**CLASS DESCREPTION**

1. MNuser: this is the data model of the user. We distinguish primary user and assistant user by the property userType. And for this version we just assume that a primary user is only paired with one assistant user.
2. So for each user. No matter it is a primary or assistant, the use his ID to fetch events from database, meanwhile, we use his paired user ID to fetch other events from database, these two part of events is the whole events witch associated with that user.

**HOW TO READ AND WRITE CODE USING THIS FRAMWORK**

1. the codes with the comment  //demo  is some demo for the using of some important method. Delete them when you code. All the important ios method is included in the demo code, And you can imitate them to do the detail job.

What you need to do is indicated by the comment within that code section.

For example:

- (IBAction)logInButton:(id)sender

{

//demo

//get the username and pin from database, if match performsegue, or give user some alert.

if([self.userNameTextField.text isEqualToString:@"hello"])

[self performSegueWithIdentifier:@"logIn" sender:self];

else

{

UIAlertView \*alert = [[UIAlertView alloc]

initWithTitle:@"Error"

message:@"Your user ID or password is invalid"

delegate:nil

cancelButtonTitle:@"OK"

otherButtonTitles:nil];

[alert show];

}

//demo

}

in this function. I just compare the string of UserNameTextField with “hello”(just for test), if they equals, perform the segue, otherwise just generate an alert view to tell the user. What you need to do is to make a query to the database and check whether this is a valid user and password, if it is then you call performSeguewithIdentifier to segue to the mainMenu, otherwise, give some alert to the user.

2. comment //??? Is the code you need to fulfill.

3. to make the textfield disappear after user type return. You need to add an uitextfileddelegate protocol in interface like this:

@interface PrimaryLoginViewController () <UITextFieldDelegate>

then add a method in the implementation like this:

//this method is to hide the textfield when the user type return

-(BOOL) textFieldShouldReturn:(UITextField \*)textField

{

[textField resignFirstResponder];

return YES;

}

then add this code in viewDidLoad() like this:

- (void)viewDidLoad

{

[super viewDidLoad];

self.userNameTextField.delegate = self.

}

remember that you need to set the self to be all the textfiled’s delegate.

3. If you are confused which kind of segue you should use. Just use the modal segue.