

# Design Document for Smart House project

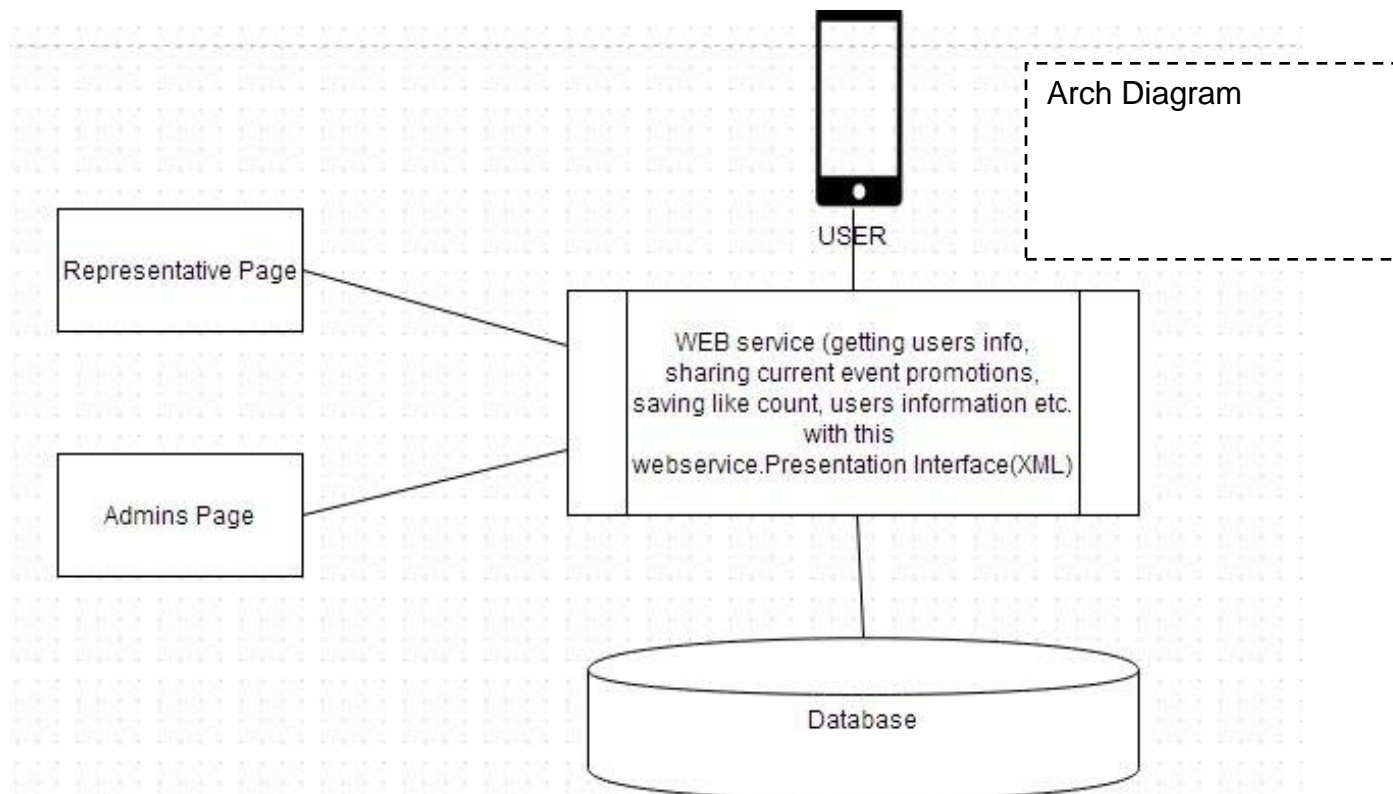
EN\_3\_B04

Alisher Kerimbek

We didn't have any meeting hours

## 3 Decomposition Description

### 3.1 Module Decomposition



The project consists of 3 main layers: user part, WEB server part and database. Also there are additional layer such as «Social network services»(VK, FACEBOOK) which are aimed to provide extra functionality to our application. We decided to use MVC pattern for server part implementation that will interact with all others parts of our project architecture.

#### 3.1.1 Devices description

We will be able to provide our site services for any device that has connection to internet. In addition there will be mobile application for users of smartphones. We will work on doing mobile application, which will maintain Android platform.

### **3.1.2 WEB server description**

Like typical MVC app it will consists from models, views and controllers. The main manager is controller; it will receive requests from users and interact with models and views to give some response and then saves likes , users info, shares etc. Most of our project logic will be in controllers. Models will return needed data from database that controller is asking. Views will be composing of html, css, java-script to give GUI with which user can interact.

### **3.1.3 Database description**

Database will store all tables that our project need, and also API with which mobile application will interact with our server. We didn't have any logic in this part, because model part of WEB server will do all work (CRUD).

### **3.1.4 Social network services description**

It is the last layer of our system., it is external library (libraries), which will give opportunity for our users to not register to our site, but just sign in with their social network account. We will implement it by using their API and our logic of registration.

## **3.2 Concurrent Process**

### **3.2.1 Mobile application process**

**Mobile application process will have functions to see the list of events,u can share an event, like , search those, sign in or sign up to our app, manage his account and events. There is one main thread – session thread. It will give information to application is user registered or not, and user info.**

### **3.2.2 WEB server process**

**This process has virtually 3 threads, and responsible for them are controllers, models and views. Controllers will have main one which will coordinate work of others. For any users action it should give some response. So, it will be real time thread that will call other 2 threads. Second thread by appeal of first one will take requested data from database and send to first thread. It is the main purpose of this thread. The last thread will give response to user by information provided by first in web page format.**

### **3.2.3 Database process**

**It has one thread that waits till 2nd thread from WEB server process will send request for doing actions on data. And according to type of action, this process will create, delete or update some information in database.**

### **3.2.4 Social network services process**

**This process also has one thread, which firstly checks through which social network user wants to sign in, and then make request through selected service API to take info about user if he exists.**

## **5.1 MODULE INTERFACE**

### **5.1.1 INTERFACE BETWEEN MAIN PAGE AND DATABASE CONNECTION LIBRARY**

#### **1) public function createUser(String url)**

**is used to save new user in database, it makes "url request" and saves new user in WebService's database.**

**Parameters:**

**url = it gives to url for making request, then in webservice accepts response(user profile) by this url. After this user will be created.**

#### **2)public function newsOfUser(intuser\_id, int count, intcurrTime)**

**is used to show all news in main page and all these data takes from WebServices's database.**

**Parameters:**

**user\_id = registered user's id**

**count = how many news will be shown in main page.(in our case 15)**

**currTime = current time will help us, when was last request to webservice.**

**4) public function newsById(intuser\_id, intnews\_id)**

**when we click one news on the "Main page" then, it will show these news fully on another page.**

**Parameters:**

**user\_id = the unique id of user**

**news\_id = the news id which is pressed on "Main Page".**

**5) public function likeNews(intuser\_id, news\_id**

**is used to like the news,**

**Parameters:**

**id = the unique id of user**

**news\_id = it is used to know which news liked user.**

**6)public function allCategories(intuser\_id)**

**is used to show all categories from database,**

**Parameters:**

**user\_id = the unique id of each user,**

**7) public function categoryById(intuser\_id, intcategory\_id)**

**Is used to show each category on one page also it shows all news which are turned in this category.**

**Parameters:**

**user\_id = the unique id of user**

**category\_id = it helps to get all category information from Database.**

**8) public function followcategory(intuser\_id, intcategory\_id)**

**is helps to user follow each category.**

**Parameters:**

**user\_id = the unique id of user,**

**category\_id = it is used to know which category followed user.**

**9) public function unfollowcategory(intuser\_id, intcategory\_id)**

**is helps to user unfollow each category.**

**Parameters:**

**user\_id = the unique id of user,**

**category\_id = it is used to know which category unfollowed user.**

**10) public function followpartner(intuser\_id, intpartner\_id)**

**is helps to user follow each promotion.**

**Parameters:**

**user\_id = the unique id of user,**

**category\_id = it is used to know which promotion followed user.**

**11) public function unfollowpartner(intuser\_id, intpartner\_id)**

**Parameters:**

**user\_id = the unique id of user,**

**category\_id = it is used to know which promotion unfollowed user.**

**12) public function partnersofuser(intuser\_id)**

**Is used to show which promotions followed user.**

#### **5.1.1.2 Interface between code and Database connection library**

```
1) function __construct(){  
    parent::__construct();  
    $this->load->database();  
    $this->load->model('mmob');  
}
```

#### **5.1.2 Function between main page and vk-api and facebook-api functions**

**1) functiongetProfiles()**

**Is used to get all information about user.**

**Parameters: facebook or vkontakte ID, and TOKEN of application**

**Return type: first\_name, last\_name, sex, bdate, photo\_medium.**

## **5.2 PROCESS INTERFACE**

### **5.2.1 Main Page View process**

**Description:** This process shows main page's graphical interface. On the first page will be, gallery of our application.

**Also will be 2 buttons (First: Authorization via Facebook, and second: authorization via Vkontakte.**

**5.2.1.1 Interface View Process is created when the application calls onCreate() method on the activity.**

**5.2.1.2 Closed when applications "Exit" button is pressed**

### **5.2.2 Database Listener process**

**5.2.2.1 This process is included before opening main processes and it require user and password of database.**

**Database**

**5.2.2.2 Database listener process - listened query on each page. And it returns required data.**

### **5.2.3 Controller process**

**5.2.3.1 This thread is created after Database Listener process**

**5.2.3.2 Mainly this thread interacts between Main thread and Database listener tread**

**5.2.3.3 Thread is terminated when one of other two threads terminates**

