CSP203 - Software Systems Lab

Project's Test Document
Events App

(Version: 2.0)

Naman Goyal (2015CSB1021)
Sarthak Gupta (2015CSB1029)
Vishal Singh (2015CSB1040)
Nittin Singh (2015CSB1067)

April 24, 2017

Contents

1 Introduction

To ease students' stay at IIT Ropar, and provide one click access to all relevant event updates at the institute.

2 PHP + mySQL

The mySQL databale has 3 table consists

1. bookmarks - Stores all the bookmarks of the all the users
2. user - Store user's profile
3. events - Store all event related details with organizer id
4. usertype - Store entry for different types of users

Now their are php scripts to access and update these data

2.1 ListAll.php

Able to list all the events in the databse.
Test: Running this php listed all the events in the Events table of the Dateaase in JSON format with "success".

2.2 Login.php

Able to authenticate login for a user-id.
Test: Many users were created on the database. PHP script was tested by entering the user credentials using the POST method. Script responded "success" or "failure" after veririfying with the database.

2.3 Register.php

Able to register a new user having unique email iitrpr email.
Test: Users were registered using the POST method by entering the correct fields. The script responded with "success" or "failure" depending on the correct format with that of database.

2.4 AddEvent.php

Able to create new row for an event in sql table for a organizer-id.
Test: Events were added using the POST method by entering the events details and was checked in the databse. Also "success" or "failure" was responded by the script depending on the correct format.

2.5 category.php

Tells all the types of events allowed in a database.
Test: Running this script responded with the event categories and their IDs.

2.6 changePassword.php

Able to update the password of the user.
Test: Tested by giving the email, oldpassword, newpassword by POST method. It responded with "success" or "failure".

2.7 connect.php

Helps in connecting with the databse
Test: It responded with "Success" or "failure" depending on the credentials.

2.8 detailEvent.php

Gives the details of the events
Test: Tested by giving the eventid by POST method. It responded with the events details and "Success" or "failure".

2.9 getInfo.php

Tells about the topics to which the user is subscribed.
Test: Tested by giving the userid by POST method. It responded with the topics id and ”Success” or ”failure”.

2.10 listBookmark.php

Gives all the bookmarks of the user.
Test: Tested by giving the userid by POST method. It responded with all the bookmarked events in the JSON format and ”Success” or ”failure”.

2.11 toggleBookmark.php

It is used to toggle the bookmarked/unbookmarked events of the user.
Test: Tested by giving the userid and event by POST method. It responded with ”Success” or ”failure”.

2.12 usertypeRegister.php

It gives the all the types of the user that can be registered in the database.
Test: Running this script responded with the user categories and their IDs.

3 System Test

When whole module is integrated a user should be able to perform all actions synchronously. The cases test interactions of all the module where php scripts are called by volley in android which is then processed by the app.
Following are the test cases that are used in working of the app -

- New user was registered by entering the details. The app is working fine showing the message ’user registered’ and a new user is registered in the ’user’ table in the database.
- After login the events are listed in different categories (eg. All, Cultural etc). The data is loaded by makimg a volley request to the server which gives the events details. Also the app got the token from the FCM.
- Events can be bookmarked, in which case the app sends the bookmark status through volley library and the ’toggleBookmark.php’ updated the bookmark table in the database.
- The user organised an event by entering the details and the app notified with the message ’event added’. Further, the app makes a volley request and the ’addEvent.php’ adds the event in the event table in the database.
- After adding the event the app called the ’Firebase Cloud Messaging’ service that sends the push notifications to all the registered apps .
- Password by changed where the app asks for the new password and is sent to the server through volley request. The ’changePassword.php’ interacts with the server and update the user table in the database.
- Rest all functions doesn’t require the whole system (app side codding) and are worked fined when tested.