Software Requirements Specification

For

Face Recognition System
[Facilitator Module]

Prepared By Naïve Team

School of computer science and technology
Jilin University

2016/4/15

Table of Contents

1. Introduction

1.1 Purpose

This SRS describes the module functional and nonfunctional requirements for version 1.0 of the Facilitator Module. This document is intended to be used by team members whom will implement and verify the system.

1.2 Product Scope

Facilitator Module is a module in Face recognition project; to create software for helping instructor associate student's name with their face and keep track of class attendance record. A client software could use a camera to capture student's picture in a classroom and display their names in real-time on instructor's desktop/laptop. Client software could then send a picture to record student who attend a class. The attendance record could then available for instructor and students.

Facilitator receive the request to recognize the student attending the class from Client then send request to Server to let them recognize the student face and return its information back to Client.

1.3 References

SRS template from IE (http://github.com)

2. Overall Description

2.1 Module Functions

Facilitator use REST Protocol with JSON format to send and receive data between client and server. Facilitator may send the training request to the servers at the first time to train let servers knowing student face or send the recognition request to server to detect the student face in a picture.

To train the server, Facilitator sends the message in format "TRAIN<URL>" which the URL contains a set of images (3-5 pictures per student), student id, student name, and university code to server.. The results of training will be return to Facilitator as well.

To recognize the student face, Facilitator receive a picture in binary file from client then store in Facilitator storage then sends recognition request with image URL to server. Facilitator gets the recognition result from each server which includes the student information and their location in an image. After Facilitator analyzed and the best result for each face has been selected, Facilitator will sends those information back to client.

2.2 Context Diagram

2.2.1 Top-level data flow diagram

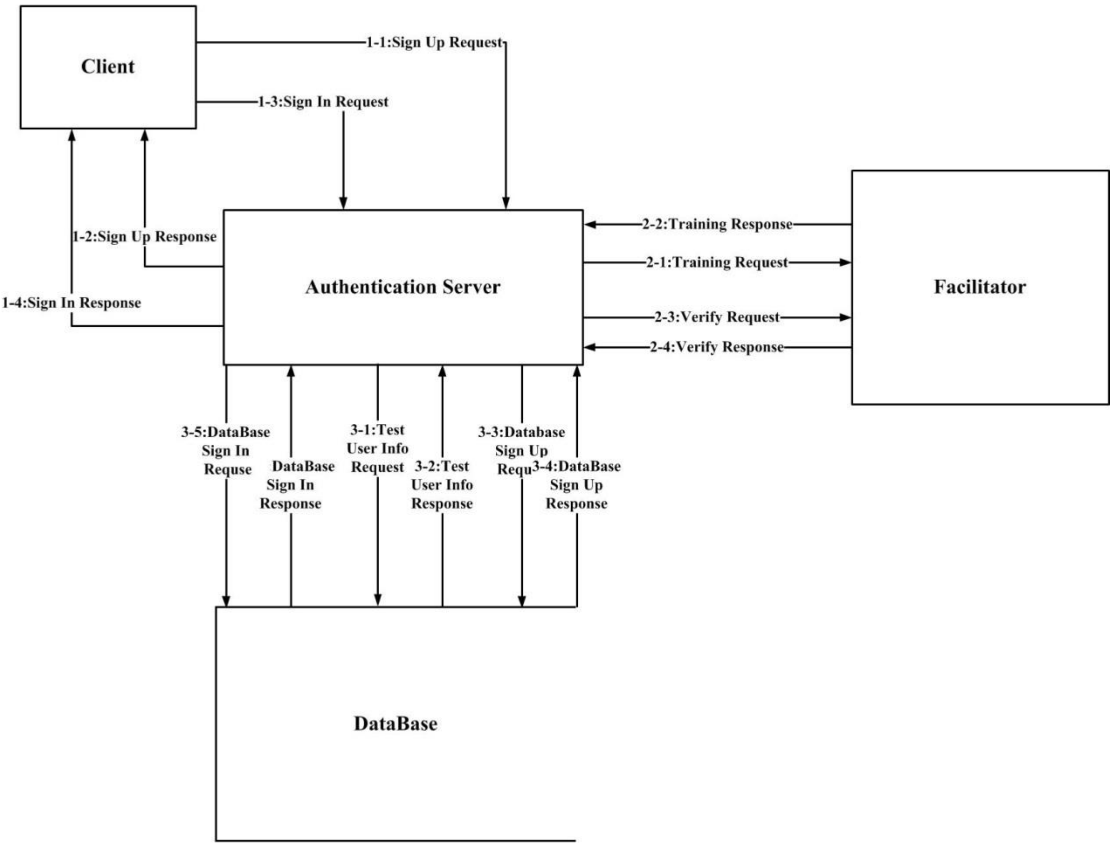


Figure 1: Top-level data flow diagram

Figure 1 description

1. 1-1:Sign Up Request Data Flow, contains the user name, password MD5 and picture, which is the data flow that client sent to the authentication server.

2. 1-2:Sign Up Response Data Flow, contains the return code and a flag represent the successful or failed, which is the data flow that client sent to the authentication server.

3. 1-3:Sign in Request Data Flow which is the same as the 1-1:Sign Up Request Data Flow.

4. 1-4:Sign in Response Data Flow which is the same 1-2:Sign Up Response Data Flow.

5. 2-1:Training Request Data Flow, which contains 5 picture's url.

6. 2-2:Training Response Data Flow, which contains return code and identifiers and the identifiers consist of the alpha and the digit, besides this it's length must be between 1 and 32.

7. 2-3:Verify Request Data Flow which contains the image's url and identifiers and the

identifiers is the same as the 2-2:Training Response Data Flow's identifiers.

8. 2-4:Verify Response Data Flow consist of return code and the flag of success, where the return code is the same as the 1-2:Sign Up Response Data Flow.

9. 3-1:Test User Info Request Data Flow contains the user name and password MD5.

10. 3-2:Test User Info Response Data Flow contains the return code and the flag of success.

11. 3-3:DataBase Sign Up Request Data Flow contains the user name, password MD5 and the Identifiers which length is over 1.

12. 3-4:DataBase Sign Up Response contains the return code and the flag of success.

13. 3-5:DataBase Sign In Request contains the only user name.

14. 3-6:DataBase Sign In Response contains password MD5 and identifiers, which's length is over 1, and besides this it has the return code too.

All of the over data flow is design to the json type.

2.2.2 Zero-level data flow diagram

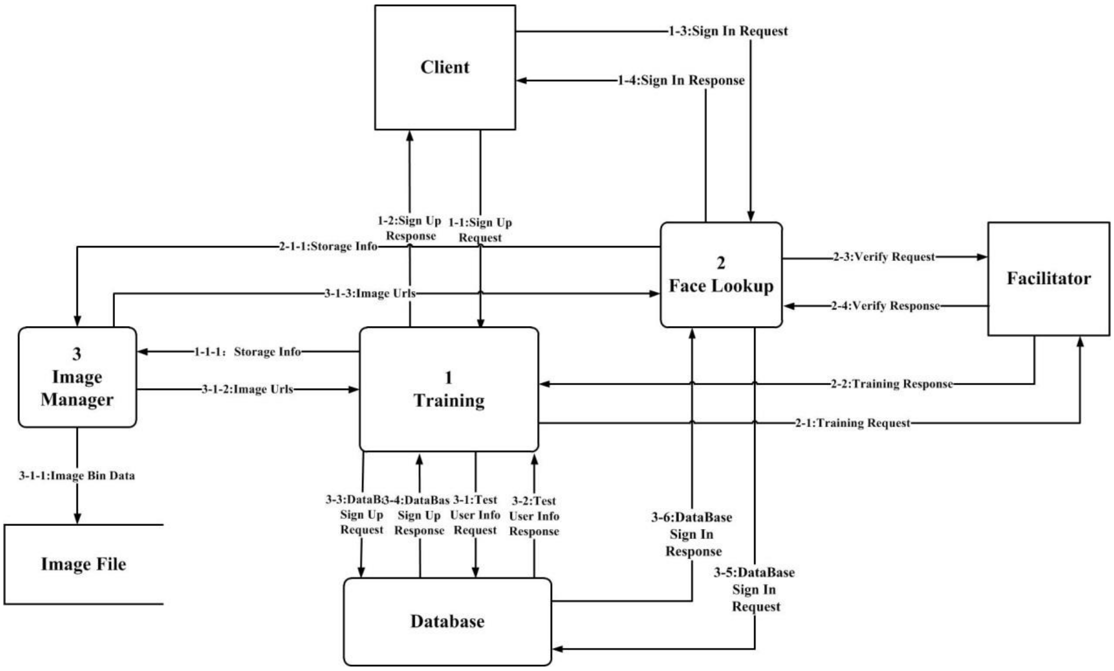


Figure 2: Zero-level data flow diagram

Figure 2 description

1. 1-1-1:Storage Info Data Flow, which contains information of need to be storage.

2. 2-2-1:Storage Info Data Flow, which is the same as the 1-1-1:Storage Info Data Flow.

3. 3-1-1:Image Bin Data Flow, which is the binary data flow of image that description the face of user.

4. 3-1-2:Image Urls Data Flow, which is locations of image in the internet.

5. 3-1-3:Image Urls Data Flow, which is the same as the 3-1-2:Image Urls Data Flow.

2.2.3 Training Module data flow diagram

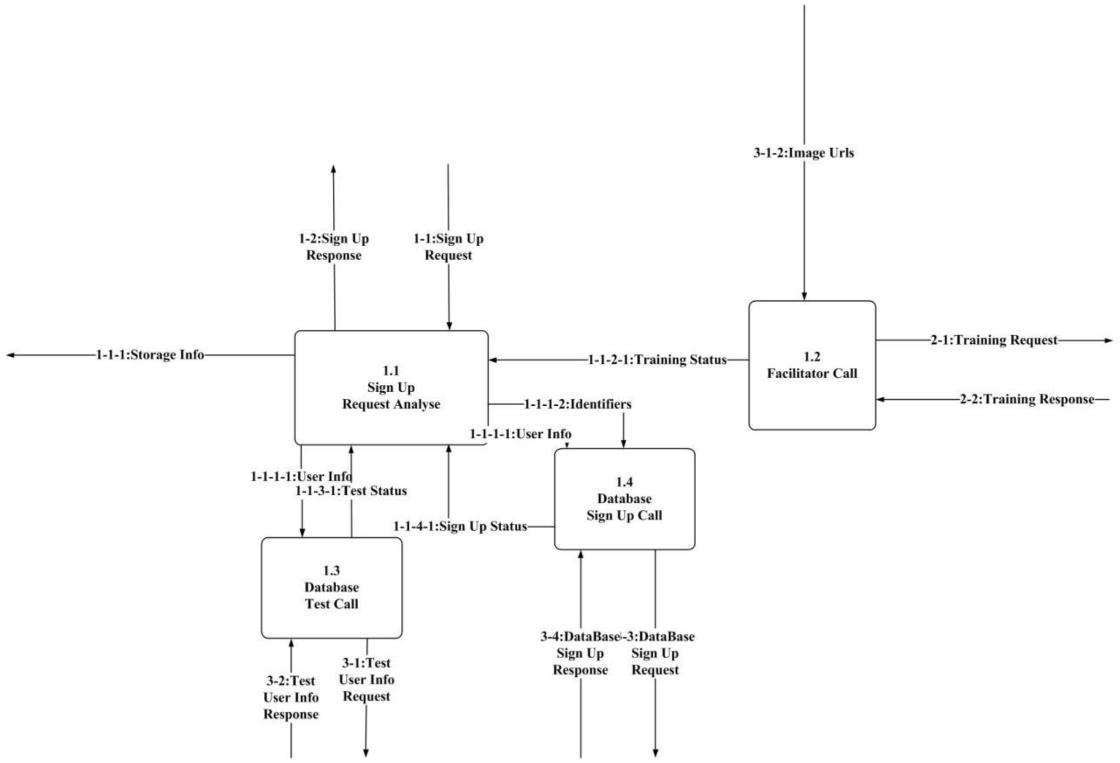


Figure 3: Training Module data flow diagram

Figure 3 description

1. 1-1-1-1:User Info Data Flow, which contains the information of the user that the database need.

2. 1-1-1-2:Identifiers Data Flow, which consist of alpha and digit and it's length is between 1 and 32.

3. 1-1-2-1:Training Status Data Flow, which consist of a flag that represent the true or false.

4. 1-1-3-1:Test Status Data Flow, which is the same as 1-1-2-1:Training Status Data Flow.

5. 1-1-4-1:Sign Up Status Data Flow, which is the same as 1-1-2-1:Training Status Data Flow.

2.2.4 Face-LookUp Module data flow diagram

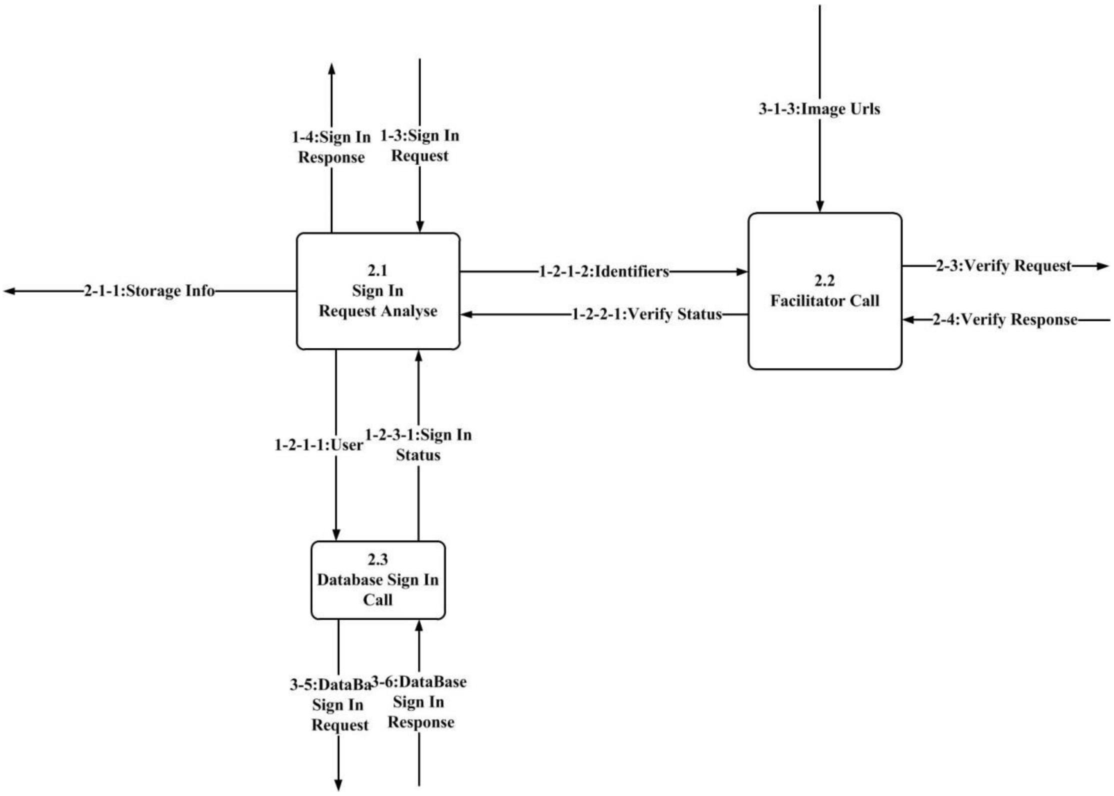


Figure 4: Face-LookUp Module data flow diagram

Figure 4 description

1. 1-2-1-1:User Info Data Flow, which contains the information that the database need to be preservation.

2. 1-2-1-2:Identifiers Data Flow, which is consist of alpha and digit and it’s length is between 1 and 32.

3. 1-2-2-1:Verify Status, which is consist of a flag that represent the true or false.

4. 1-2-3-1:Sign In Status, which is consist of a flag that represent the true of false.

2.2.5 Sign Up Request Analyze Module data flow diagram

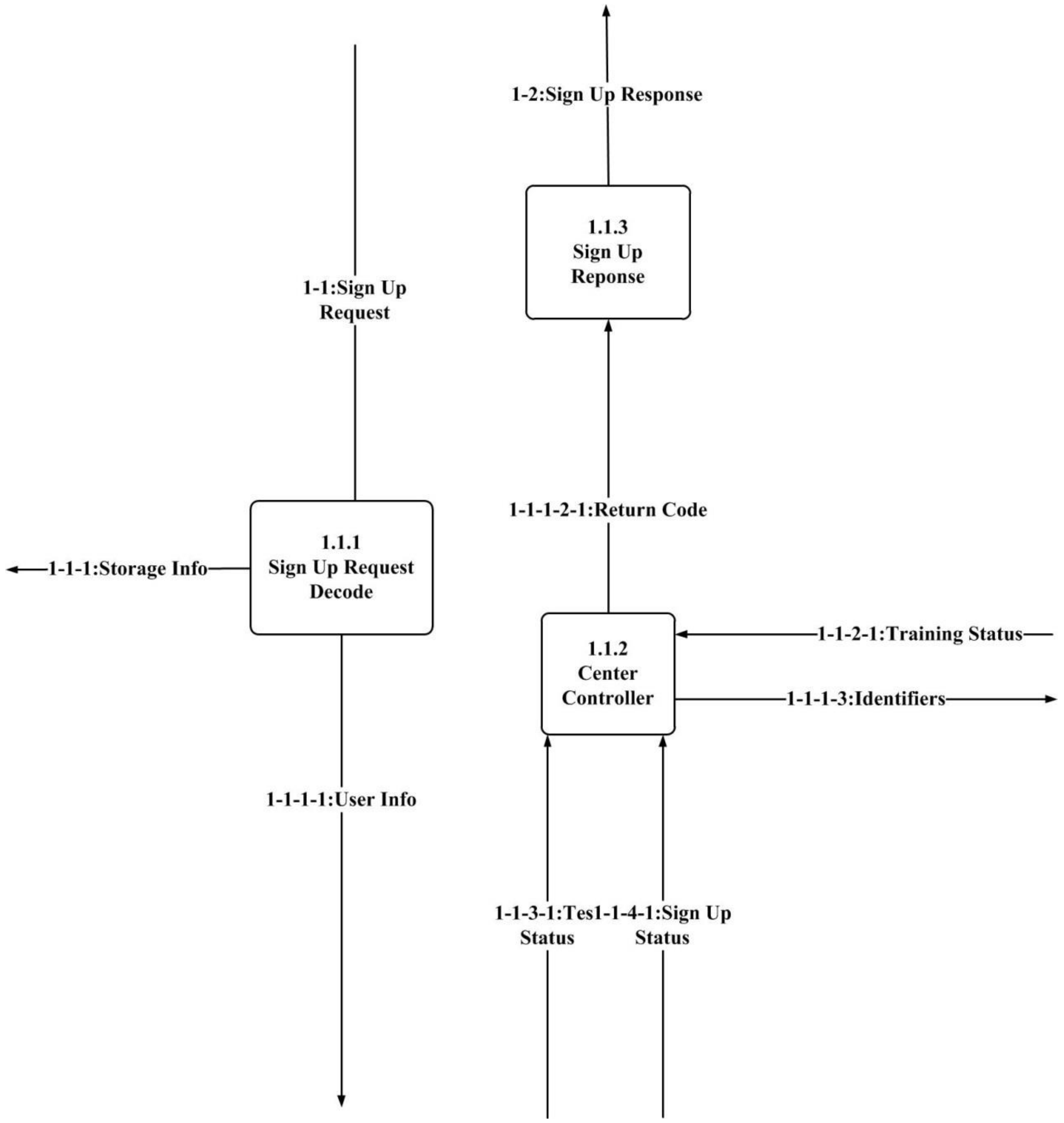


Figure 5: Sign Up Request Analyze Module data flow diagram

Figure 5 description

1. 1-1-1-2-1:Return Code Data Flow, which is the Center Controller send the code to the Sign UP Response and different code represent different meaning.

2.2.6 Sign In Request Analyze Module data flow diagram

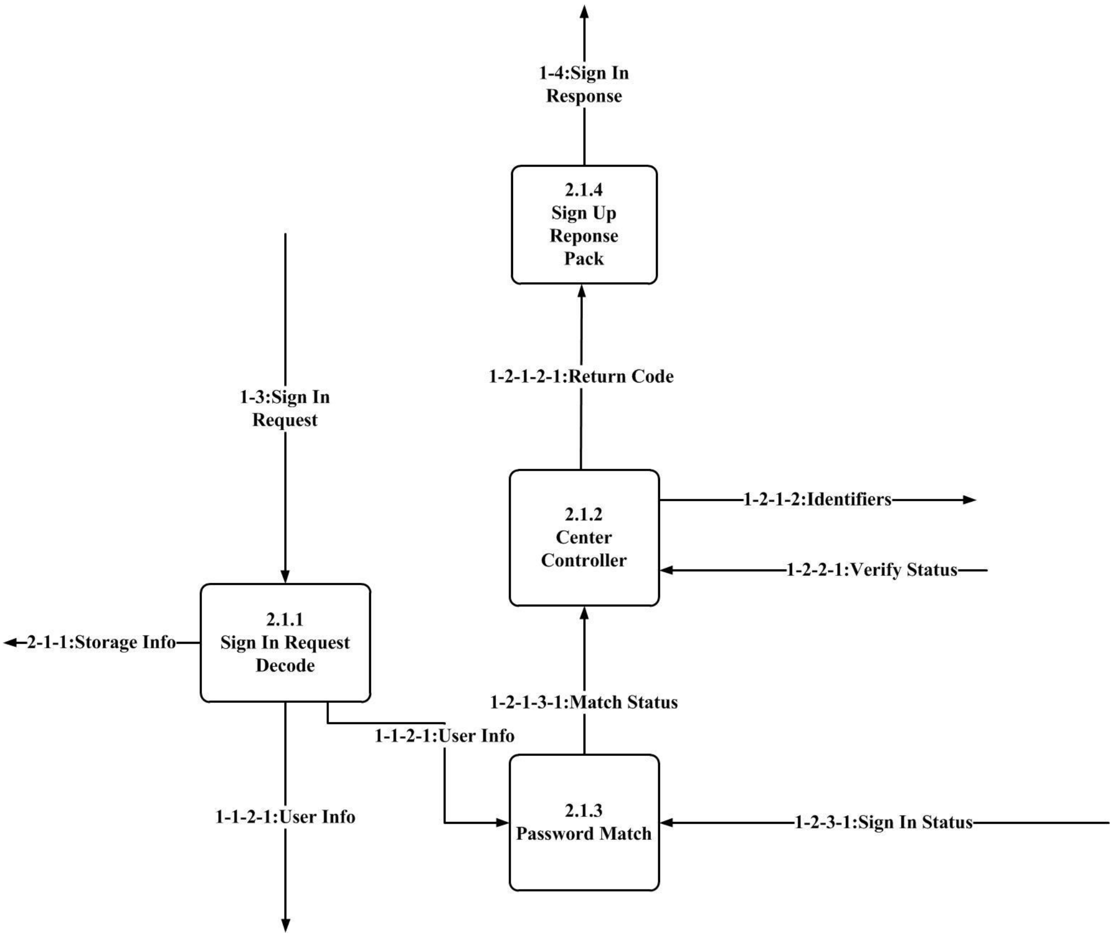


Figure 6: Sign Up Request Analyze Module data flow diagram

Figure 6 description

1. 1-2-1-2-1:Return Code Data Flow, which represent the status that the Center Controller send to the Sign Up Response Pack. Different code represent different status of Center Controller wield.

2. 1-2-1-3-1:Match Status Data Flow, which represent the status that if the Password Match run successful or failed.

2.3 Actor Descriptions

2.3.1 Client

Client acts as a requestor who take a photo of student attend the class and send the photo to Facilitator. Client can also request to train the server (add new student information, update their information, and remove the student information from the server).

2.3.2 Facilitator

Facilitator acts as a person who makes a decision for selecting the best result of student information from the servers and returns the result back to Client. Facilitator also receives the request from client to do the server and return its result back to client.

2.3.3 Server

Server is the recognizer; Server will receive the picture of students from Facilitator then identifies the students face and their position in the picture, mapping with their information and return the student information and position to Facilitator.

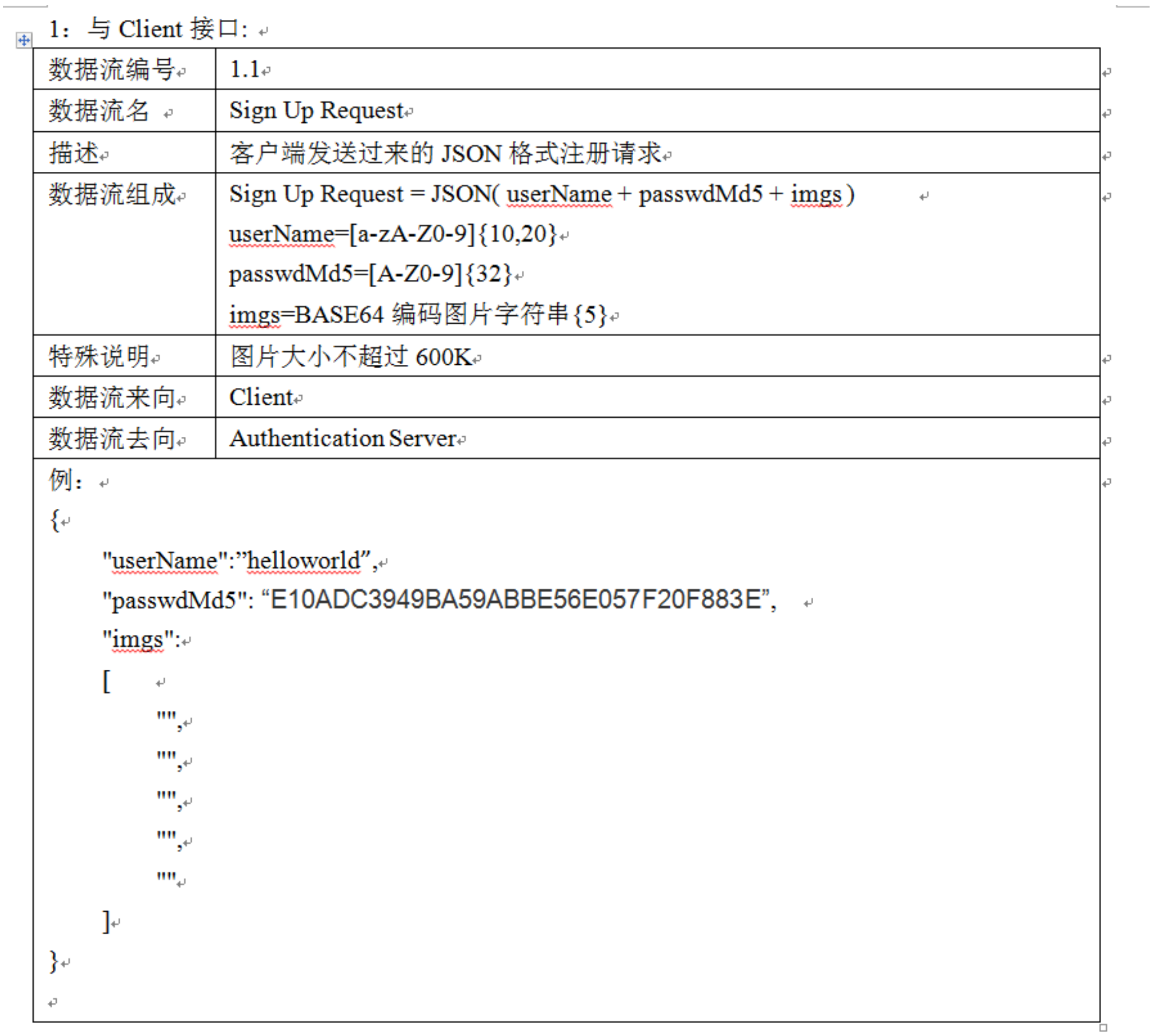
2.3.4 DataBase

Database acts as a storage for preservation the user info and the image code provide by face plus plus and so on. When the client sent a sign up request, the database need to provide server to preservation the info of user, and besides this, if the client sent a sign in request, the database need to provide user name to the facilitator, so that the facilitator can sent the image code and the user info to verify if the user is the customer.

3. Interface and Data Flow

3.1 Client Interface

3.1.1 Sign Up Request



1：与 Client 接口：

数据流编号	1.1
数据流名	Sign Up Request
描述	客户端发送过来的 JSON 格式注册请求

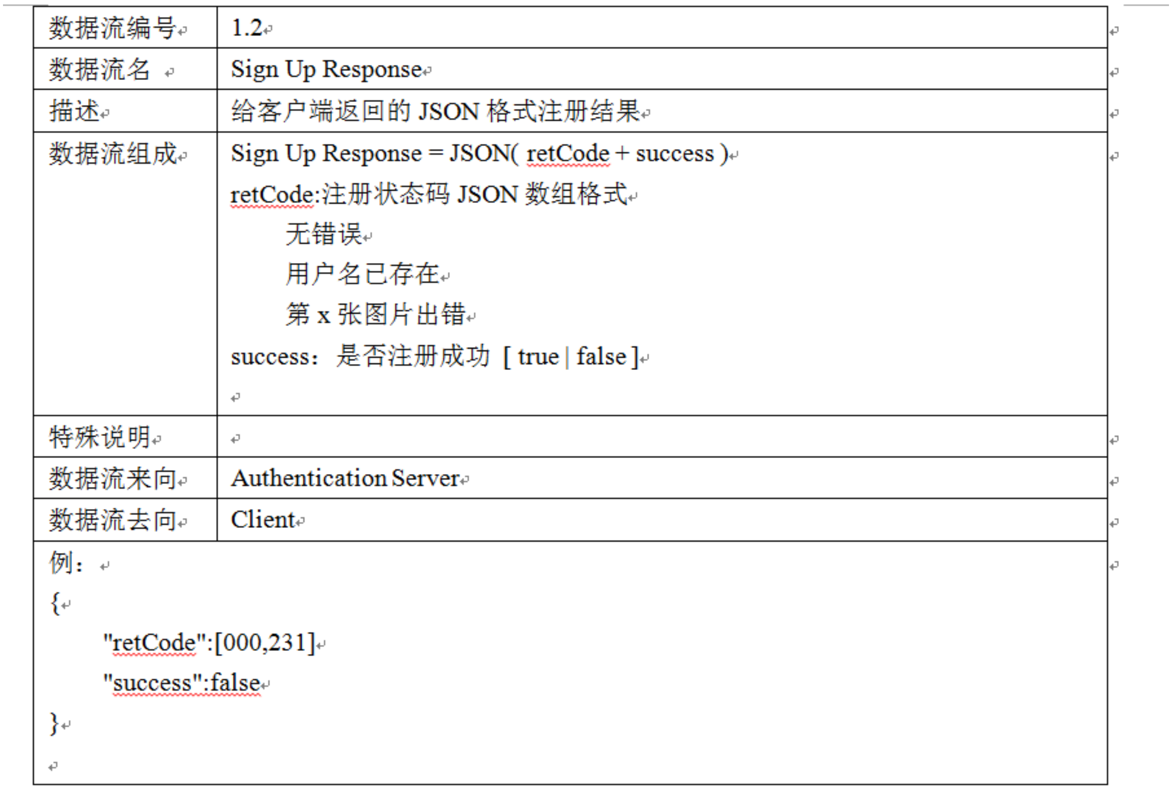
数据流组成	Sign Up Request = JSON(userName + passwdMd5 + imgs)
	userName=[a-zA-Z0-9]{10,20}
	passwdMd5=[A-Z0-9]{32}
	imgs=BASE64 编码图片字符串{5}

特殊说明	图片大小不超过 600K
数据流来向	Client
数据流去向	Authentication Server

例：

```
{
    "userName":”helloworld”,
    "passwdMd5": “E10ADC3949BA59ABBE56E057F20F883E”,
    "imgs":
    [
        "",
        "",
        "",
        "",
        ""
    ]
}
```

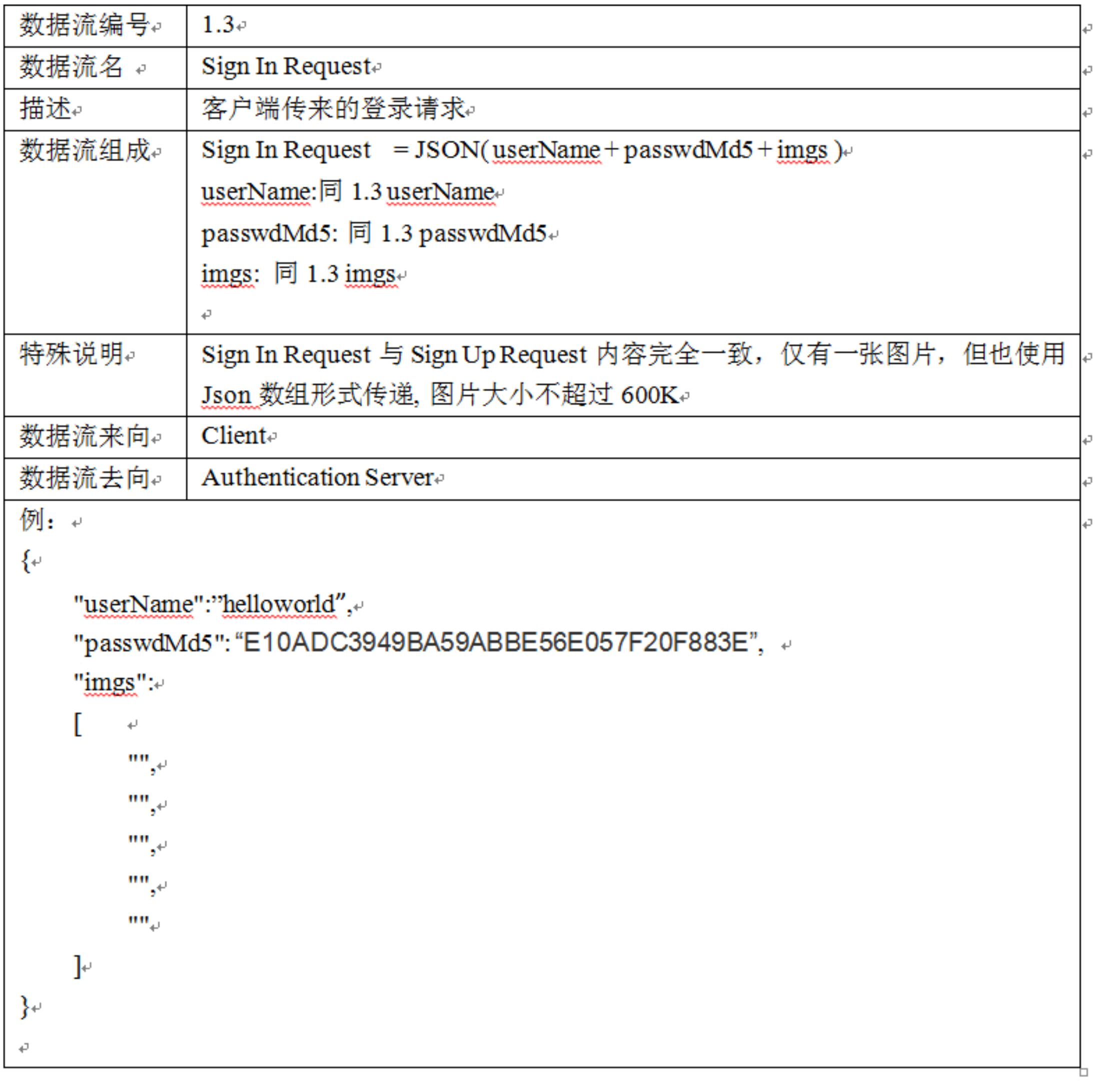
3.1.2 Sign Up Response



数据流编号	1.2
数据流名	Sign Up Response
描述	给客户端返回的 JSON 格式注册结果
数据流组成	Sign Up Response = JSON(retCode + success) retCode:注册状态码 JSON 数组格式 无错误 用户名已存在 第 x 张图片出错 success：是否注册成功 [true | false]
特殊说明	
数据流来向	Authentication Server
数据流去向	Client

例：
{
"retCode":[000,231]
"success":false
}	

3.1.3 Sign In Request



数据流编号	1.3
数据流名	Sign In Request
描述	客户端传来的登录请求
数据流组成	Sign In Request ＝JSON(userName＋passwdMd5＋imgs) userName:同 1.3 userName passwdMd5: 同 1.3 passwdMd5 imgs: 同 1.3 imgs
特殊说明	Sign In Request 与 Sign Up Request 内容完全一致，仅有一张图片，但也使用 Json 数组形式传递, 图片大小不超过 600K
数据流来向	Client
数据流去向	Authentication Server
例：	

```
{
      "userName":”helloworld”,
      "passwdMd5": “E10ADC3949BA59ABBE56E057F20F883E”,
      "imgs":
      [
            "",
            "",
            "",
            "",
            ""
      ]
}
```

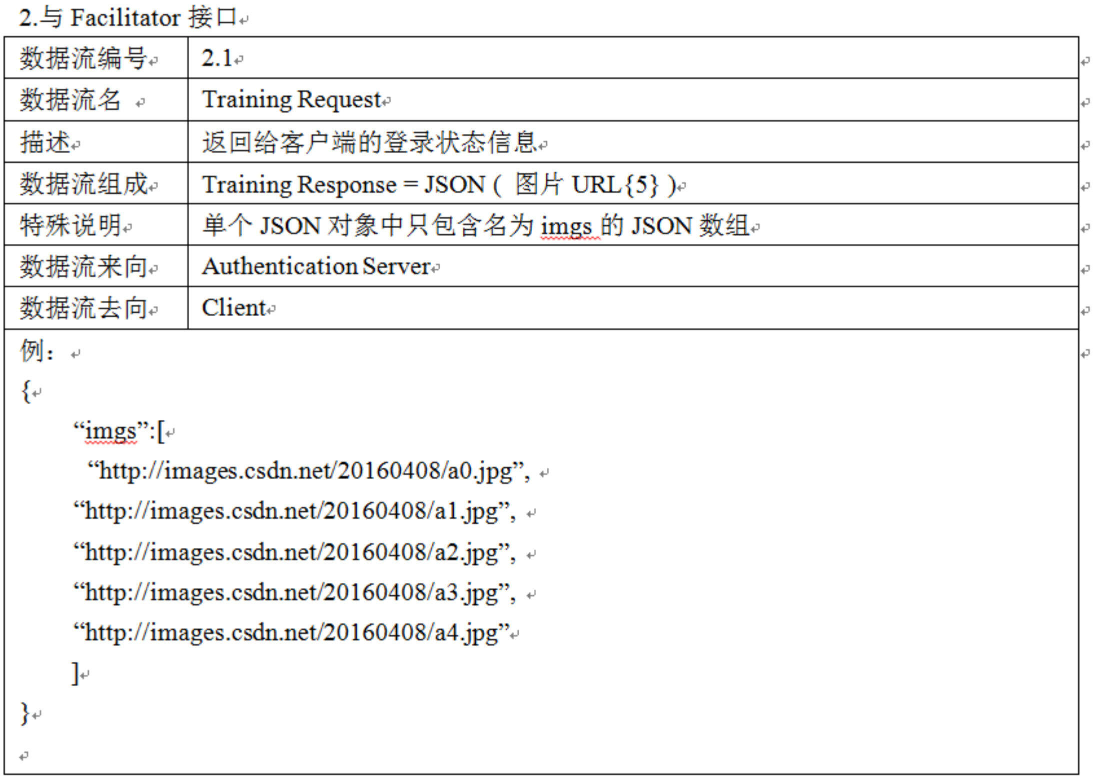
3.1.4 Sign In Response

数据流编号	1.4
数据流名	Sign In Response
描述	返回给客户端的登录状态信息
数据流组成	Sign In Response=JSON(retCode + success) retCode：登录状态码 JSON 数组格式 账号不存在 账号密码不匹配 图片不匹配
特殊说明	返回多个状态码
数据流来向	Authentication Server
数据流去向	Client

例：
{
"retCode":[000,231]
"success":false
}	

3.2 Facilitator Interface

3.2.1 Training Request



2.与 Facilitator 接口

数据流编号	2.1
数据流名	Training Request
描述	返回给客户端的登录状态信息
数据流组成	Training Response = JSON（ 图片 URL{5} ）
特殊说明	单个 JSON 对象中只包含名为 imgs 的 JSON 数组
数据流来向	Authentication Server
数据流去向	Client

例：

```
{
      “imgs”:[
        “http://images.csdn.net/20160408/a0.jpg”,
      “http://images.csdn.net/20160408/a1.jpg”,
      “http://images.csdn.net/20160408/a2.jpg”,
      “http://images.csdn.net/20160408/a3.jpg”,
      “http://images.csdn.net/20160408/a4.jpg”
      ]
}
```

3.2.2 Training Response

数据流编号	2.2
数据流名	Training Response
描述	Facilitator 返回的训练结果
数据流组成	Training Response = JSON (Identifiers + retCode) Identifier=[a-zA-Z0-9]{1,32} retCode：多个 3 位整数组成的返回状态码 第 x 张图片无效 URL 第 x 张图片中没有有效的面部 第 x 张图片损坏
特殊说明	retCode 是由多个整型组成的数组,Identifier 是由网站名与 identifier 组成的 JSON 对象
数据流来向	Facilitator
数据流去向	Authentication Server

例：
{
"retCode":[000,231]
"identifiers":
[
{
webName="face++",
identifier="aaaaaaaaa"
},
{
webName="gface++",
identifier="bbbbbbbb"
}
]
}	

3.2.3 Verify Request

数据流编号	2.3
数据流名	Verify Request
描述	向 Facilitator 发送的认证照片与用户是否为同一个的请求
数据流组成	Verify Request =JSON(URL + Identifiers) URL：合法 URL 格式 Identifiers：多个待认证用户的特征码 =[a-zA-Z0-9]{1,32}
特殊说明	URL 为只有一个字符串元素的数组
数据流来向	Authentication Server
数据流去向	Facilitator

例：

```
{
        "url":[“http://images.csdn.net/20160408/a0.jpg”],
        "identifiers":
        [
            {
                webName=”face++”,
                identifier=”aaaaaaaaa”
            },

            {
                webName=”gface++”,
                identifier=”bbbbbbbb”
            }
        ]

}
```

3.2.4 Verify Response

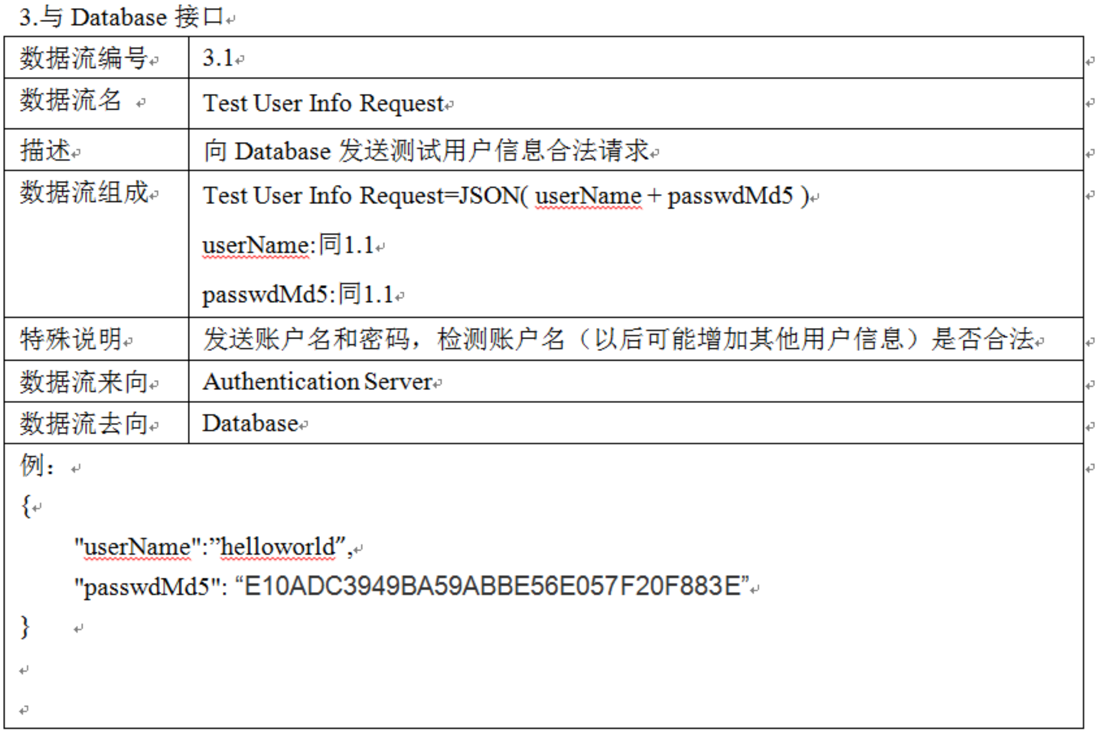
数据流编号	2.3
数据流名	Verify Response
描述	Facilitator 返回的认证照片与用户是否为同一个的结果
数据流组成	Verify Response=JSON(retCode + success) retCode：多个 3 位整数组成的 JSON 数组 无错误 URL 无效 认证成功 认证失败 success: [true | false]
特殊说明	
数据流来向	Facilitator
数据流去向	Authentication Server

例：

```
{
      "retCode":[000,003],
      “success”:true
}
```

3.3 DataBase Interface

3.3.1 Test User Info Request



3.与 Database 接口

数据流编号	3.1
数据流名	Test User Info Request
描述	向 Database 发送测试用户信息合法请求
数据流组成	Test User Info Request=JSON(userName + passwdMd5) userName:同1.1 passwdMd5:同1.1
特殊说明	发送账户名和密码，检测账户名（以后可能增加其他用户信息）是否合法
数据流来向	Authentication Server
数据流去向	Database

例：

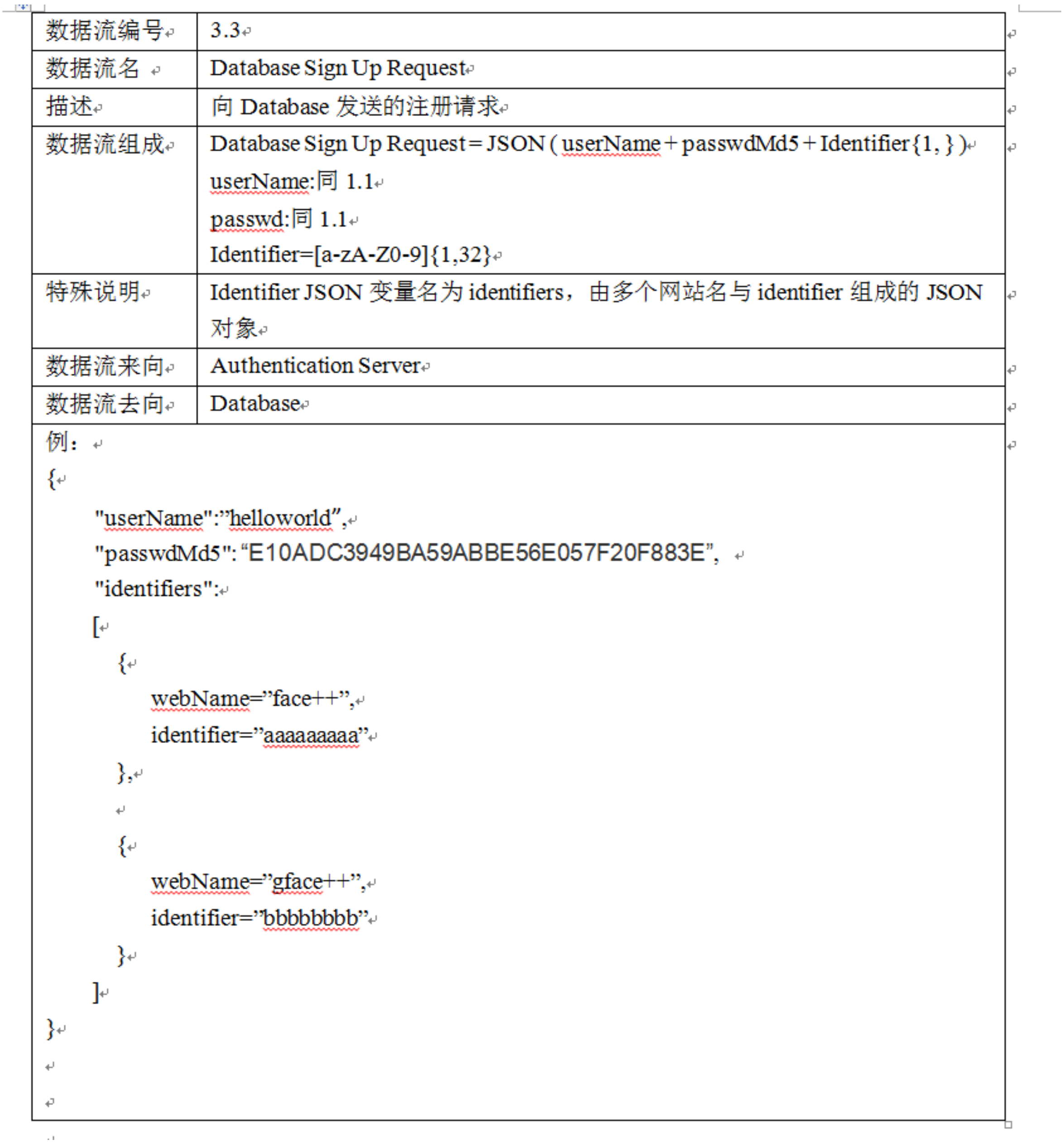
```
{
     "userName":”helloworld”,
     "passwdMd5": “E10ADC3949BA59ABBE56E057F20F883E”
}
```

3.3.2 Test User Info Response

数据流编号	3.2
数据流名	Test User Info Response
描述	Database 返回测试用户信息合法结果
数据流组成	Test User Info Request = JSON(retCode + success) retCode: 用户信息合法 用户名已存在 其他错误 success：是否合法 [true | false]
特殊说明	
数据流来向	Database
数据流去向	Authentication Server

例：
{
"retCode":[001,231]
"success":false
}	

3.3.3 DataBase Sign Up Request



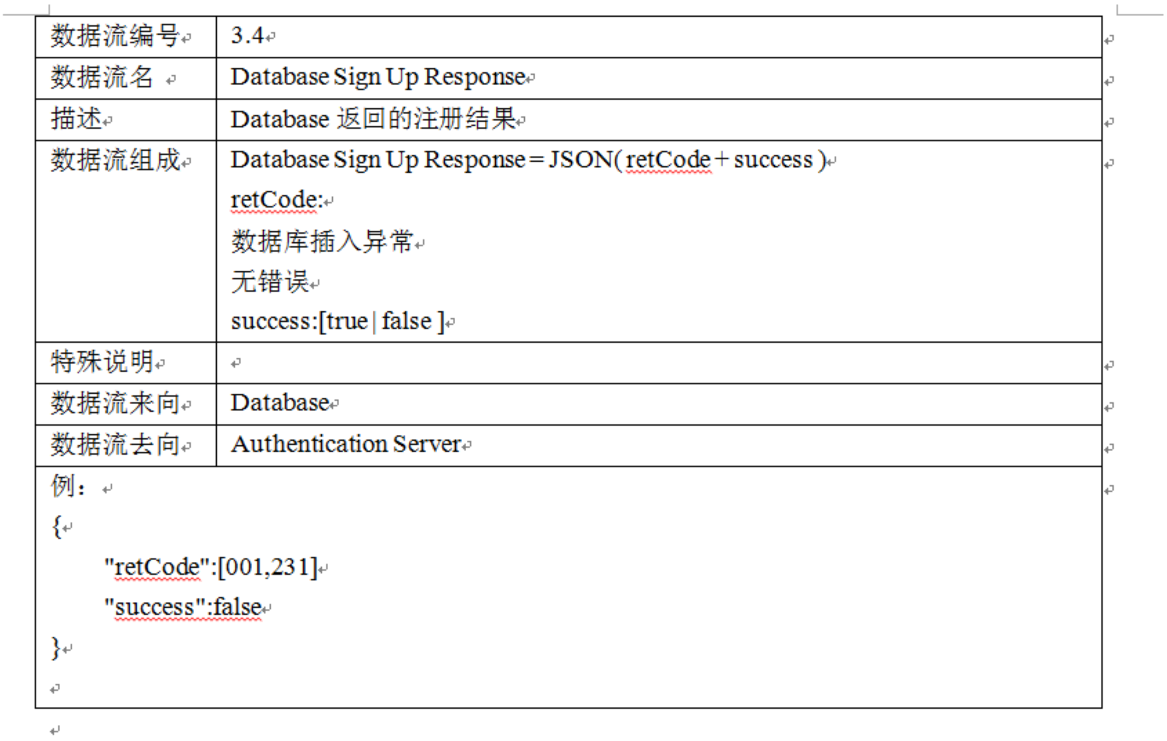
数据流编号	3.3
数据流名	Database Sign Up Request
描述	向 Database 发送的注册请求
数据流组成	Database Sign Up Request＝JSON（userName＋passwdMd5＋Identifier{1,}） userName:同 1.1 passwd:同 1.1 Identifier=[a-zA-Z0-9]{1,32}
特殊说明	Identifier JSON 变量名为 identifiers，由多个网站名与 identifier 组成的 JSON 对象
数据流来向	Authentication Server
数据流去向	Database

例：

```
{
    "userName":”helloworld”,
    "passwdMd5": “E10ADC3949BA59ABBE56E057F20F883E”,
    "identifiers":
    [
      {
         webName=”face++”,
         identifier=”aaaaaaaaa”
      },

      {
         webName=”gface++”,
         identifier=”bbbbbbbb”
      }
    ]
}
```

3.3.4 DataBase Sign Up Response



数据流编号	3.4
数据流名	Database Sign Up Response
描述	Database 返回的注册结果
数据流组成	Database Sign Up Response＝JSON(retCode＋success) retCode: 数据库插入异常 无错误 success:[true | false]
特殊说明	
数据流来向	Database
数据流去向	Authentication Server

例：
{
"retCode":[001,231]
"success":false
}	

3.3.5 DataBase Sign In Request

数据流编号	3.5
数据流名	Database Sign In Request
描述	向 Database 发送的登录请求
数据流组成	Database Sign In Request = JSON (userName) userName:同 1.1
特殊说明	
数据流来向	Authentication Server
数据流去向	Database

例：
{
“userName”:”helloworld”
}	

3.3.6 DataBase Sign In Response

数据流编号	3.6
数据流名	Database Sign In Response
描述	Database 返回的用户信息
数据流组成	Database Sign In Response＝JSON (passwdMd5＋Identifiers{1, }＋retCode) passwdMd5:同 1.1 Identifiers： 同 2.2 retCode: 用户名不存在 其他错误
特殊说明	
数据流来向	Database
数据流去向	Authentication Server

例：

```
{
    “passwdMd5”:” E10ADC3949BA59ABBE56E057F20F883E”,
    "identifiers":
    [
      {
         webName=”face++”,
         identifier=”aaaaaaaa”
      },

      {
         webName=”gface++”,
         identifier=”bbbbbbbb”
      }
    ]
    retCode:[000,001,002]
}
```