

# 1. Userinfo System Design

## 1. Requirement Background

- We want to provide a platform where user can view and edit their profiles.

## 2. Design Target

### 2.1 Core functions

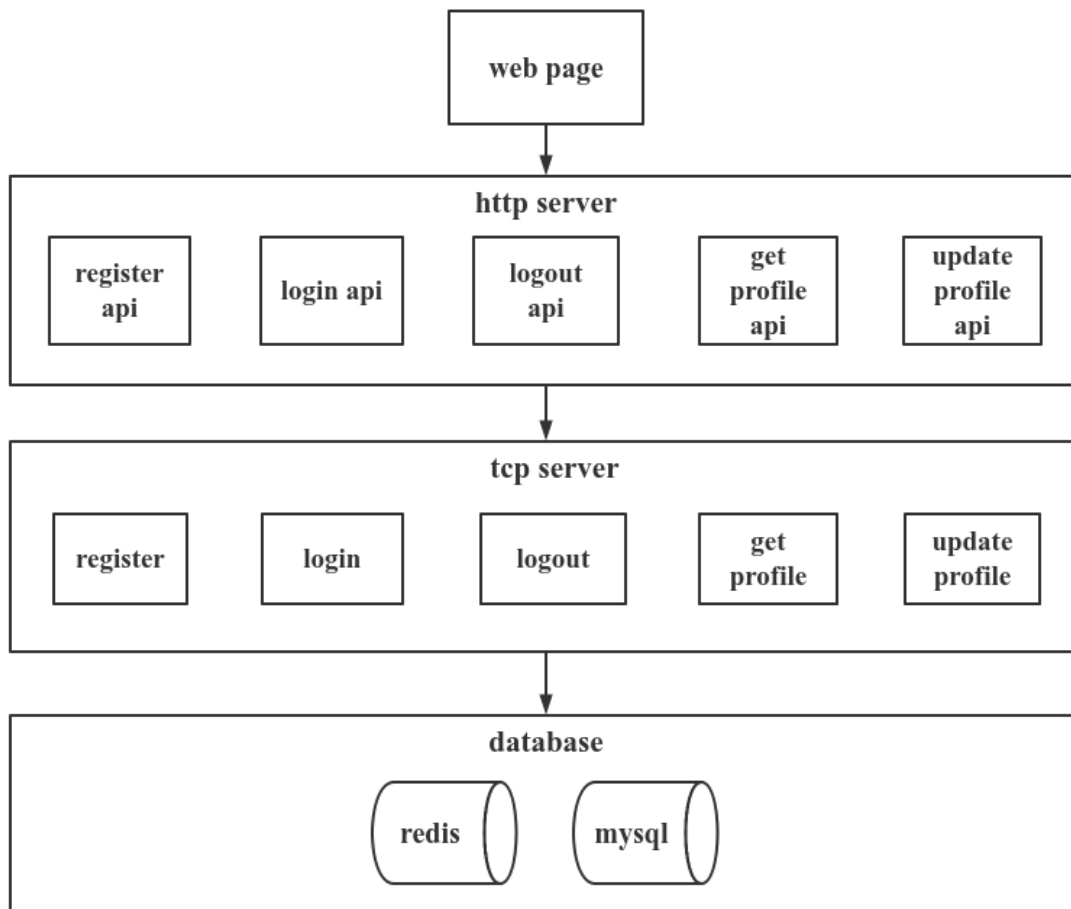
- User account: Users can register an account, login and logout.
- User profile: Users can get their profile after login, and is able to update the profile. Note that the login status should be kept for a certain time so that users can visit their profiles without entering id and password.

### 2.2 Performance

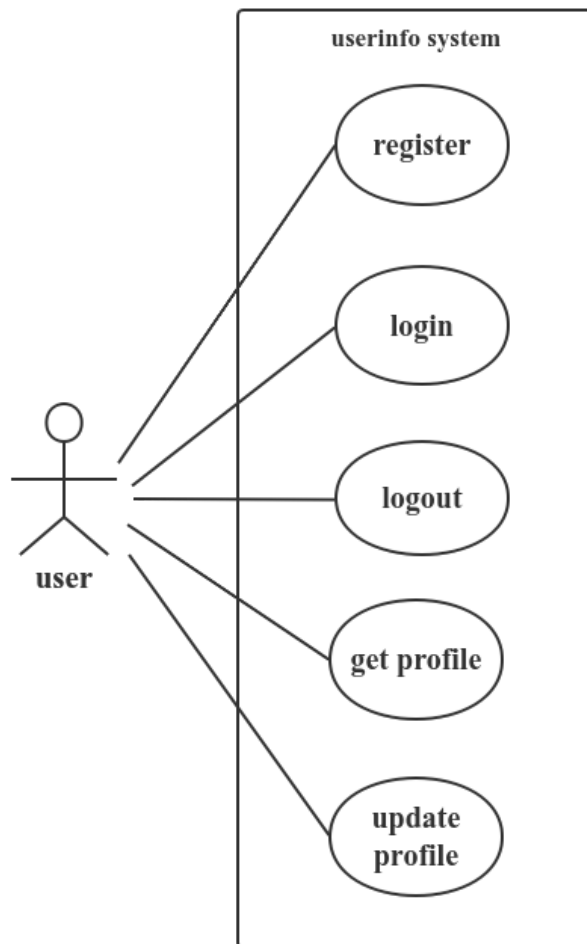
- qps and TP999

## 3. System Architecture

### 3.1 Architecture diagram

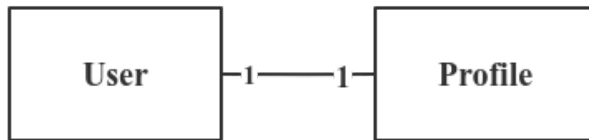


## 3.2 Use case diagram



## 4. Data Model Design

### 4.1 Data model diagram



## 4.2 Database table design

### profile\_tab

```

CREATE TABLE `profile_tab`
(
  `id` bigint unsigned NOT NULL AUTO_INCREMENT,
  `user_id` bigint unsigned NOT NULL,
  `username` varchar(255) NOT NULL DEFAULT '',
  `birthday` DATE,
  `email` varchar(255) NOT NULL DEFAULT '',
  `avatar_url` varchar(255) NOT NULL DEFAULT '',
  `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
  CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `idx_user_id` (`user_id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
  
```

### user\_tab

```

CREATE TABLE `user_tab`
(
  `id` bigint unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL DEFAULT '',
  `password` varchar(255) NOT NULL DEFAULT '' COMMENT 'encrypted by md5',
  `email` varchar(255) NOT NULL DEFAULT '',
  `status` tinyint(3) unsigned NOT NULL DEFAULT 0 COMMENT '0-available, 1-suspended, 2-
  deleted',
  `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  
```

<b>user_tab</b>
<pre> `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, PRIMARY KEY (`id`), UNIQUE KEY `email` (`email`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci; </pre>

## 4.3 Cache design

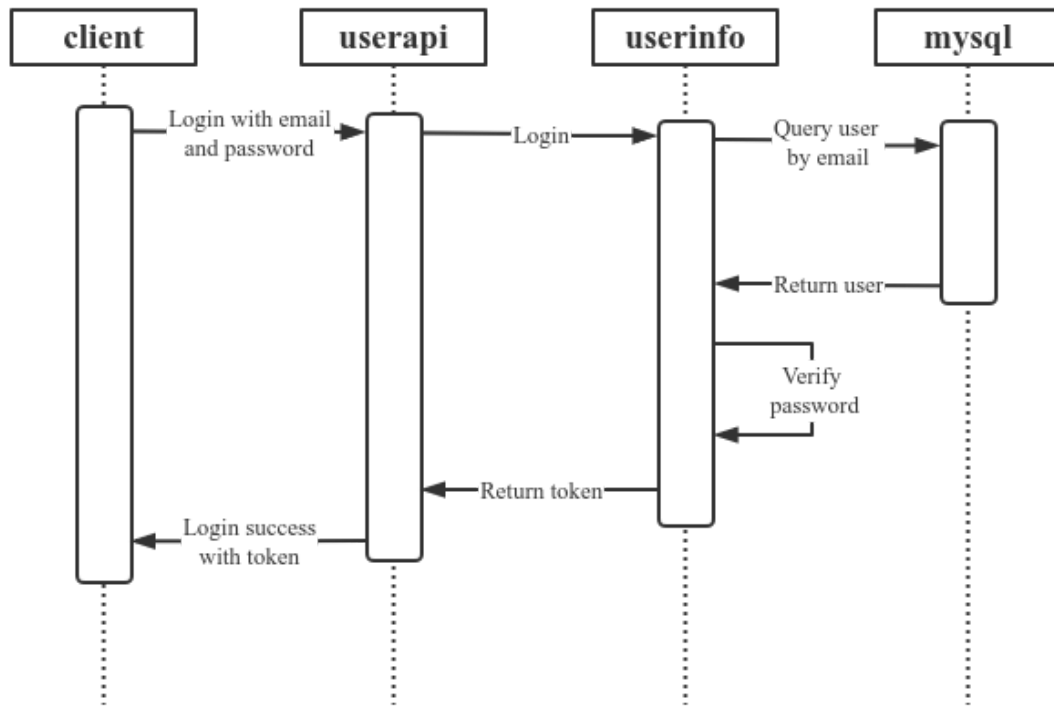
Key	Value	Expire time	Description
userinfo:get_profile: {user_id}	{"id": 1,"user_id": 1,.....}	60 ± rand(30) s	Cache profile as json string type.

# 5. Description of Core Processes

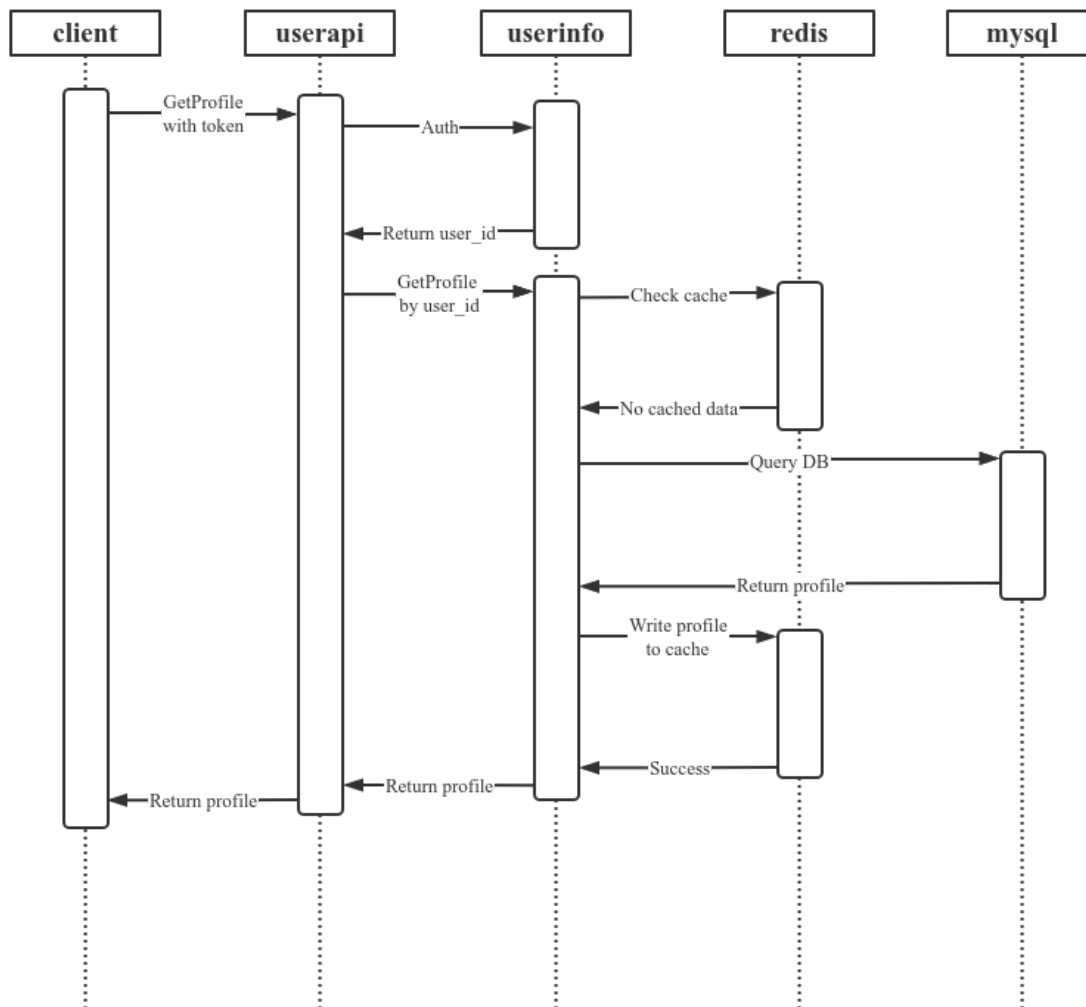
## 5.1 Flowchart

## 5.2 Sequence diagram

### 5.2.1 Login



### 5.2.2 Get profile



## 5.3 API description


### 5.3.1 Register

#### Request

POST api/account/register

#### Fields

Field	Type	Desc	Mandatory	Remark
email	string	email for	✓	

Field	Type	Desc	Mandatory	Remark
		registration		
password	string	password for. registration, processed by MD5		

## Response

```
{
  "code": 0,
  "msg": success,
  "data": {}
}
```

## Error code



Code	Desc	Remark
200001	Register failed, email has been registered.	
200002	Register failed, internal server error.	
200003	Register failed, bad request.	

## 5.3.2 Login

### Request

POST api/account/login

### Fields

Field	Type	Desc	Mandatory	Remark
email	string	email for registration		
password	string	password for. registration, processed by MD5		

## Response

```
{
  "code": 0,
```



```
"msg": success,  
"data": {}  
}
```

### Error code

Code	Desc	Remark
200004	Login failed, no such user.	
200005	Login failed, internal server error.	
200006	Login failed, password mismatch.	
200007	Login failed, password mismatch.	

### 5.3.3 Logout

#### Request

```
POST api/account/logout
```

#### Response

```
{  
  "code": 0,  
  "msg": success,  
  "data": {}  
}
```

### 5.3.4 Get profile

#### Request

```
GET api/user/profile
```

#### Response

```
{  
  "code": 0,  
  "msg": success,  
  "data": {  
    "user_id": 1,  
    "user_name": "lgk",  
    "email": "lgk@gmail.com",  
    "birthday": "1995-03-01"  
  }  
}
```

```
}  
}
```

## Fields

Field	Type	Desc	Mandatory	Remark
user_id	int	user id	✓	
user_name	string	user name		
email	string	registered email	✓	
birthday	string	birthday		

## Error code

Code	Desc	Remark
100001	Get profile failed.	

## 5.3.5 Create profile

### Request

POST api/user/profile

## Fields

Field	Type	Desc	Mandatory	Remark
user_name	string	user name		
birthday	string	birthday		

## Response

```
{  
  "code": 0,  
  "msg": success,  
  "data": {}  
}
```

## Error code

Code	Desc	Remark
100002	Create profile failed.	

### 5.3.6 Update profile

#### Request

PUT api/user/profile

#### Fields

Field	Type	Desc	Mandatory	Remark
user_name	string	user name		
birthday	string	birthday		

#### Response

```
{
  "code": 0,
  "msg": success,
  "data": {}
}
```

#### Error code

Code	Desc	Remark
100003	Update profile failed.	