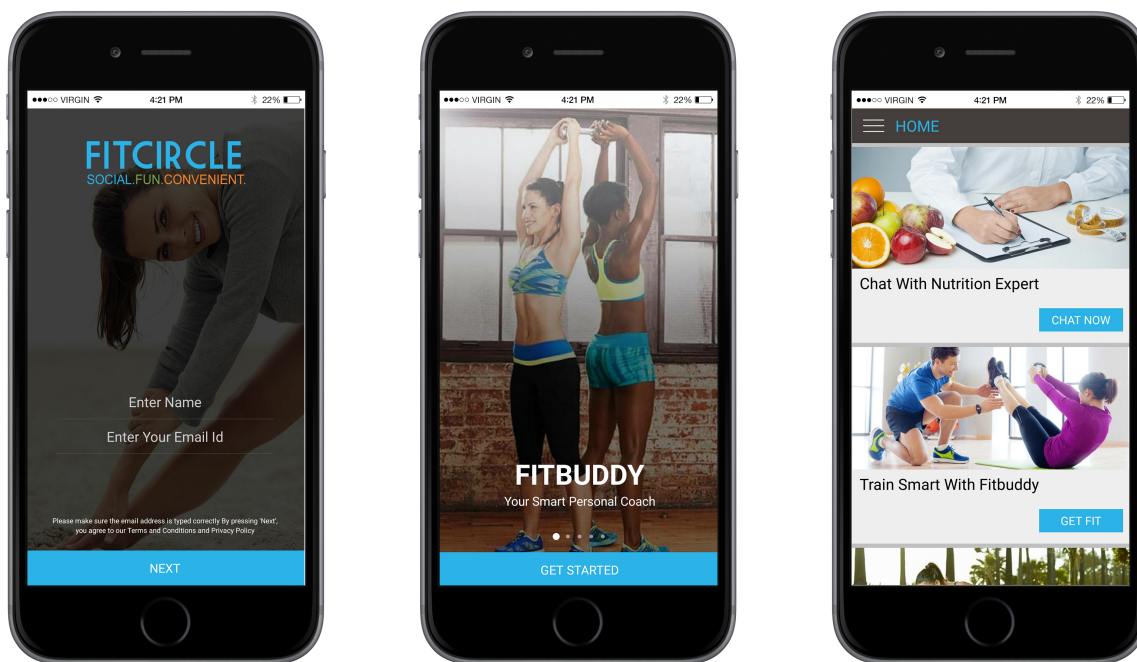


Embed iOS SDK in to a fitness app

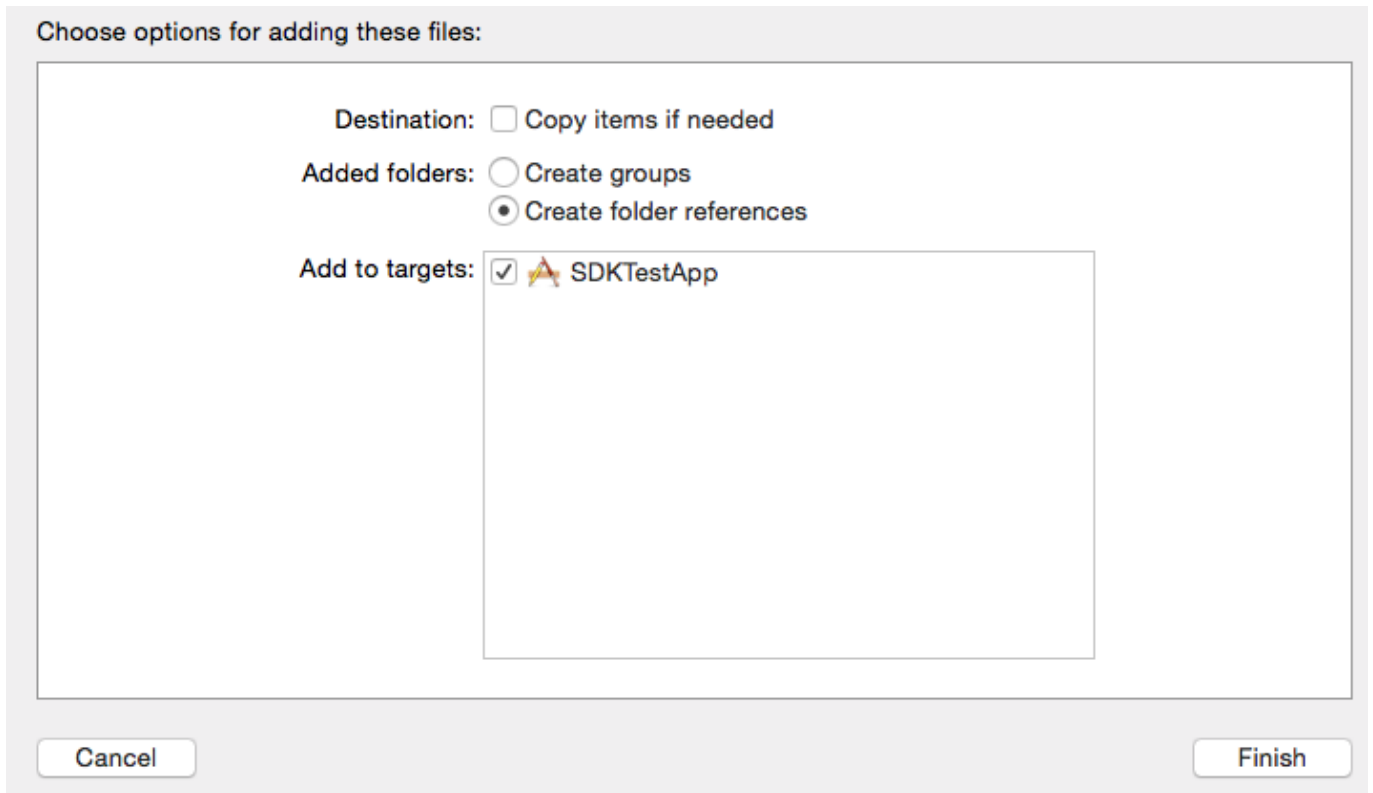


Objective:

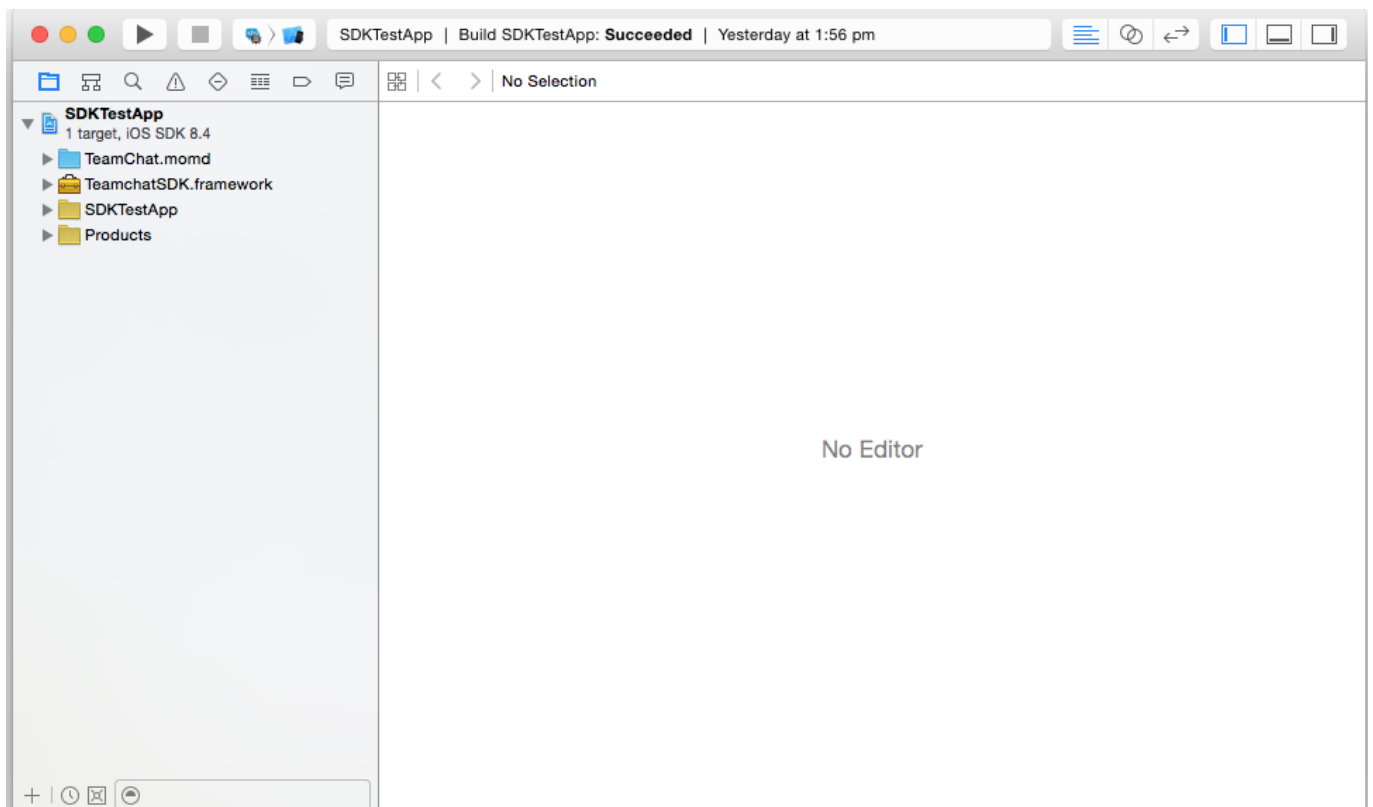
To embed smart messaging in to an existing iOS app for fitness

Implementation:

1. Add TeamchatSDK.framework to your project.
2. Add TeamchatSDK.framework to "Embedded Binaries" field under "General" tab of your project target.
3. Add TeamchatSDK.framework/TeamChat.momd folder to your project using the "Create folder references" option as shown below.



Now, project navigator will look like:



4. To support some HTTP requests in iOS 9.0 and above, add the key 'NSAppTransportSecurity' as a dictionary in Info.plist of your project and within the dictionary add the key 'NSAllowsArbitraryLoads' with BOOL value 'YES'.

Now you need to initialize the Teamchat object with appId in the AppDelegate method `application:didFinishLaunchingWithOptions:`.

```

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

    [Teamchat initializeWithAppID:<App ID> withCompletionHanlder:^(BOOL success,
NSError *error) {
        if (!success)
        {
            [[UIAlertView alloc] initWithTitle:@"Failed"
            message:[error localizedDescription] delegate:nil
            cancelButtonTitle:@"OK" otherButtonTitles:nil, nil] show];
        }
    }];

    return YES;
}

```

Enable remote notifications by setting deviceId in the AppDelegate method
application:didRegisterForRemoteNotificationsWithDeviceToken:

```

- (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {
    [Teamchat setRemoteNotificationsDeviceToken:deviceTokenString];
}

```

You can login to Teamchat in two ways:

1. Set Authentication code, emailID and userID.

```

[Teamchat setAuthenticationCode:@"<Auth-code>"];
[Teamchat setUserEmailID:@"<Email-ID>"];
[Teamchat setUserID:@"<User-ID>"];

```

2. Use Teamchat login view controller.

```

UIViewController *loginViewController = [Teamchat
teamchatLoginViewController:^(BOOL success, NSError *error, NSString *errMsg) {
    if (success)
    {
        // Successful login
    }
    else
    {
        // Login error
    }
}];

if (loginViewController)
{
    [self presentViewController:loginViewController animated:YES completion:nil];
}

```

If you want to configure UI of the Teamchat screens, you can set them as follows:

```
[Teamchat setNavigationTitle:@"Chats"];
[Teamchat setNavigationTitleColor:[UIColor greenColor]];
[Teamchat setTableViewSeparatorColor:[UIColor redColor]];
```

To get the chat groups list controller, you can use the following API:

```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UIViewController *teamchatGroupsViewController = [Teamchat
TeamchatGroupsViewController];
        if (teamchatGroupsViewController)
        {
            [self presentViewController:teamchatGroupsViewController animated:YES
completion:nil];
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];
```

+(void)initWithCompletionHandler:^(BOOL, NSError*, NSString*) handler method should be called before calling groups list API.

To get the chat window controller, you can use the following API:

```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        NSError *error;
        TeamchatGroup *group = [Teamchat teamchatGroupWithID:@"<GroupID>"
error:&error];

        UIViewController *groupViewController = [Teamchat
createTeamchatWindowViewControllerWithGroup:group];
        if (groupViewController)
        {
            [self presentViewController:groupViewController animated:YES
```

```

completion:nil];
    }
    else
    {
        // Error
    }
}
else
{
    // Error
}
}];

```

+(void)initWithCompletionHandler:^(BOOL, NSError*,NSString*)handler method should be called before calling chat window controller API.

To get user profile view controller, you can use the following API:

```

[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UIViewController *userProfileViewController = [Teamchat
userProfileViewController];

        if (userProfileViewController)
        {
            [self presentViewController:userProfileViewController animated:YES
completion:nil];
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];

```

+(void)initWithCompletionHandler:^(BOOL, NSError*,NSString*)handler method should be called before calling chat window controller API.

To get the public groups view controller, you can use the following API:

```

[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UIViewController *publicGroupsViewController = [Teamchat
publicGroupsViewController:^(TeamchatGroup *selectedGroup) {

```

```

        if (selectedGroup)
        {
            // Push the selected group's chat window.
            UINavigationController *chatWindowViewController = [Teamchat
createTeamchatWindowViewControllerWithGroup:selectedGroup];
            if (chatWindowViewController)
            {
                [self.navigationController
pushViewController:chatWindowViewController animated:YES];
            }
            else
            {
                // Error
            }
        }
    }];

    if (publicGroupsViewController)
    {
        [self presentViewController:publicGroupsViewController animated:YES
completion:nil];
    }
    else
    {
        // Error
    }
}];

```

+(void)initWithCompletionHandler:^(BOOL, NSError*, NSString*)handler method should be called before calling public group API.

To get Teamchat settings view controller, you can use the following API:

```

[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UINavigationController *settingsViewController = [Teamchat
settingsViewController];

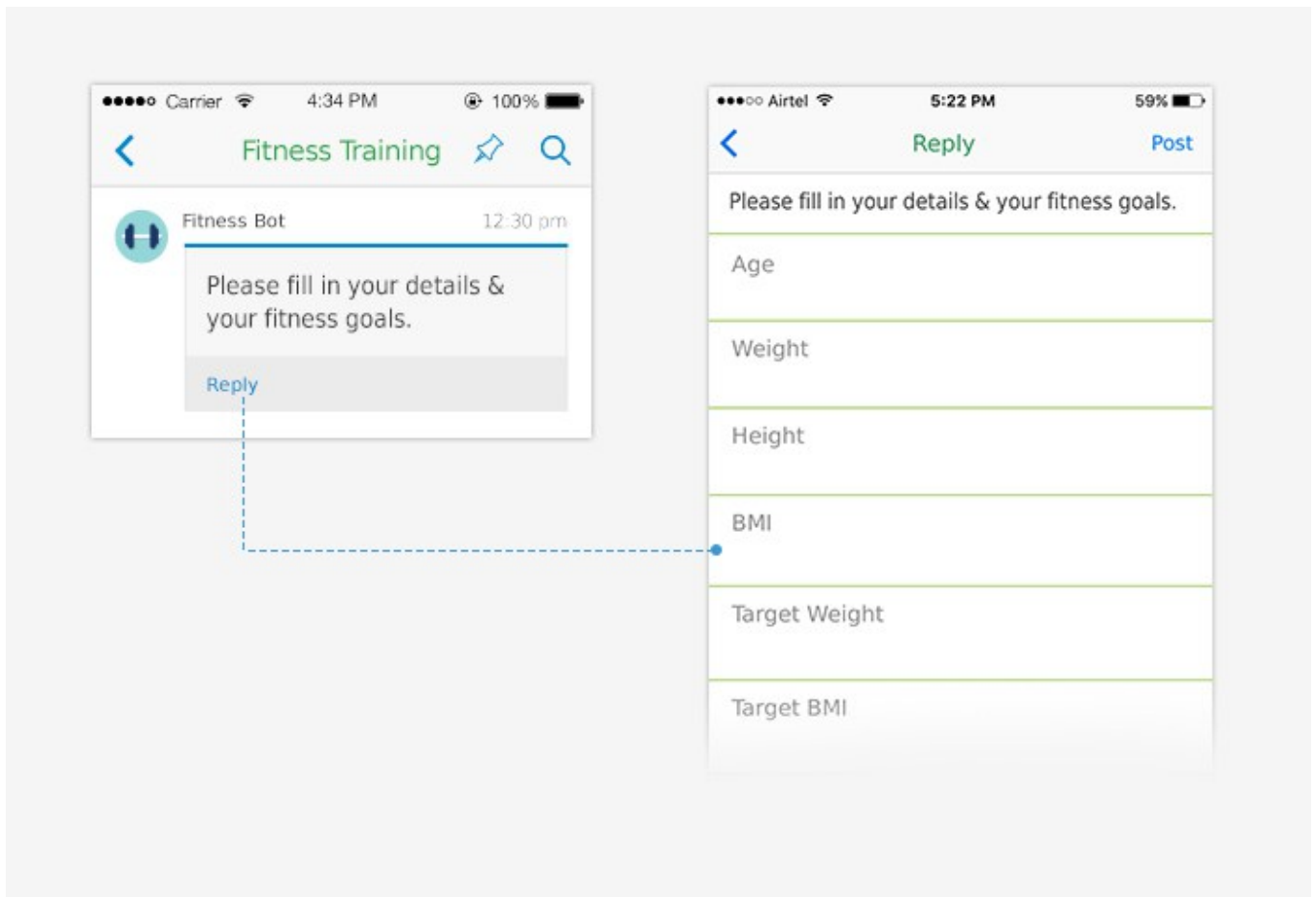
        if (settingsViewController)
        {
            [self presentViewController:settingsViewController animated:YES
completion:nil];
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];

```

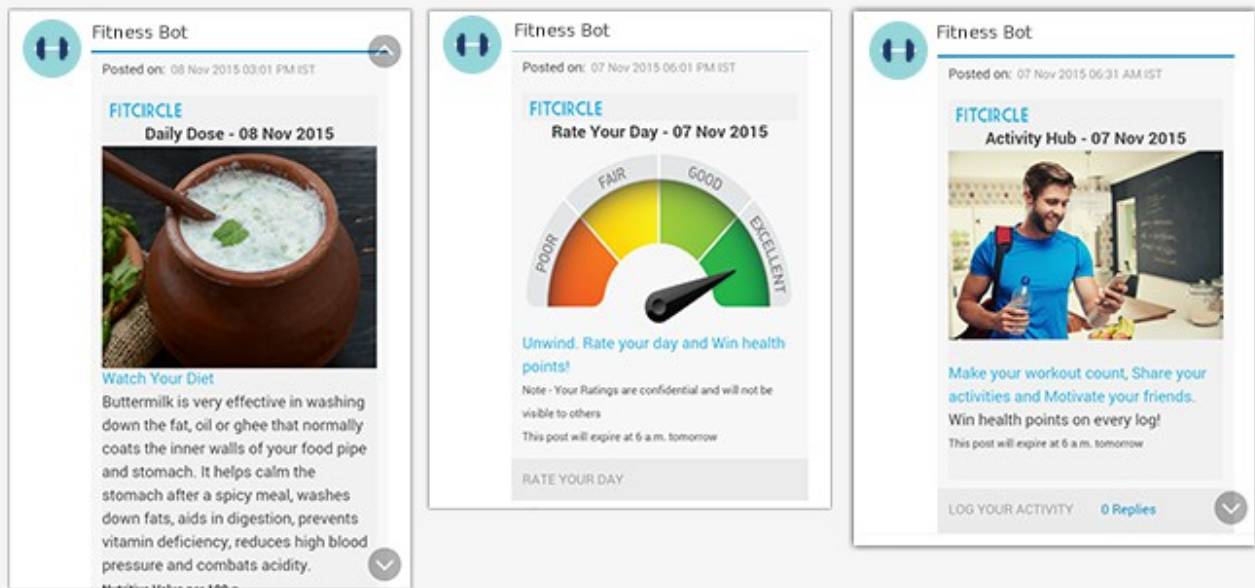
```
}  
}];
```

`+(void)initWithCompletionHandler:^(BOOL, NSError*, NSString*)handler` method should be called before calling public group API.

When new users came on board, they were asked to reply to a Teamchat message stating their fitness goals and motivations.



The trainers also share daily motivational quotes and messages on these groups.



Users are assigned a “fitscore” based on the consistency and intensity of their fitness training. Due to the ease of use, the company now builds their own messaging bots and workflows using Teamchat SDK.

