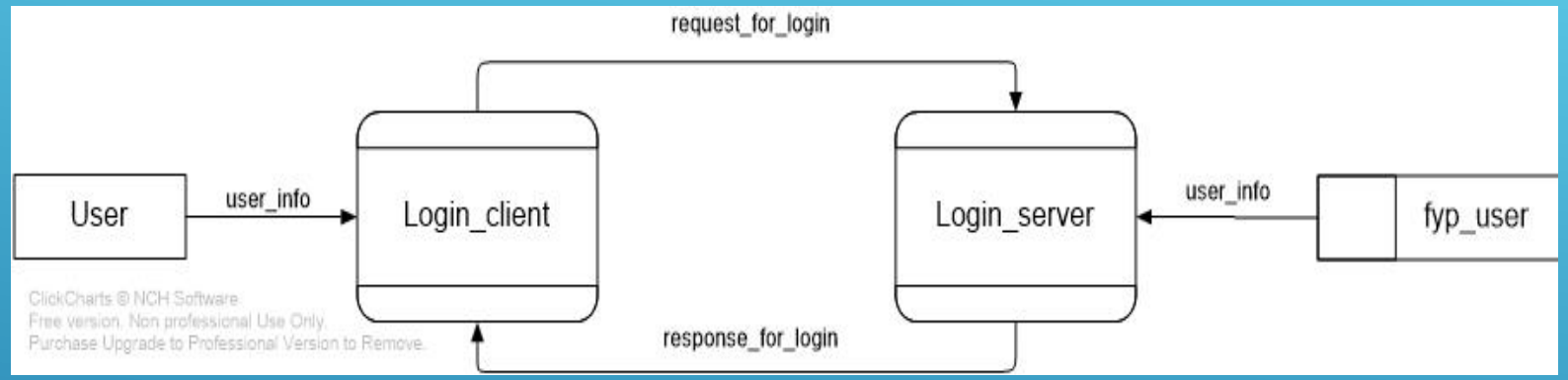
3. FUNCTION DESIGN

3. FUNCTION DESIGN – FUNCTION LIST

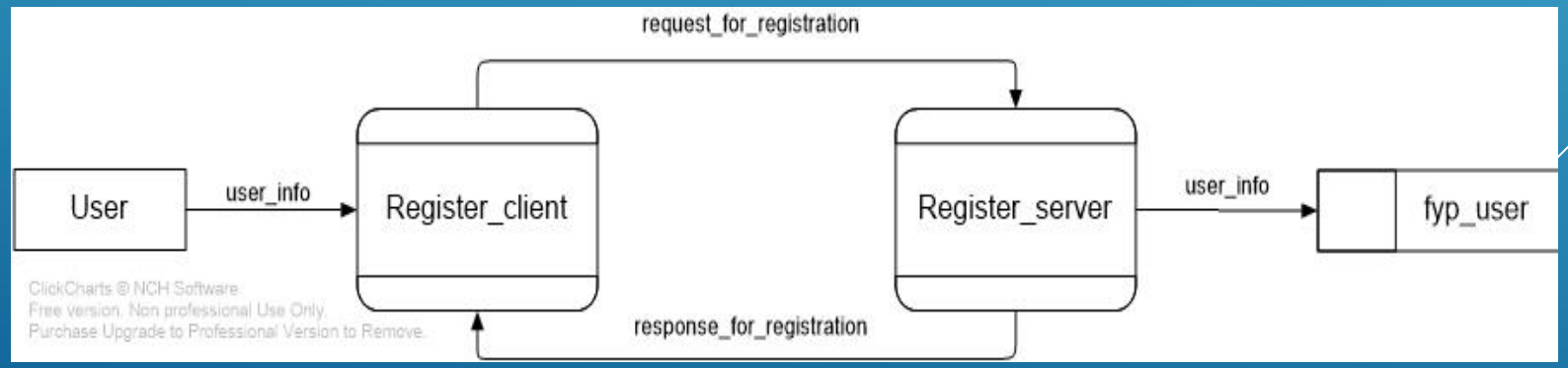
Function Name	Description
Login	Users log in with necessary information
Registration	Users register with necessary information
Payment_byScan	Users scan a QR Code to pay. Transfer value input is required
Payment_byQRCode	Users generate a QR Code which can be scanned by others. Transfer value input is required before generating
Payment_byAccount	Users directly type in target and value to transfer
Gathering_byScan	Users scan a QR Code for gathering. Transfer value is defined in the QR Code
Gathering_byQRCode	Users generate a QR Code which can be scanned by others to transfer
GatherbyFace	Users scan a person's face, who has already registered as a member and pre-upload the face images, and then the target user who has been scanned will receive a notification, which allows the target user to make the payment.
GCM notification	Users received notifications when transactions happened.

3. FUNCTION DESIGN – WORKING FLOW

- **Login**

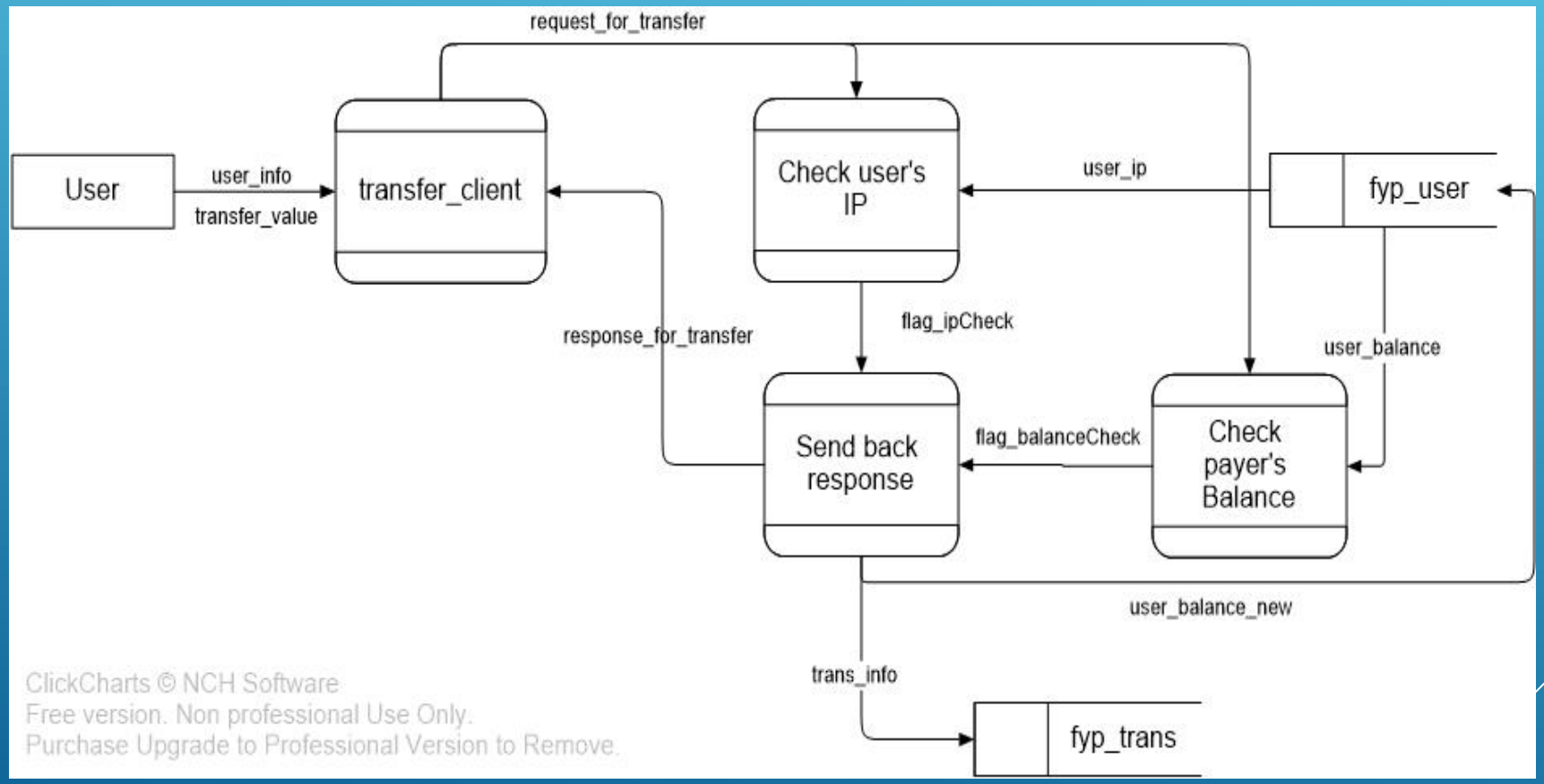


- **Register**



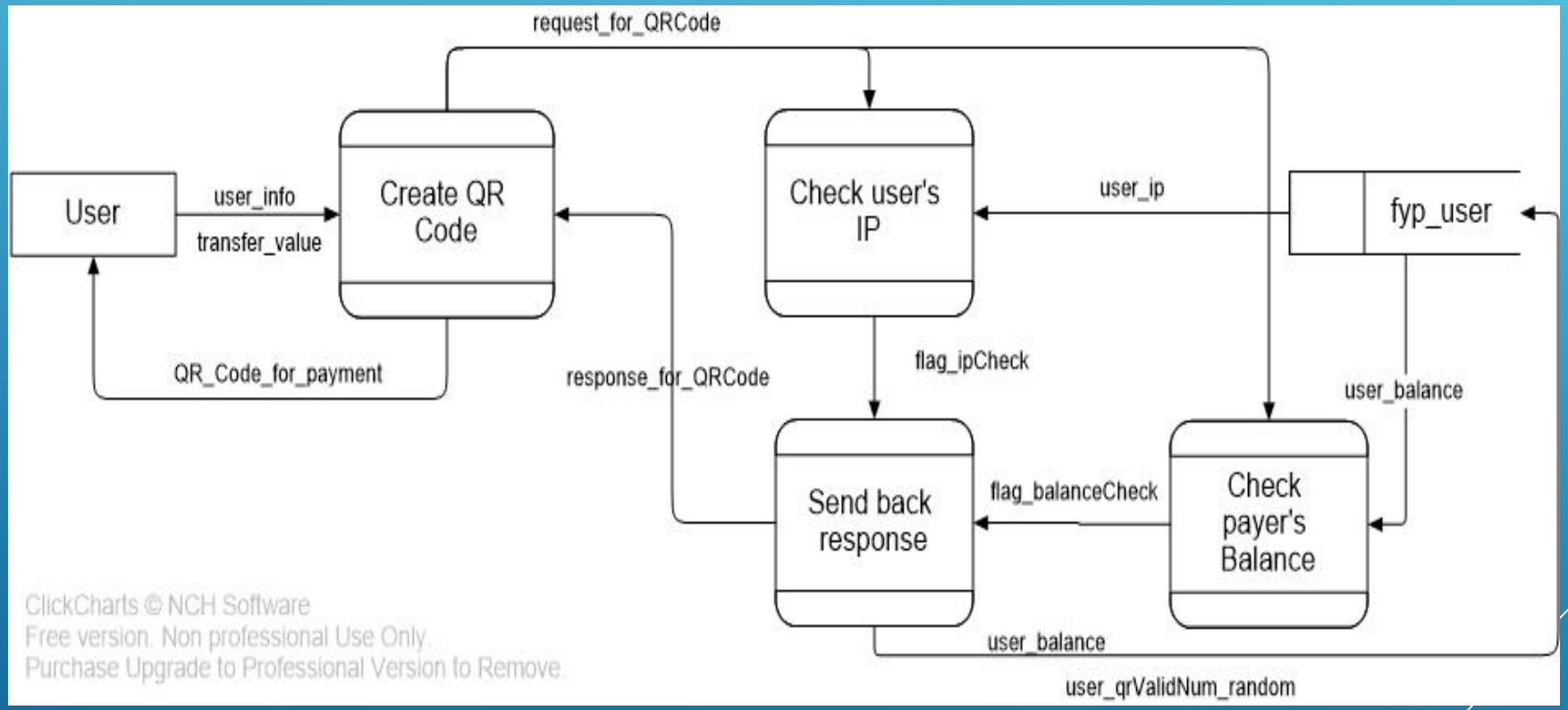
3. FUNCTION DESIGN – WORKING FLOW

- **Payment by Transfer**



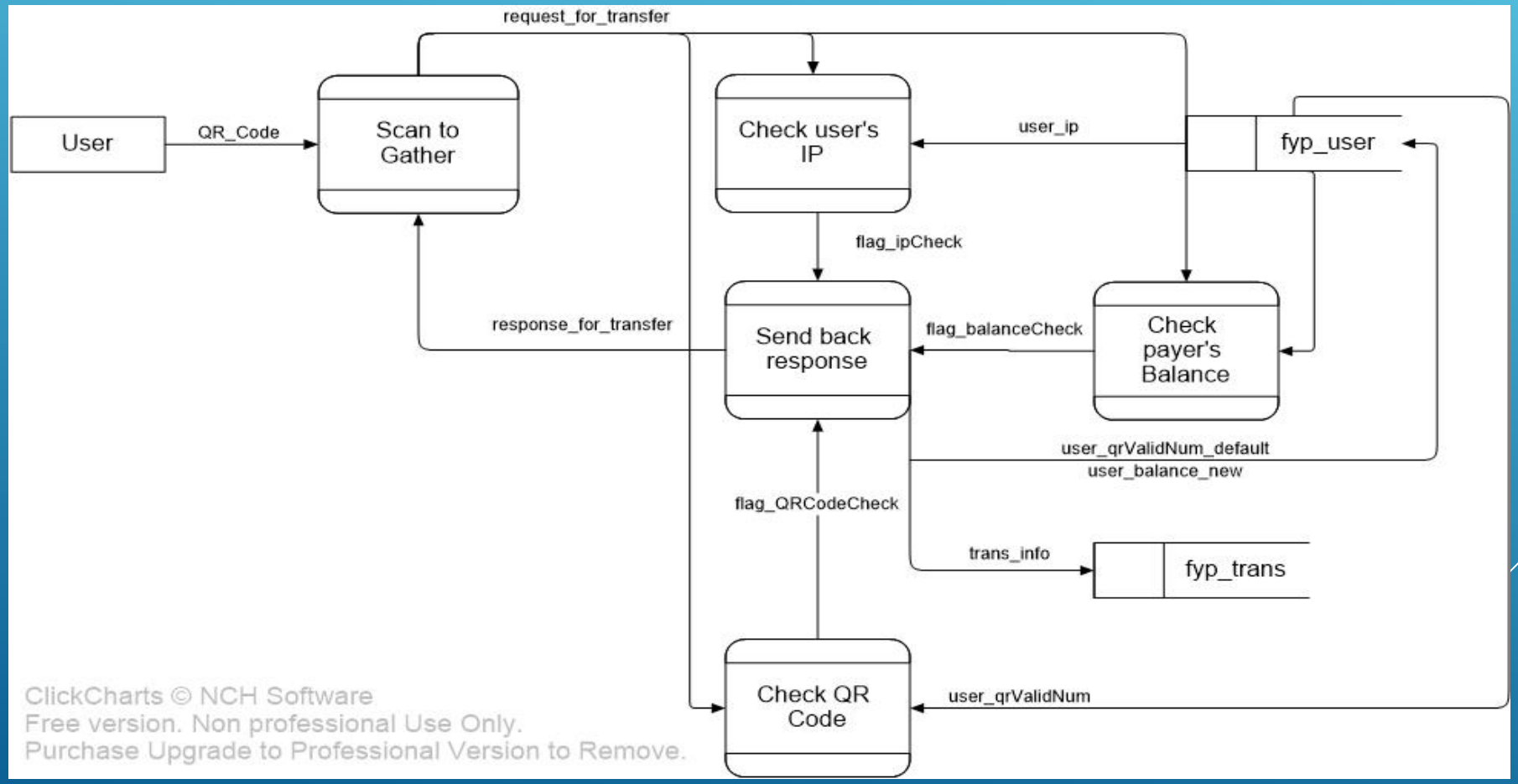
3. FUNCTION DESIGN – WORKING FLOW

- **Create a QR Code**



3. FUNCTION DESIGN – WORKING FLOW

- **Scan a QR Code**



3. FUNCTION DESIGN

PAY BY FACE

3. FUNCTION DESIGN – PAY BY FACE

- **Pre-requirements**
  - **Google Play Service is on**
  - **Camera is supported and permitted**
  - **Target user has registered as a member**
  - **Target user has uploaded 10 images**

3. FUNCTION DESIGN – PAY BY FACE

Image Processing

Grayscale
Size [92*112]

3. FUNCTION DESIGN – PAY BY FACE

Image Transfer

HTTP POST

FYP SERVER

Predication ID

HTTP POST

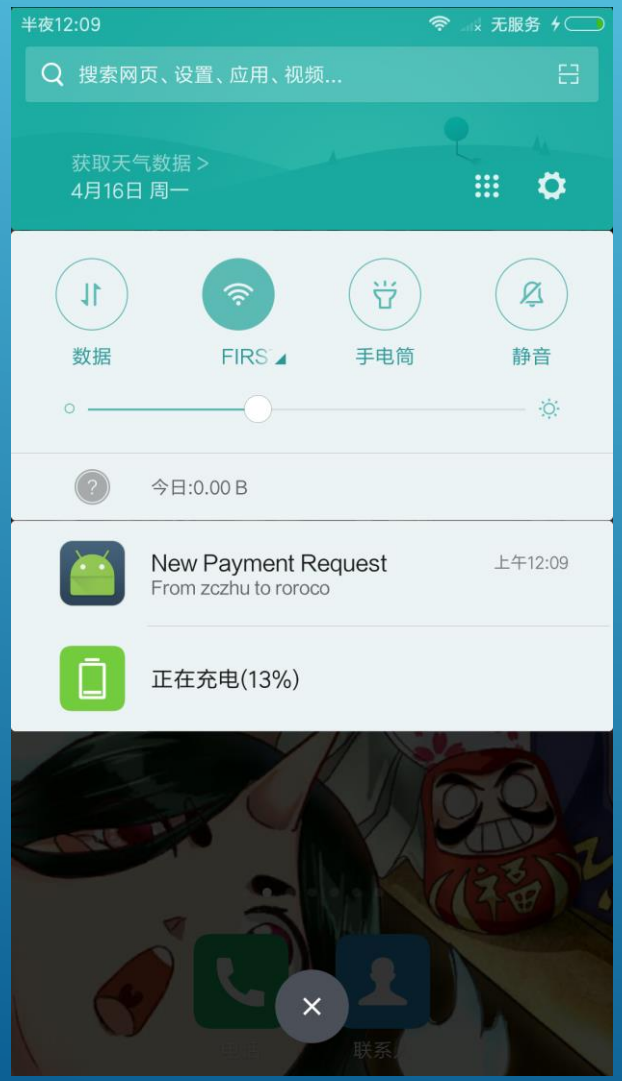
GCM SERVER

Notification

User Device

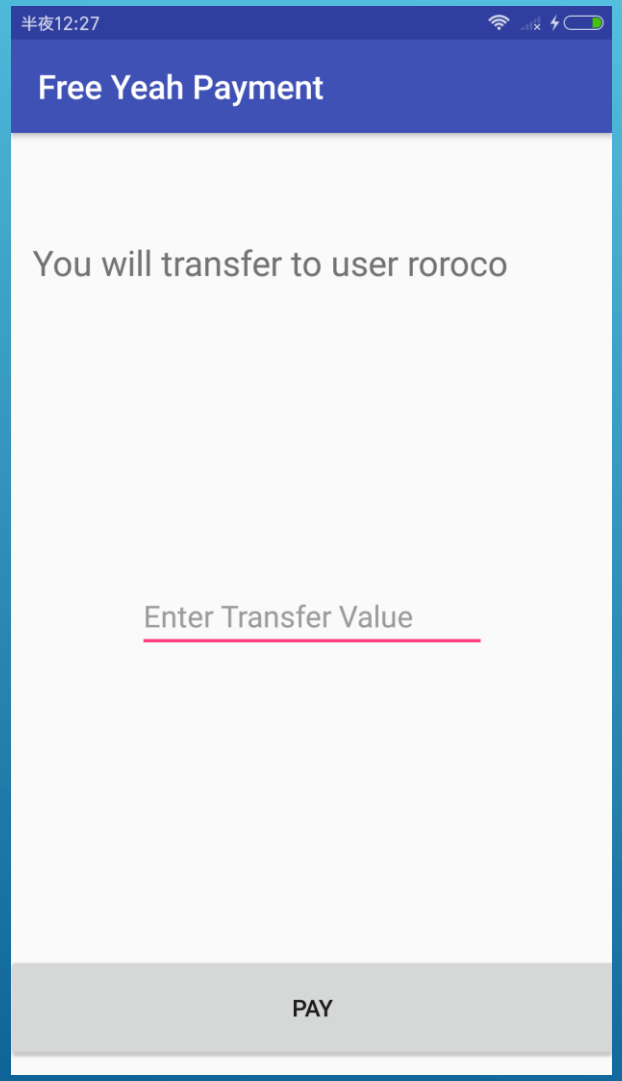
Encode by Base64

3. FUNCTION DESIGN – PAY BY FACE



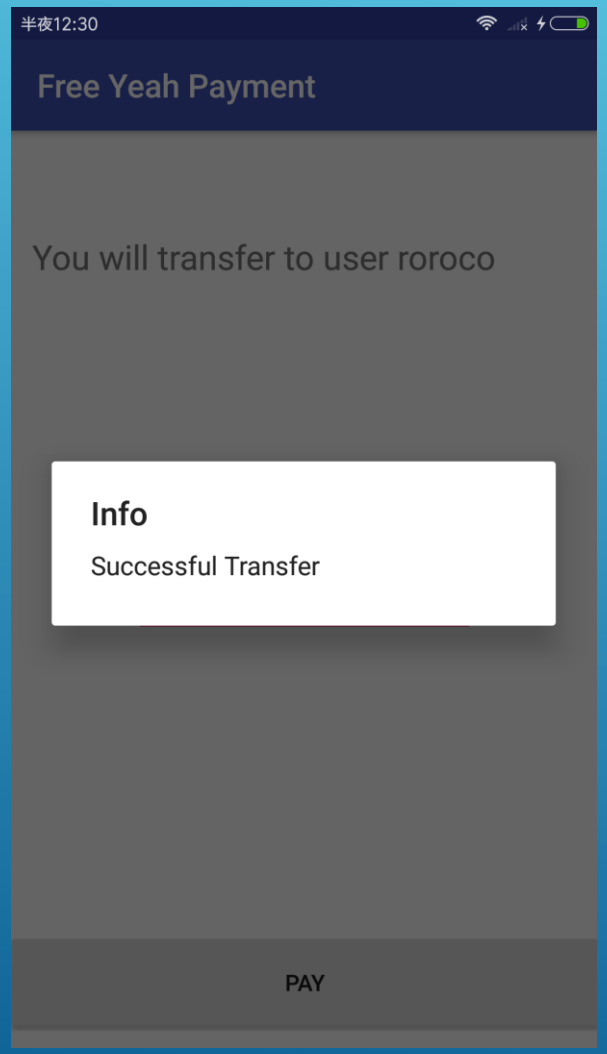
- **Target user receives a notification**
- **Ignore the notification if incorrect**
- **Click the notification to pay**

3. FUNCTION DESIGN – PAY BY FACE



- **Specify the transfer user name**
- **Enter the value**
- **Click the button to pay**

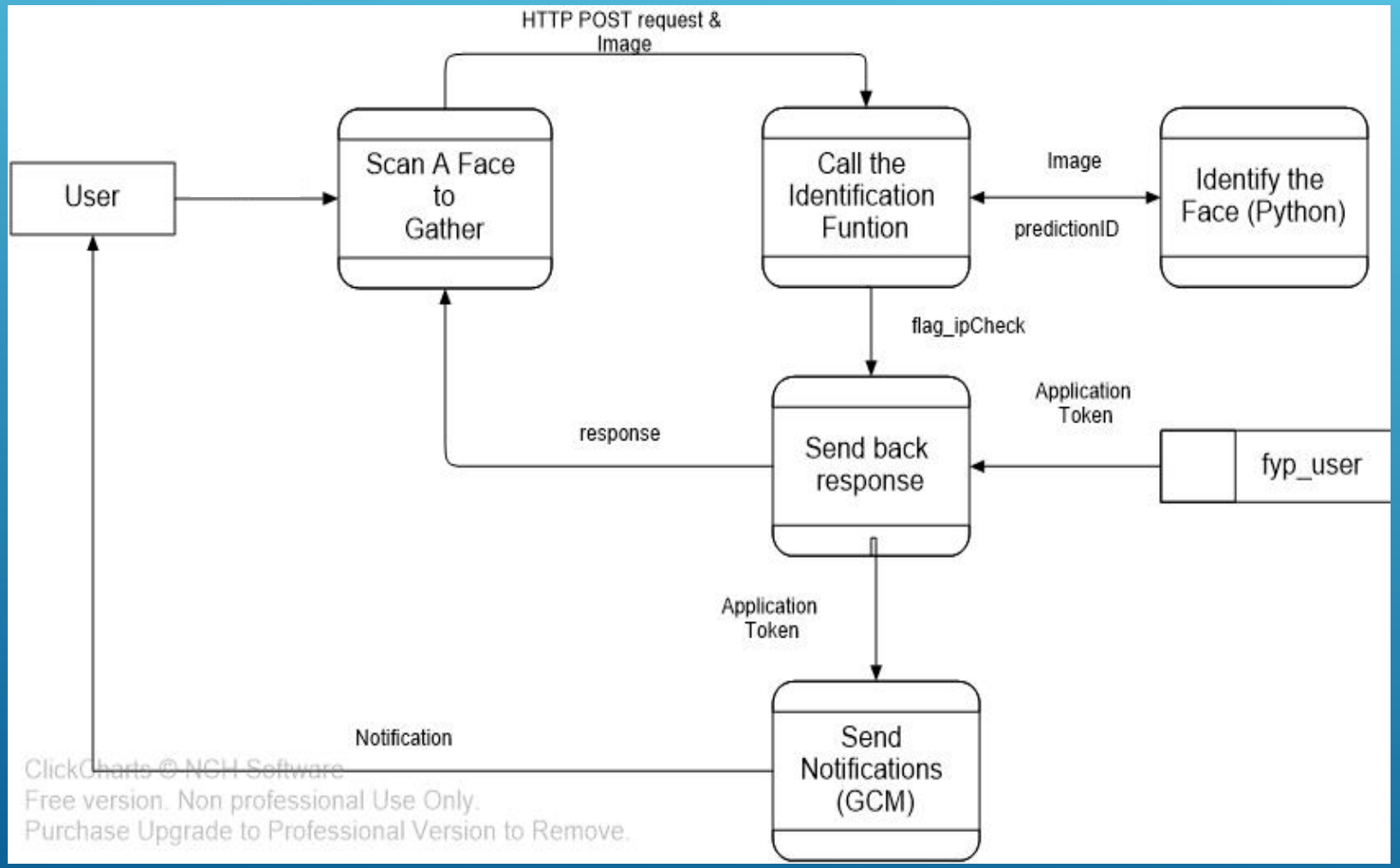
3. FUNCTION DESIGN – PAY BY FACE



- **The dialog shows the transaction status**

3. FUNCTION DESIGN – WORKING FLOW

- **Scan a Face**



3. FUNCTION DESIGN – PAY BY FACE

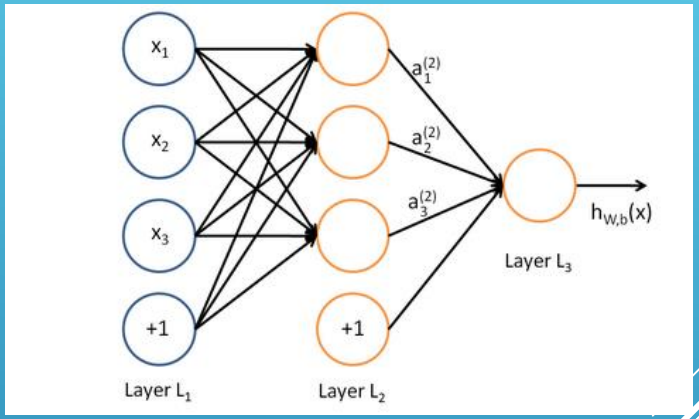
- **Identification Method**
  - Eigenface – Principle Component Analysis
    - Ref. M. Turk and A. Pentland (1991). "Face recognition using eigenfaces" (PDF). Proc. IEEE Conference on Computer Vision and Pattern Recognition. Pp. 586–591.
  - Artificial Neural Network Approach
- **Construction Environment**
  - Tensorflow
    - Deep Learning Tool developed by Google
    - Open source
  - Python 3.6

3. FUNCTION DESIGN – PAY BY FACE

- **File Structure**
  - **model.py**
    - Run to train & save the model
  - **restore.py**
    - Run to restore the saved model
    - And make the identification on images
  - **/Models**
    - Models and variables, graphs are stored
  - **/FaceDatabase**
    - All the users' face images and images for identification are stored

3. FUNCTION DESIGN – PAY BY FACE

- **Neural Network Structure**
  - 2 Layers
    - Input layer & hidden layer
  - Input Nodes
    - determined by PCA
  - Hidden Nodes
    - determined by database size
  - Output Nodes
    - determined by number of users



3. FUNCTION DESIGN – PAY BY FACE

- **Neural Network Structure**
  - Active function for layers
    - Rectified Linear Unit (ReLU) function
  - Loss function
    - Softmax & cross-entropy
  - Optimizer
    - Gradient Descent Optimizer
  - Learning rate
    - Manually justify

3. FUNCTION DESIGN – PAY BY FACE

- **Sample Test – Training Settings**
  - 10 people
  - 10 images for each
  - 60% for training & 40% for testing
  - 50 hidden nodes
  - learning rate is 0.01
  - training literation number is 5000

3. FUNCTION DESIGN – PAY BY FACE

- **Sample Test – Training Result**

```
0.26331827
0.26330757
0.2632965
0.26328528
0.26327488
The accuracy is
0.975
Saving Completed
PS D:\Files\FYP_Project\Server\identificationFile\Codes>
```

3. FUNCTION DESIGN – PAY BY FACE

- **Sample Test – Identification Result**

- Test Case 1
  - Image:

  - Result:

```
192.168.137.10 - - [09/Apr/2018:04:23:22 +0000] "POST /identification/upload
true
[[6.152719  0.        0.4921527 0.        4.130851  0.3715695 0.
  0.        0.        0.       ]]
1
```

3. FUNCTION DESIGN – PAY BY FACE

- **Sample Test – Identification Result**

- Test Case 2
  - Image:

  - Result:

```
192.168.137.10 - - [09/Apr/2018:04:24:01 +0000] "POST /identification/upload/
true
[[5.699717   0.05348476 0.6991321  0.         3.6150718  0.00730522
  0.         0.         0.         0.        ]]
1
```

3. FUNCTION DESIGN – PAY BY FACE

- **Sample Test – Identification Result**

- Test Case 3
  - Image:

  - Result:

```
192.168.137.10 - - [09/Apr/2018:04:26:01 +0000] "POST /identification/upload/ HT
true
[[6.392293   0.26577845 0.94909364 0.         3.6844635  0.
  0.         0.         0.         0.        ]]
1
```

4. FURTHER IMPROVEMENTS

4. FURTHER IMPROVEMENT

- Security
  - URL Security
  - QR Code Security

- Face Identification
  - Capacity of the Model
  - Processing the Failed Identification
  - Retraining of the Model

5. CONCLUSIONS

5. CONCLUSIONS

- The whole system was completely implemented
- Multiple payment methods are supported
- Still many challenges on applying face identification to real-life electronic payment system

6. Q & A