

Sardar Khan

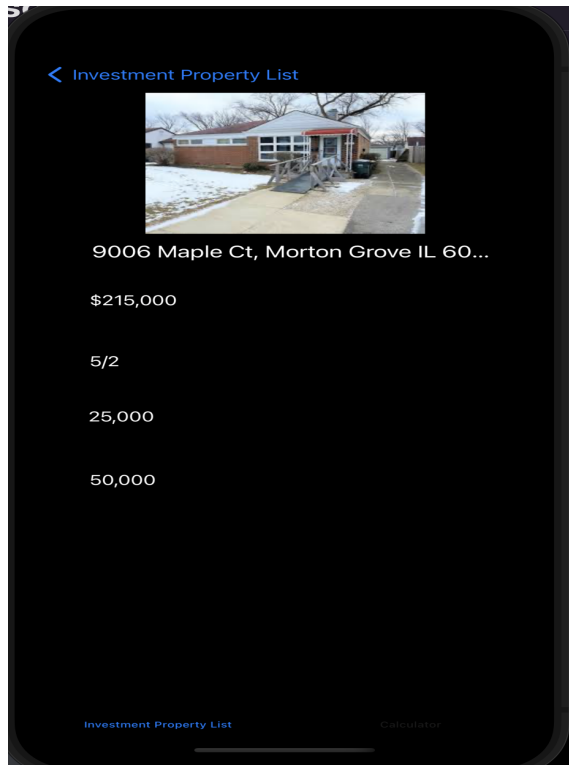
The Feature used in this project were the following:

- UIAlertController
 - Used this to collect the rental income amount before going to the final screen
- MFMessageComposeViewController
 - This was used on the final page to send a text of the final results in final page
- MessageUI
 - We imported this in order to use the Apple messenger UI, I was not able to test this since the messenger UI does not work on a xcode simulator, but when comparing to others seemed to be legit working code
- AVAudioPlayer
 - Used this when final results were calculated on the final page screen, you can hear the cha ching kind noise on the viewdidload of the screen
- Table controller view
 - Used this store the data - was not able to finish this feature in time. But we were supposed to be able to add a table cell with mortgage information from the final page
- Segues
 - Used segues to go through the flow of the application hopping from one screen to another

First Screen is a master list of all the saved properties from before. It's a table view controller with each cell being part of a property cell class with the data needed.



When a cell in the master list is clicked you are brought to a new view controller which is connected through a segue and the data with the correlating cell of an array with the objects of property cell is sent to this view controller. All the data is shown in the following format.



The next screen is shown when you click on the right tab view titled Calculator. I regret not making the keyboard a number pad but if I could go back I would make all these input number pads! This input only takes numbers and anything else will make the app crash. A number pad would have avoided this issue. The version 2 before I put the app store will definitely have this feature. But here the user inputs the data needed to get the principal interest monthly payment. I have made it where the user can either put a percentage or cash amount for Down Payment and the other one will automate. The rest are simple UI text fields and UISegmentedControl

The screenshot shows a mobile application interface with a white background and a black status bar at the top. The status bar displays the time 9:36, signal strength, and battery level. The app interface includes the following elements:

- Property Price:** A text label followed by a text input field containing the value "200000".
- Down Payment:** A text label followed by a text input field containing "\$ 20000" and a percentage input field containing "10.0 %".
- Interest Rate:** A text label followed by a text input field containing "2.0" and a percentage label "%".
- Loan Period:** A text label followed by three segmented buttons labeled "5 Year", "15 Year", and "30 Year".
- Next Button:** A blue button with the text "Next" located at the bottom right of the form area.
- Footer:** Two links, "Investment Property List" and "Calculator", are located at the bottom of the screen.

The next page we can go two routes depending on what kind of property it is. If in the first `UISegmentedControl` you have selected Primary residence then we don't need the following Management fees, Misc fees, and rental income. So we won't be presented with the Alert screen you'll see later down the app. So if it's a primary residence you'll simple just need to fill out Property Taxes and Insurance.

9:37

< Back

Primary Residence Rental/Investment Property

Principle & Interest Amount

665

Property Taxes:

\$ 10000 /year 5.0 %

Insurance: 11000

Management Fees:

\$ 5,000 /year 3.0 %

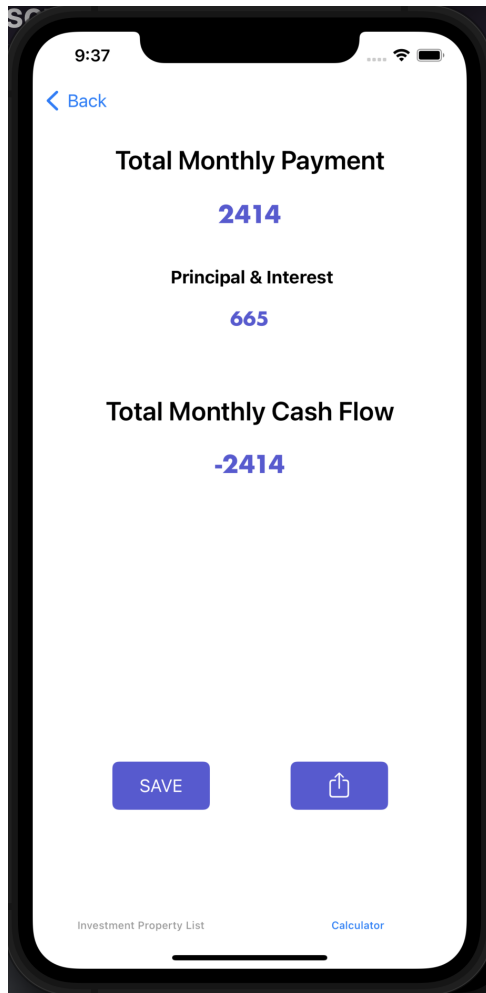
Misc/Vacancy Fees:

\$ 5,000 /year 3.0 %

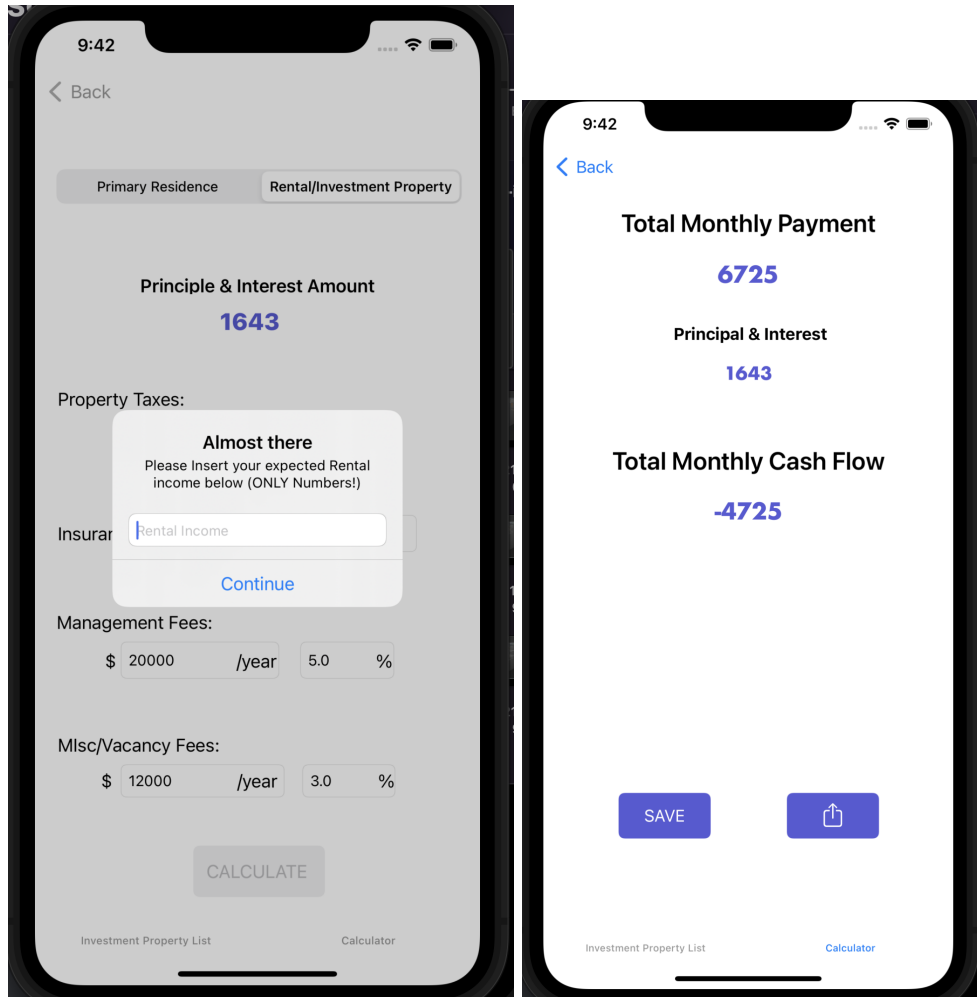
CALCULATE

Investment Property List Calculator

Once we submit our primary residence data in we get to our final page which allows you to add to the master list in the first page and also send a message to anyone about the results of your mortgage, which i created through MessageUI and MFMessageComposeViewController. You also hear a cha ching sound when this page first loads using AVAudioPlayer. You are unable to see the test message UI since it does not show in Xcode simulator. I also did not configure the master list to match the class currently in the master list and did not add the code for it. As I was already burnt out and will create in the version 2 of this app.



Now going back the the second page if you choose rental/investment property for the UISegmentedControl then you must fill out the MIsc fees and the Management fees. Once you click calculate you'll be shown a UIAlertcontroller where you must submit only numbers of the total rental income from your property. Once you click ok. Then the segue will be called. This got a little tricky since the segue was not performing since the alert acts as a view controller itself.



This project had a lot of challenges from sending data through segues to using API features I've never used such as MessageUI and the AVAudioPlayer. As well as the population of the percentage of the cash amount and vice versa. Also the architecture of the tabbed view to the navigation controller and using segues to connect the view controllers in the flow. As well on the left tab - using a table view controller and using cells and create a class for the cells was very nitty gritty and made my head spin. But it was a great learning experience of Swift and IOS dev.

If I could go back I would first change the UITextField fields to number pads to solve the issue of users putting something other than numbers. That's the first. Next thing I would change is of course integrate my master list in the project by first changing the objects used in the property cells and the propertyDetails screen. I would have to change the class to integrate my mortgage info in my MortgageCalcFirst.swift file. I would then add the data and create a new cell when the IB button save was pressed. It's not too hard. But I just felt too burnt out to finish this feature. I would then save my data to the dom so my array of data in my masterlist will be saved even when the app is closed and opened etc. I would lastly make the app better looking with, logos and a color theme with more structure in the input fields and cool data visualization in my results page when showing the total monthly payment amount.

Overall, I fell in love with IOS dev and all the cool features Apple offers. Will be adding all my projects to my github account and my resume. Thank you Professor Panchal for this awesome class. Looking forward to the Senior Project with you next Quarter.