package com.checkout.payments.addtip  
  
import android.view.View  
import androidx.lifecycle.MutableLiveData  
import androidx.lifecycle.ViewModel  
import com.chase.payments.CurrencyFormatter  
import com.chase.payments.Payment  
import com.chase.payments.PercentFormatter  
import com.chase.payments.isZero  
import com.chase.preferences.tip.TipsUseCase  
import com.cp.mpos.R  
import java.math.BigDecimal  
  
class AddTipViewModel(var payment: Payment, var tipsUseCase: TipsUseCase) : ViewModel() {  
  
 var amountLabel: MutableLiveData<String> = MutableLiveData()  
 var tipBreakdown: MutableLiveData<String> = MutableLiveData()  
 var tipBreakdownVisibility: MutableLiveData<Int> = MutableLiveData()  
  
 var customTipVisibility: MutableLiveData<Int> = MutableLiveData()  
 var tip1Visibility: MutableLiveData<Int> = MutableLiveData()  
 var tip2Visibility: MutableLiveData<Int> = MutableLiveData()  
 var tip3Visibility: MutableLiveData<Int> = MutableLiveData()  
 var tip1Text: MutableLiveData<String> = MutableLiveData()  
 var tip2Text: MutableLiveData<String> = MutableLiveData()  
 var tip3Text: MutableLiveData<String> = MutableLiveData()  
 var tip1TextVisibility: MutableLiveData<Int> = MutableLiveData()  
 var tip2TextVisibility: MutableLiveData<Int> = MutableLiveData()  
 var tip3TextVisibility: MutableLiveData<Int> = MutableLiveData()  
  
 init {  
 customTipVisibility.*value* = View.*GONE* tip1Visibility.*value* = View.*GONE* tip2Visibility.*value* = View.*GONE* tip3Visibility.*value* = View.*GONE* tip1TextVisibility.*value* = View.*GONE* tip2TextVisibility.*value* = View.*GONE* tip3TextVisibility.*value* = View.*GONE* updateAmounts()  
 setupTip(tipsUseCase)  
 }  
  
 fun setupTip(tipsUseCase: TipsUseCase){  
  
 val tipPresetButtonVisibility = *arrayOf*(tip1Visibility, tip2Visibility, tip3Visibility)  
 val tipPercentButtonVisibility = *arrayOf*(tip1TextVisibility, tip2TextVisibility, tip3TextVisibility)  
 val tipPresetButtonsText = *arrayOf*(tip1Text, tip2Text, tip3Text)  
 val tipIsPercent = tipsUseCase.isTipInPercent()  
  
 for(i in tipsUseCase.tipPresets().*indices*){  
  
 val tipButtonText = tipPresetButtonsText[i]  
 val tipPercentButton = tipPercentButtonVisibility[i]  
 val tipButtonVisibility = tipPresetButtonVisibility[i]  
 val tip = tipsUseCase.tipPresets()[i].tip  
  
 if(tip > BigDecimal.*ZERO*) {  
 tipButtonText.*value* = if (tipIsPercent)PercentFormatter.format(tip) else CurrencyFormatter.format(tip)  
  
 payment.setTip(tip,tipIsPercent)  
  
 tipButtonText.*value* = if(tipsUseCase.isDisplayCalculatedAmountsEnabled() && tipIsPercent) tipButtonText.*value* + "\n"+ CurrencyFormatter.format(payment.tip(false)) else tipButtonText.*value* tipButtonVisibility.*value* = View.*VISIBLE* tipPercentButton.*value* = if (tipsUseCase.isDisplayCalculatedAmountsEnabled() && tipIsPercent) View.*VISIBLE* else View.*GONE* }  
 else{  
 tipButtonVisibility.*value* = View.*GONE* tipPercentButton.*value* = View.*GONE* }  
 }  
  
 clearTip()  
 customTipVisibility.*value* = if(tipsUseCase.isAllowCustomTipEnabled()) View.*VISIBLE* else View.*GONE* }  
  
 fun setTipPercent(buttonId: Int){  
 try {  
 when(buttonId){  
 R.id.*tipOneButton* -> payment.setTip(tipsUseCase.tipPresets()[0].tip, tipsUseCase.tipPresets()[0].percent)  
 R.id.*tipTwoButton* -> payment.setTip(tipsUseCase.tipPresets()[1].tip, tipsUseCase.tipPresets()[1].percent)  
 R.id.*tipThreeButton* -> payment.setTip(tipsUseCase.tipPresets()[2].tip, tipsUseCase.tipPresets()[2].percent)  
 R.id.*noTipButton* -> clearTip()  
 }  
  
 }  
 catch(e: NumberFormatException){  
 clearTip()  
 }  
  
 updateAmounts()  
 }  
  
 fun setTip(tip: BigDecimal, tipInPercent: Boolean){  
 payment.setTip(tip, tipInPercent)  
 updateAmounts()  
 }  
  
 fun updateAmounts(){  
 tipBreakdownVisibility.*value* = if(payment.tip(true).*isZero*()) View.*GONE* else View.*VISIBLE* tipBreakdown.*value* = CurrencyFormatter.format(payment.tipableAmount()) + " + " + CurrencyFormatter.format(payment.tip(false)) + " tip"  
 amountLabel.*value* = CurrencyFormatter.format(payment.total())  
 }  
  
 fun clearTip() {  
 payment.setTip(BigDecimal.*ZERO*, false)  
 }  
  
}