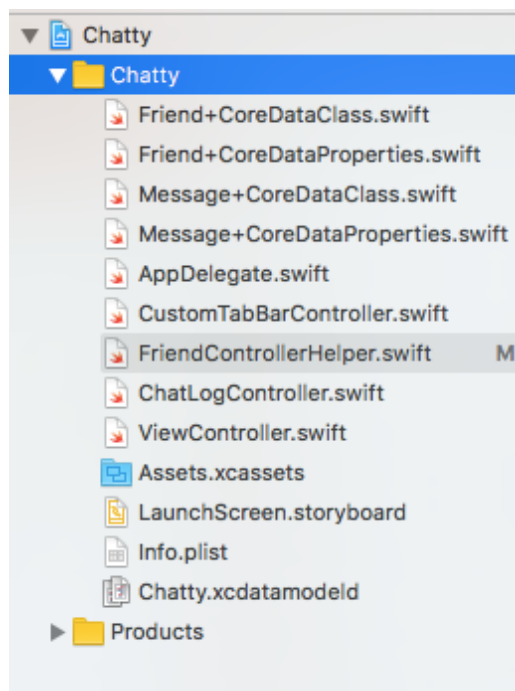


# Project Report

## ---- Development of a chat App, “Chatty”

Since pace of life is increasing, the way of communicating has changed. It get shorter and more frequent, that’s why chat App emerged and it becomes more and more popular.

### 1. Project Structure:

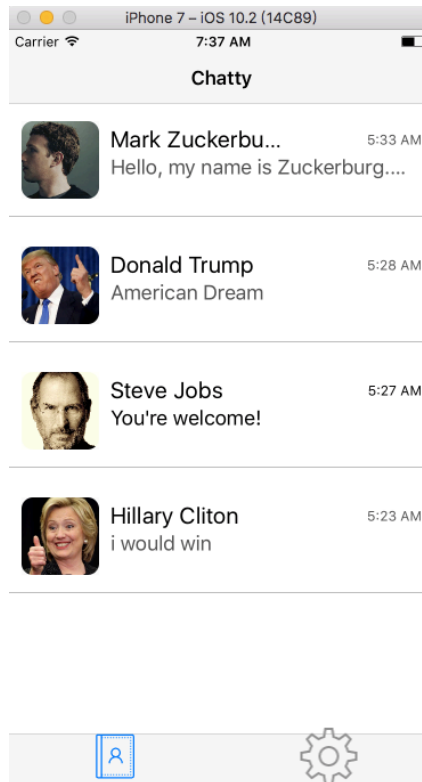


The project is divided into five sections, which are Main View, Friend List, Chat Log and Core Data.

### 2. Development Process

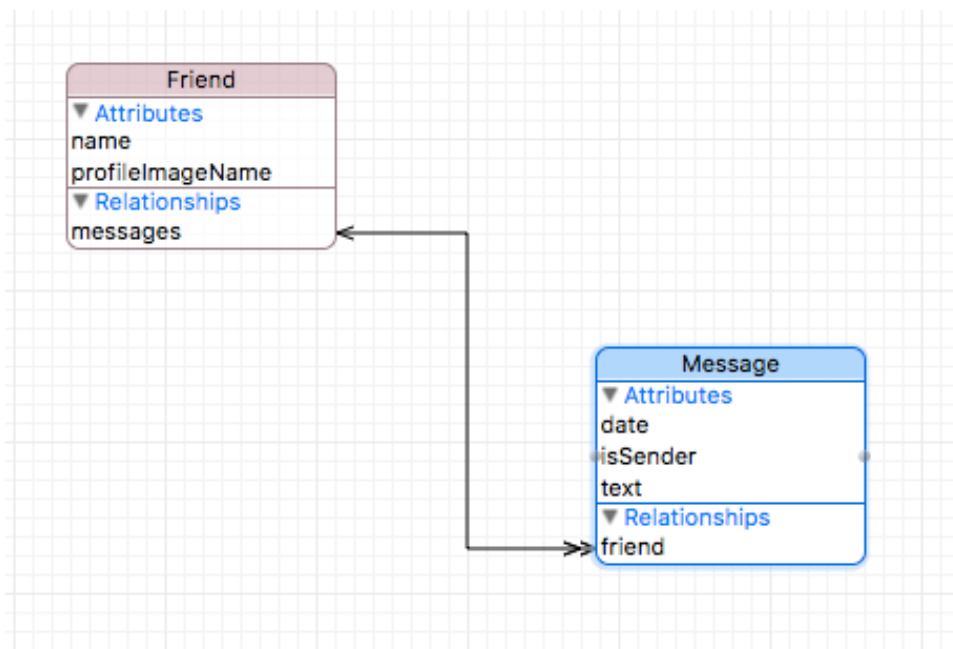
There are four stages in the process of developing, basic layout, core data, chat bubbles and show with simulator.

#### 2.1 Basic Layout



Build the friends list, and put the name, profile image, latest message and time into each cell. All friends cells are sorted based on time which is real time.

## 2.2 Core Data



Create two entities in the core data, Friend and Message. The relationship is one to many. Friend has two attributes which are name and profileImageName. Message has three attributes which are date, text and isSender.

## 2.3 Chat Bubbles

isSender is one of Message's attribute which decides if message is receiving message or sending message. It's Boolean type so that the location and color of message and bubble even if the message should be with a profile image are able to be decided.

```
if message.isSender{
    cell.messageTextView.frame = CGRect(x: view.frame.width-estimatedFrame.width-16-16, y:
    0, width: estimatedFrame.width+16, height: estimatedFrame.height+10)
    cell.textBubbleView.frame = CGRect(x: view.frame.width-estimatedFrame.width-16-8-16,
    y: 0, width: estimatedFrame.width+16+8, height: estimatedFrame.height+20)

    cell.profileImageView.isHidden = true

    cell.textBubbleView.backgroundColor = UIColor(colorLiteralRed: 34, green: 139/255,
    blue: 34/255, alpha: 1)
    cell.messageTextView.textColor = UIColor.white
}
else {
    cell.messageTextView.frame = CGRect(x: 48+8, y: 0, width: estimatedFrame.width+16,
    height: estimatedFrame.height+10)
    cell.textBubbleView.frame = CGRect(x: 48, y: 0, width: estimatedFrame.width+16+8,
    height: estimatedFrame.height+20)

    cell.profileImageView.isHidden = false

    cell.textBubbleView.backgroundColor = UIColor(white: 0.95, alpha: 1)
    cell.messageTextView.textColor = UIColor.black
}
}
return cell
}
```

## 2.4 Show with Simulator

There is a button in the chat log, "simulator", which simulates server side and interacts with user. Messages sent by simulator are set just one minute ago.



```

func simulate() {
    let delegate = UIApplication.shared.delegate as? AppDelegate
    let context = delegate?.persistentContainer.viewContext

    // insert message before the latest message
    let message = FriendsController.createMessageWithText(text: "Here is a message
    which is sent from simulator just one mins ago.", friend: friend!,
    minsAgo: 1, context: context!)

    messages?.append(message)
    messages = messages?.sorted(by: {$0.date!.compare($1.date! as Date) == .
    orderedAscending})
    if let item = messages?.index(of: message){
        let receivingIndexPath = NSIndexPath(item: item, section: 0)
        collectionView?.insertItems(at: [receivingIndexPath as IndexPath])
    }
}

```

### 3. Communication Principle:

How to get multiple messages from a friend? It is divided into three subparts.

#### 3.1 Get all messages in order:

Solution: Sort all the messages in descending order based on time

#### 3.2 Distribute all the message

Solution: create a predicator that filters all messages based on a singular friend

#### 3.3 Show the latest message in main view

Solution: fetch the most recent message using power of core data