

实验 2



创建数据库、连接应用 并插入数据

实验结构

- ▶▶ 在 Xcode 中创建和使用 SQLite 数据库

实验目标

本实验结束后，你将能够：

- ▶▶ 创建 SQLite 数据表
- ▶▶ 连接到应用

导论

你想创建一个应用，让用户能够创建一个注册表单来获得申请人的详情。本节中，你将在 Xcode 中修改之前创建的应用，并将它同 SQLite 数据库连接起来。

实验：创建数据库、连接应用并插入数据

你已经创建了注册表单，下面你要创建并添加 SQLite 数据库到这一注册表单，这涉及到一些静态表，用于填入个人细节。

背景

SQLite 是一个开源库，由 C 写成。它实现了一个完备的 SQL 关系型数据库引擎。你可以使用 SQLite 来存储大量关系型数据。SQLite 数据库可以被用于 iPhone 和 iPad。

虽然 Core Data 应用编程接口（API）也被用于在 iOS 上存储数据，但其主要用途是保存应用创建的对象。使用 SQLite 更适合为应用预加载大量数据，而 Core Data 更适合管理设备上创建的数据。

本实验中，你需要执行下面这些任务：

1. 在 Xcode 中打开 RegistrationForm 应用。
2. 添加数据库库到项目。
3. 通过 terminal 创建一个数据库。
4. 在 Xcode 中添加代码来打开、创建和关闭 RegistrationForm 应用中的数据库。

实验准备

要执行这些任务，你需要有：

- iOS 8 和 Xcode
- 个人苹果帐户

实验推荐解决方案

1. 在 Xcode 中打开 RegistrationForm 应用，如图 1 所示：

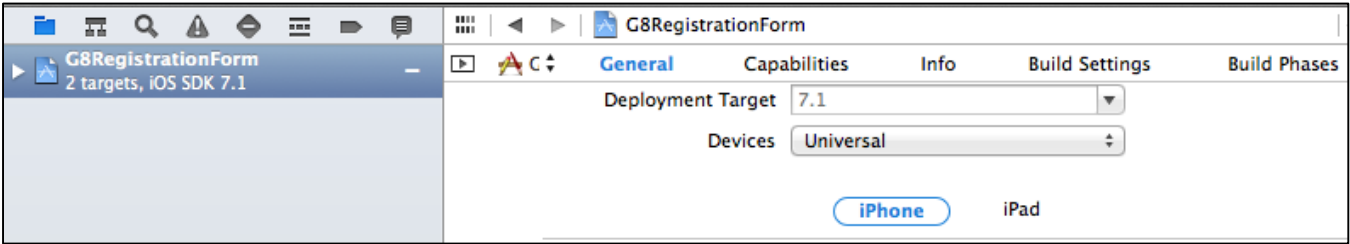


图 1: 在 Xcode 中打开 RegistrationForm 应用

2. 选择 **Linked Frameworks and Libraries** 部分（参见图 2），然后点+图标来添加新库，如图 3 所示：

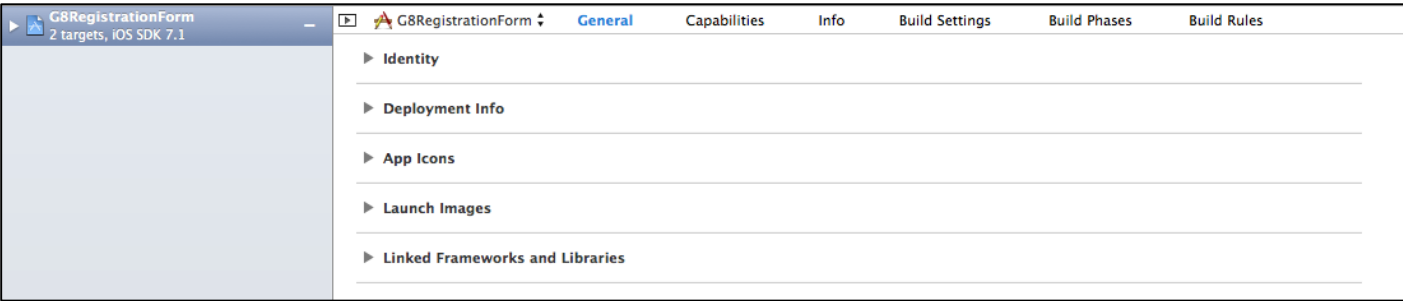


图 2: 点 Linked Frameworks and Libraries

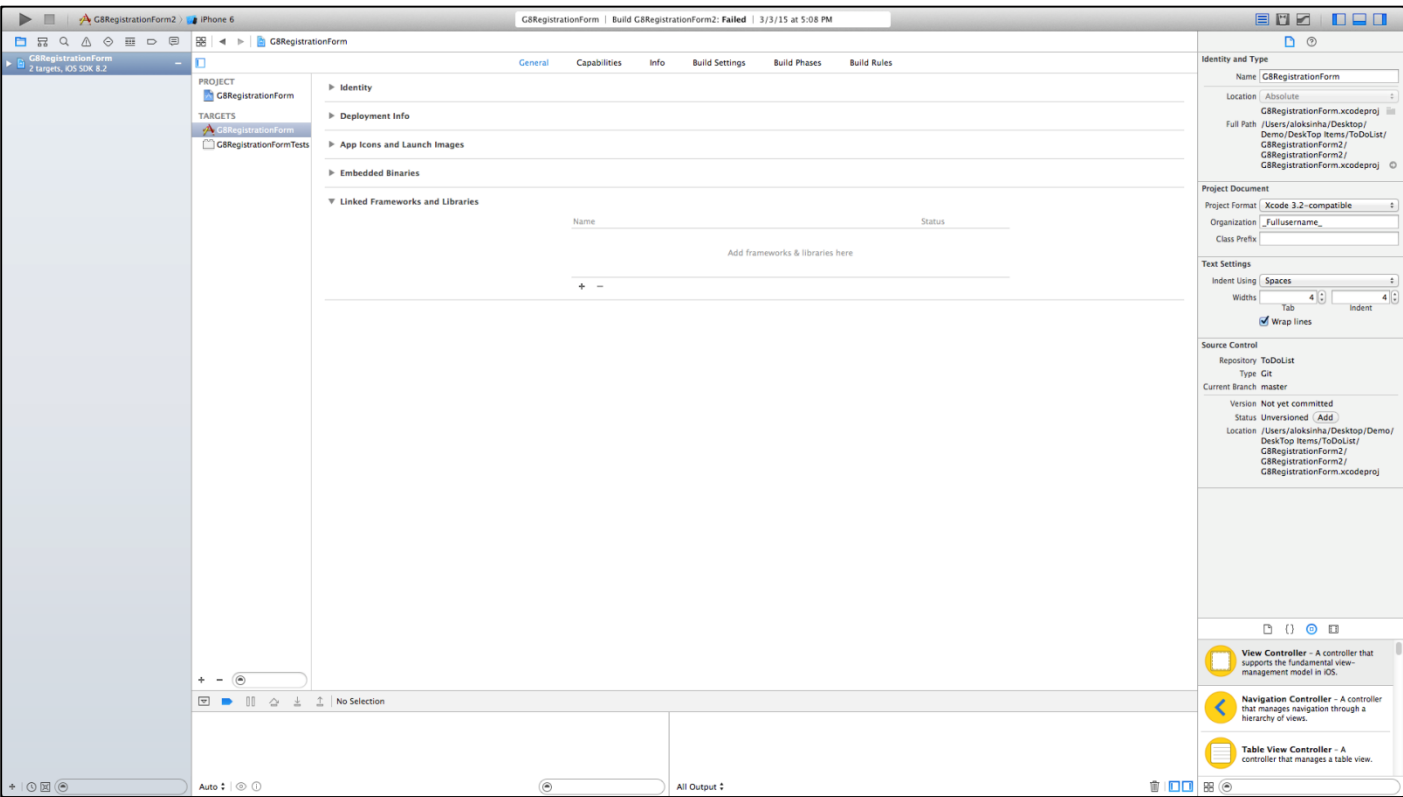


图 3: 点+图标来添加库

3. 在 **Choose frameworks and libraries to add** 对话框中，搜索 **libsqlite3.dylib** 文件并点 **Add** 按钮，如图 4 所示：

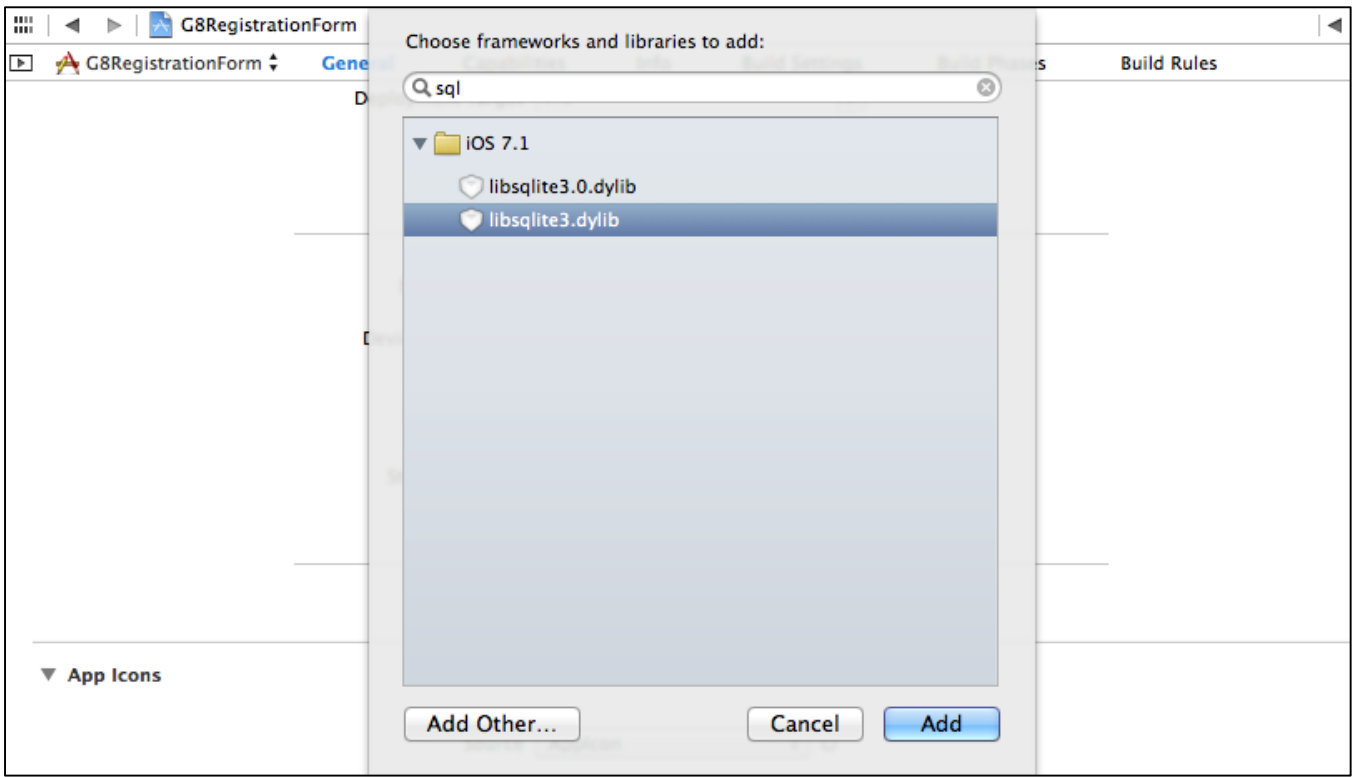


图 4: 添加一个 SQLite 库到项目

4. 下面使用 **terminal** 命令创建一个 SQLite 数据库。首先，输入 `# sqlite3 RegistrationForm.sql` 并使用如下命令创建一张表。`.table` 命令被用于表的检查，`.quit` 命令被用于退出 SQLite 数据库。

```
SQLite version 3.7.13 2012-07-17 17:46:21
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> Create TABLE RegistrationForm(name text, email text, phone text, company text, id
text);
sqlite> .table
RegistrationForm
Sqlite> .quit
```

5. 要添加 `RegistrationForm.sql` 到 Xcode，从 **File** 菜单选择 **Add Files to "G8RegistrationForm"** 选项，如图 5 所示：

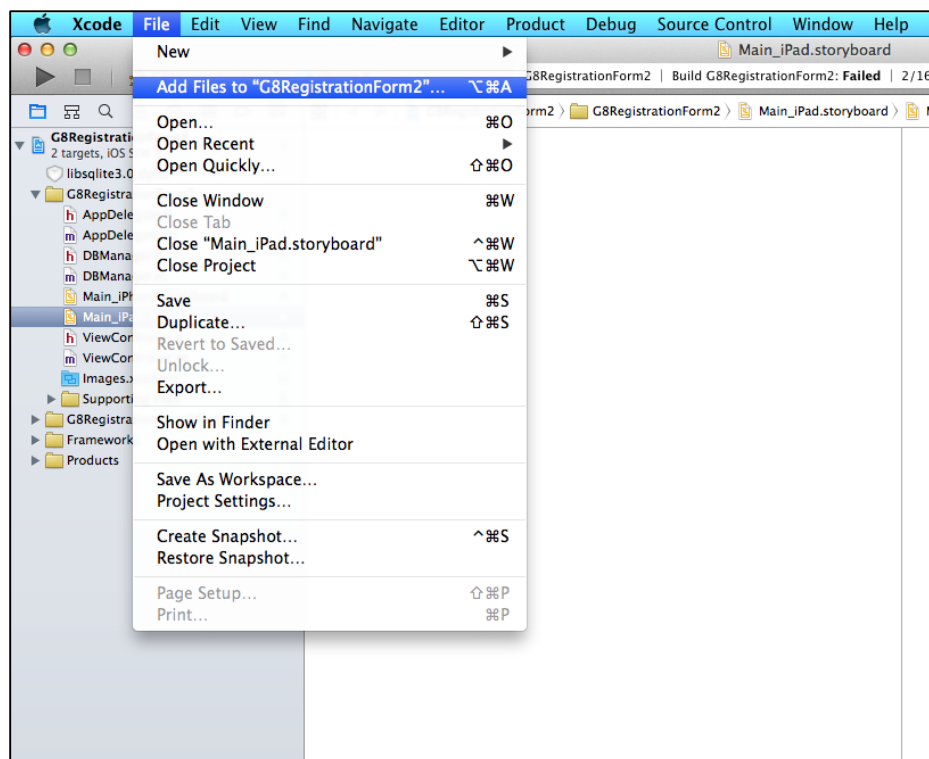


图 5: 选择 File -> Add Files to “G8RegistrationForm”

6. 选择你所创建的 RegistrationForm.sql。选择 **Copy items into destination group's folders** 复选框，然后点 **Add**，如图 6 所示。这会添加 RegistrationForm.sql 数据库文件。

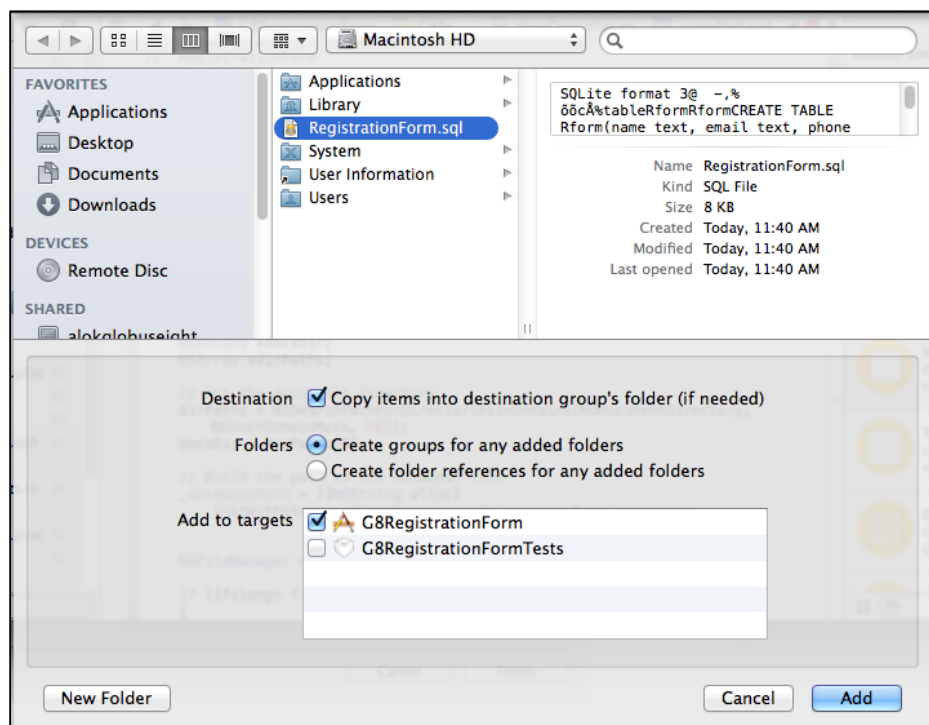


图 6: 添加 RegistrationForm.sql

7. 之后编辑 ViewController.h 文件，如下所示：

a. 获得 `databasePath` 及 `contactDB` 的代码：

```
- (IBAction)saveData:(id)sender; // Submit button to save data
@property (strong, nonatomic) NSString *databasePath; // Save database path
@property (nonatomic) sqlite3 *contactDB; // Contact database
@end
```

b. 在 ViewController.m 文件中，像下面这样编辑 (void)viewDidLoad 方法：

- 写代码来辨别应用的文档目录并构造一个路径到 RegistrationForm.sql 数据库文件。
- 写代码来创建一个 NSFileManager 实例，来探测数据库文件是否存在。如果数据库文件不存在，代码会将路径转化为 UTF-8 字符串并使用 SQLite `sqlite3_open()` 函数创建一个数据库。
- 最后，写代码来关闭数据库。

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    NSString *docsDir;
    NSArray *dirPaths;

    // Get document Directory path
    dirPaths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);

    docsDir = dirPaths[0];

    // Build path to the database file
    _databasePath = [[NSString alloc]
initWithString: [docsDir stringByAppendingPathComponent:@"RegistrationForm.sql"]];
    NSFileManager *filemgr = [NSFileManager defaultManager];
    if ([filemgr fileExistsAtPath: _databasePath] == NO)
    {
        const char *dbpath = [_databasePath UTF8String];

        // open database
        if (sqlite3_open(dbpath, &_contactDB) == SQLITE_OK)
        {
            char *errMsg;
            // Adding database if not present
            const char *sql_stmt = "CREATE TABLE IF NOT EXISTS RegistrationForm (EnterName text,
EnterEmail text, EnterPhone text, EnterCompany text , EnterID text)";
            if (sqlite3_exec(_contactDB, sql_stmt, NULL, NULL, &errMsg) != SQLITE_OK)
            {
                NSLog(@"Failed to create table");
            }
        }

        //Closing database
        sqlite3_close(_contactDB);
    }
}
```

```
        } else {
NSLog(@"Failed to open/create database");
        }
    }
}
```

- c. 在 `ViewController.m` 文件中，编辑 `(void) saveData:(id) sender` 方法，做法如下：
 - 写代码来保存表单数据到数据库中。在 `saveData` 中进行添加。
 - 这段代码会打开数据库文件，从表中提取文本，并将数据作为记录添加到数据库。

```
- (void) saveData:(id)sender
{
    sqlite3_stmt *statement;
    const char *dbpath = [_databasePath UTF8String];

    if (sqlite3_open(dbpath, &_contactDB) == SQLITE_OK)
    {
        NSString *insertSQL = [NSString stringWithFormat:@"INSERT INTO RegistrationForm (NAME text,
EMAIL text, PHONE text , COMPANY text, ID text ) VALUES (\"%@\", \"%@\", \"%@\", \"%@\",
\"%@\" )", _EnterName.text, _EnterEmail.text, _EnterPhone.text, _EnterCompany.text, _EnterID.text];

        const char *insert_stmt = [insertSQL UTF8String];
        sqlite3_prepare_v2(_contactDB, insert_stmt, -1, &statement, NULL);
        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            _EnterName.text = @"";
            _EnterEmail.text = @"";
            _EnterPhone.text = @"";
            _EnterCompany.text = @"";
            _EnterID.text = @"";
        }

        sqlite3_finalize(statement);
        sqlite3_close(_contactDB);
    }
}

@end
```

8. 添加代码到 ViewController.m 文件后，运行程序。运行可以点 **Product** 菜单下的 **Run**，也可以点 **Run** 按钮（参见图 8）。两者都能运行程序，如图 9 所示：

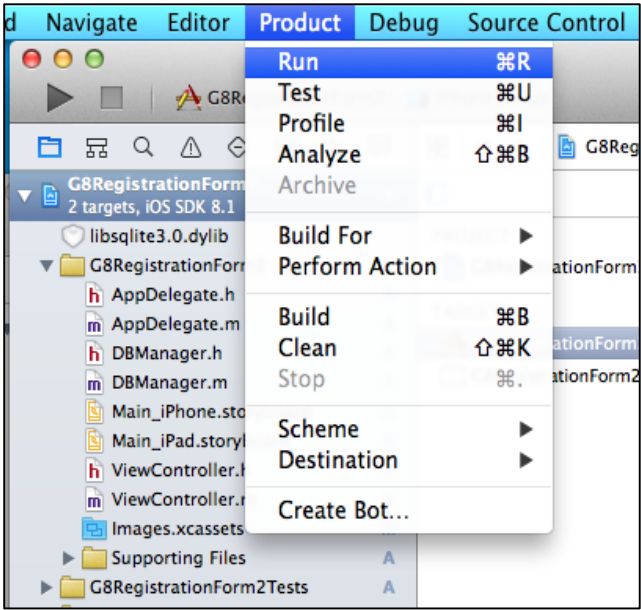


图 7: 选择 Product -> Run 来执行程序

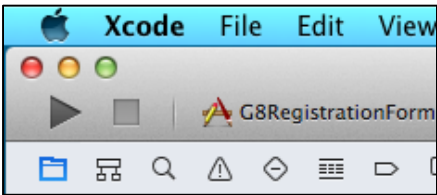


图 8: 点 Run

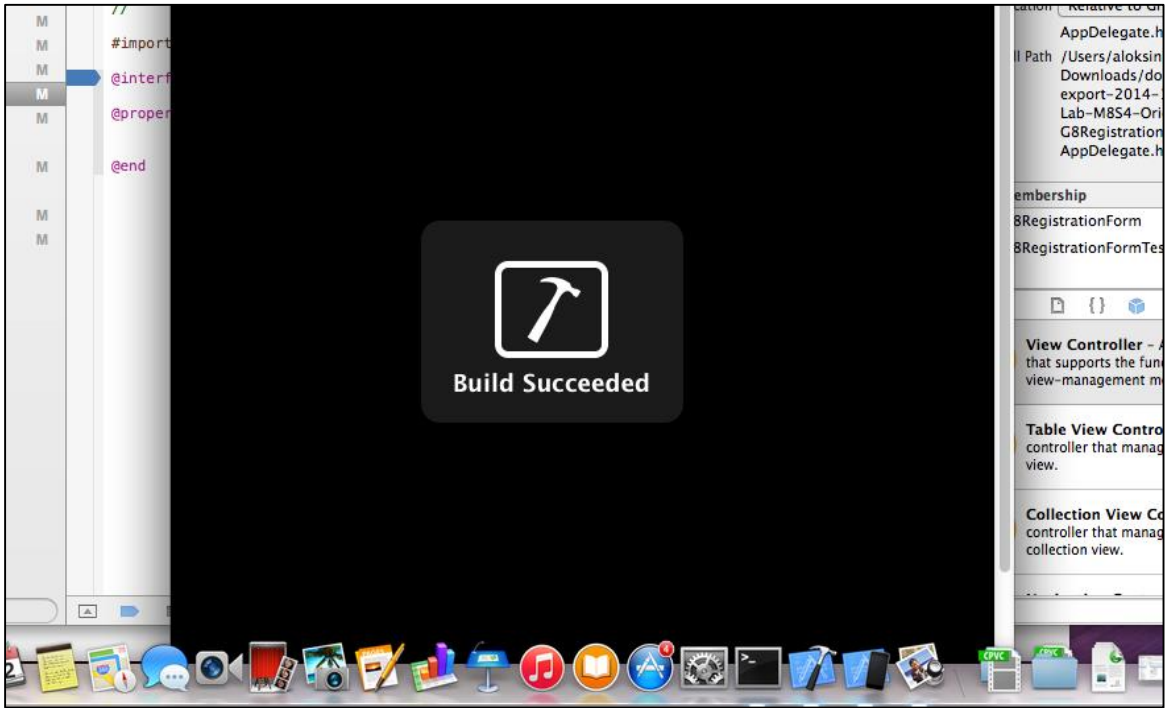


图 9: 运行 Xcode

9. 最后，注册表单得以创建，如图 10 所示。你可以添加细节到这个表单。

The screenshot shows an iOS Simulator window titled 'iOS Simulator - iPhone Retina (3.5-inch) / iOS 7.1 (11D167)'. Inside the simulator, there is a 'Registration Form' with five text input fields. The first field contains 'John', the second contains 'john@gmail.com', the third contains '898765490', the fourth contains 'xyz', and the fifth contains '12903'. A virtual keyboard is visible at the bottom of the simulator screen.

图 10: 输出画面：在表单中添加细节

源码

1. ViewController.h 文件

```
#import <UIKit/UIKit.h>
#import <sqlite3.h>

@interface ViewController : UIViewController
@property (strong, nonatomic) IBOutletUITextField *EnterName;
@property (strong, nonatomic) IBOutletUITextField *EnterEmail;
@property (strong, nonatomic) IBOutletUITextField *EnterPhone;
@property (strong, nonatomic) IBOutletUITextField *EnterCompany;
@property (strong, nonatomic) IBOutletUITextField *EnterID;

- (IBAction)saveData:(id)sender; // Submit button to save data
@property (strong, nonatomic) NSString *databasePath; // Save database path
@property (nonatomic) sqlite3 *contactDB; // Contact database
@end
```

2. ViewController.m 文件

```
#import "ViewController.h"

@interface ViewController ()
```

```

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    NSString *docsDir;
    NSArray *dirPaths;
    // Get document Directory path
    dirPaths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);

    docsDir = dirPaths[0];

    // Build path to the database file
    _databasePath = [[NSString alloc]
initWithString: [docsDir stringByAppendingPathComponent:
@"RegistrationForm.sql"]];

    NSFileManager *filemgr = [NSFileManager defaultManager];

    if ([filemgr fileExistsAtPath: _databasePath] == NO)
    {
        const char *dbpath = [_databasePath UTF8String];

        if (sqlite3_open(dbpath, &_contactDB) == SQLITE_OK)
        {
            char *errMsg;
            const char *sql_stmt = "CREATE TABLE IF NOT EXISTS RegistrationForm
(EnterName text, EnterEmail text, EnterPhone text, EnterCompany text , EnterID text)";

            if (sqlite3_exec(_contactDB, sql_stmt, NULL, NULL, &errMsg) != SQLITE_OK)
            {
                NSLog(@"Failed to create table");
            }
            sqlite3_close(_contactDB);
        } else {
            NSLog(@"Failed to open/create database");
        }
    }

    - (void) saveData:(id)sender
    {
        sqlite3_stmt *statement;
        const char *dbpath = [_databasePath UTF8String];
    }
}

```

```
if (sqlite3_open(dbpath, &_contactDB) == SQLITE_OK)
{
    NSString *insertSQL = [NSString stringWithFormat:@"INSERT INTO RegistrationForm (NAME text,
    EMAIL text, PHONE text , COMPANY text, ID text ) VALUES (\"%@\", \"%@\", \"%@\", \"%@\",
    \"%@\" )", _EnterName.text, _EnterEmail.text, _EnterPhone.text, _EnterCompany.text, _EnterID.text];

    const char *insert_stmt = [insertSQL UTF8String];
    sqlite3_prepare_v2(_contactDB, insert_stmt, -1, &statement, NULL);
    if (sqlite3_step(statement) == SQLITE_DONE)
    {
        _EnterName.text = @"";
        _EnterEmail.text = @"";
        _EnterPhone.text = @"";
        _EnterCompany.text = @"";
        _EnterID.text = @"";
    }

    sqlite3_finalize(statement);
    sqlite3_close(_contactDB);
}

@end
```