****

Submitted by:	Sindhuri Punyamurthula
Red ID:	820923656
DOS:	12/20/2016
Submission:	CS646_MoMa_Final_Project
File Name:	ReadMe.rtf

****

INTRODUCTION:

————————————

Welcome to MoMa! MoMa stands for Money Manager. This app not only helps user plan his/her expenditure better but also reminds, logs, alerts if paid and shows last and current month's balance. This way he/she can keep track of money

Concepts used:

—————————————

a. SQLite module used. Heavily used for storing, retrieving, managing, deleting fields, table rows etc.

b. Local Reminders. used in setting recurring monthly reminders as per user's wish and/or daily reminders as per user's wish.

c. UIPickerViews, UIImageview handling.

d. Table ViewController, Alerts, segues, dismissals, unwinds, Navigation ViewControllers, etc.

e. Appropriate keyboards, ViewController moving up/down when keyboard blocks fields etc.

NOTE 1:: ***No known warnings, compilation or run time issues seen. Only warnings were seen when moving from Xcode 8.0 to Xcode 8.1 and I fixed them***. Everything is done using iPhone 6s simulator. Xcode 8.1 shows iPhone 7 but runs as iPhone 6s simulator.

NOTE 2:: ** For Reminders to show up, the app should be in the background.

NOTE 3:: ** User has to sign up since all database files are created at local path. When signing up you can use your own ID, password .

LOGIN ViewController:
—————————

a. Here, he/she signs up with his/her preferred username and password by clicking signup button.
   - Must signup because of local DB creations
   - This helps keep track of number of users and
   - Stores all the values in a Login table(Local SQL table)

b. Once signed up, user is directed to the DashBoard page (explained later)

c. From second time onwards, he/she is directly taken to Dashboard page.

DASHBOARD ViewController:
——————————————

a. This page is pre-populated with major/common recurring monthly expenses. (such as Rent, Insurance, Credit Bills, Groceries, Gym, etc.). Initially all values are 0's. A gray check image represents unpaid expense.

b. User can swipe left on the cell to edit his/her pre-populated expense values (such as Title, Amount, Due, Reminders, Card used to pay, website if the expense can be paid online etc. Once done, user clicks save and most important values are shown in the dashboard. Edit page unwinds to Dashboard page on save

c. User can use the RushPay! feature to pay the due at that time. This will take him/her to Safari app of iOS.

d. User can swipe left to delete the expense if unnecessary. This should alert the user and confirm deletion.

e. If paid, user can click simply on gray check image and it will turn from gray to green. This action will also alert the user (in case there were any unwanted touches).

f. Dashboard page also has Logout, Settings, Summary, Daily and Add Row features which are explained briefly below.

LOGOUT ViewController:
———————————

a. Self explanatory, LogsOut and shows Login page. Logs in only if correct UserID and Password is entered.

SETTINGS ViewController:
————————————

a. User can change Spending Limit and set Daily Reminders. (Should click Yes for setting reminders)

SUMMARY ViewController :
—————————————

a. Shows Last Month's Balance/Savings, Current Expenditure, Current Balance and Expenditure log (if user spent/paid that day and logs)

DAILY ViewController :
———————————

a. User can log his/her daily expenses here and this shows up in Summary VC as an embedded table view

ADD ROW (EditViewController):
————————————————

a. If the user wants to add his/her important recurring expense to the DashBoard.