## User Access Workflow and Design

Updated : 3-26-2013 ola lawal

Create user workflow

**general user login steps**

Two options for logging in

1) new users

2) existing user

1) new user

Steps:

1)gather registration data based on the order of the screens :

-**Username** - add a tooltip explaining that this is what the user uses to log on to the app from the website or the app if they log out of the app.

-**Email Address** – pre-populate this value with the users ApplieId email address if accessible via API, in case of android attempt to use GoogleID which is also their google app store id . If that is not available leave blank and force user to enter it . Add tootip statting that this is needed for updates such as email and account notifications.

-**ScreenName**- add a tooltip explaining that since the username is a private value , screen name is the name other users will know you as. i.e a nickname.

-**Birthdate** – add tootip statting must be 18 and older.

-Gender – gender list is populated from this service call: <http://localhost/Shell.MVC2.Web.Common/LookupService.svc/Rest/getgenderlist>

json response is a generic list format below

"description":"String content",

"id":2147483647

}]

Where description is the text to display and the ID is the value stored back in the database and the value.

Help page for this service call:

<http://localhost/Shell.MVC2.Web.Common/lookupservice.svc/rest/help/operations/getgenderlist>

-**Ethnicity** – is also populated by the lookup service :

<http://localhost/Shell.MVC2.Web.Common/LookupService.svc/Rest/getethnicitylist>

json response is a generic list format below

"description":"String content",

"id":2147483647

}]

Where description is the text to display and the ID is the value stored back in the database and the value.

Help page for this service call:

<http://localhost/Shell.MVC2.Web.Common/lookupservice.svc/rest/help/operations/getethnicitylist>

1) validate email

[3/24/2013 10:55:25 PM] himanshi vishnoi: 2) validate screen name

[3/24/2013 10:55:27 PM] himanshi vishnoi: 3) create user

[3/24/2013 10:55:30 PM] himanshi vishnoi: 4) upload pic

Two was of creating a new user account

**1) Non Facebook or openID user**

-prompt use that this app requires access to your location

-gather user data and fill out the RegisterModel/creatuser:

-data validation :

email address

-----------

http://localhost/Shell.MVC2.Web.MembersService/MembersService.svc/Rest/checkifemailalreadyexists/{EMAILADDRESS}

-returns a true if it exists i,e email address is invalid

screenanem

----------

must be at least six chars , can contain special chars

http://localhost/Shell.MVC2.Web.MembersService/MembersService.svc/Rest/checkifscreennamealreadyexists/{SCREENAME}

-reutrns true if screen name already exists -user must pick a new one

username

----------

must be at least six chars , can contain special chars within reason

http://localhost/Shell.MVC2.Web.MembersService/MembersService.svc/Rest/checkifprofileidalreadyexists/{PROFILEID}

- returnes true if usenrame already exists- user must pick a new one

here is the JSON data needed

{

"birthdate":"\/Date(928167600000-0500)\/",

"city":"String content",

"country":"String content",

"email":"String content",

"gender":"String content",

"isApproved":true,

"lattitude":1.26743233E+15,

"longitude":1.26743233E+15,

"openidIdentifer":"String content",

"openidProvidername":"String content",

"password":"String content",

"providerUserKey":{},

"screenname":"String content",

"securityAnswer":"String content",

"securityQuestion":"String content",

"stateprovince":"String content",

"status":0,

"username":"String content",

"zippostalcode":"String content"

}

-location data city,country,longitude,lattitude stateprovince - can be generated from phone

or use the geoservice to get that data.

-if user opts to turn off gelocation on phone we still have to find a way to veryfy that the user enteres in

location data that matches what is on thier phone/tablet , so maybee not let them turn it off

-veryfy the geodata if manually entered in using the geoservice

http://localhost/Shell.MVC2.Web.GeoService/GeoService.svc/Rest/verifyorupdateregistrationgeodata

also if you do not have lat and long info you can get that data using this method. Will provide sample calls in near future

once everythin is verifed call

http://localhost/Shell.MVC2.Web.AuthenticationService/MembershipService.svc/Rest/CreateUser

with the JSON body above and

if status is 0 on the return profile was created.

Option B. Facebook or openID user

-pretty much same as above , difference is you call

- get profileinformation after creation

------------------------------------

http://localhost/Shell.MVC2.Web.MembersService/MembersService.svc/Rest/getprofileidbyusername/{USERNAME}

or

http://localhost/Shell.MVC2.Web.MembersService/MembersService.svc/Rest/getprofileidbyscreenname/{SCREENAME}

then use the profileID information to populate photos below...

photos - adding

---------------------

http://localhost/Shell.MVC2.Web.MediaService/PhotoService.svc/Rest/help/operations/addphotos

multiple photos or one phot can be added with this call:

Json body

{

"autoupload":true,

"multiple":true,

"photosuploaded":[{

"approvalstatusid":2147483647,

"caption":"String content",

"creationdate":"\/Date(928167600000-0500)\/",

"image":[81,

109,

70,

122,

90,

83,

65,

50,

78,

67,

66,

84,

100,

72,

74,

108,

89,

87,

48,

61],

"imagename":"String content",

"imagetypeid":2147483647,

"photostatusid":2147483647,

"rejectionreasonid":2147483647,

"size":2147483647

}],

"profileid":2147483647

}