# Pseudo Code

# Yingtong Zhou

(1) MainActivity

class MainActivity extends AppCompatActivity:

onCreate(Bundle savedInstanceState):

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

//Connect checkPosition and options buttons to item id

//Initialize a FileManager object to maintain inputs

//Set activities for the buttons

(2) ListOfStocks

class ListOfStocks extends AppCompatActivity implements StockAdapter.ItemClickListener:

StockAdapter adapter

onCreate(Bundle savedInstanceState):

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_list\_of\_stocks)

//Connect addPosition and viewPortfolio buttons to item id

//Create ArrayList<UserStockPosition> to store inputs

if (inputs !isEmpty()):

emptyList.setVisibility(INVISIBLE)

userStockPositions = MyApplication.getApplication().getAllUserStockPosition()

else:

emptyList.setVisibility(VISIBLE)

//Create and connect recycler view

//Set onClickListener for buttons

(3) AddPosition

class AddPosition extends AppCompatActivity:

private DatePickerDialog datePickerDialog

private Button dateButton

private Date purchaseDate

// Initialize a SimpleDateFormat object

onCreate(Bundle savedInstanceState):

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_add\_position)

initDatePicker()

//Bundle date button to activity

showDatePickerDialog:

datePickerDialog.show()

initDatePicker():

DatePickerDialog.OnDateSetListener dateSetListener = new DatePickerDialog.OnDateSetListener():

onDateSet(DatePicker datePicker, int year, int month, int day):

month = month + 1

String date = makeDateString(day, month, year)

dateButton.setText(date)

try:

purchaseDate = curFormater.parse(day+"/"+month+"/"+year)

catch (ParseException e):

e.printStackTrace()

//Initialize a Calendar object, get year, month, day, and style

datePickerDialog = new DatePickerDialog(this, style, dateSetListener, year, month, day)

makeDateString(int day, int month, int year):

return getMonthFormat(month) + " " + day + " " + year

getMonthFormat(int month):

//switch-case: convert int month to String expression

addPosition(View view):

//Connect textviews to data

//Create an UserStockPosition object, set the stock symbol, stock name, number of stock, and stock price

(4) PositionDetails

class PositionDetails extends AppCompatActivity:

onCreate(Bundle savedInstanceState):

super.onCreate(savedInstanceState)

Bundle b = getIntent().getExtras()

assert b != null

int index = b.getInt(AuxClass.INTENT\_TAG\_STOCK)

//Create a UserStockPosition object to take inputs from the form

//Set TextView stockSymbol, stockName, stocksNumber, purchasePrice, purchaseDate, lastUpdated, gainsLosses, lastMarketPrice

//Set the color of texts according to gain/loss

if(stockGains == 0):

gainsLosses.setTextColor(YELLOW)

elif (stockGains > 0):

gainsLosses.setTextColor(GREEN)

else:

gainsLosses.setTextColor(RED)

(5) ProfitVisual

class ProfitVisual extends AppCompatActivity:

onCreate(Bundle savedInstanceState):

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_data\_analysis)

//Create an ArrayList<UserStockPosition> object to take inputs

if (inputs !isEmpty()):

emptyList.setVisibility(INVISIBLE)

//Take inputs

else:

emptyList.setVisibility(View.VISIBLE)

for UserStockPosition userStockPosition in userStockPositions:

if userStockPosition.getStockName() != null:

//Calculate gain/losses

//Set text colors as above

(6) StockAdapter

private List<UserStockPosition> mData

private LayoutInflater mInflater

private ItemClickListener mClickListener

StockAdapter(Context context, List<UserStockPosition> data):

mInflater = LayoutInflater.from(context)

mData = data

ViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener:

TextView stockName, buyPrice, sellPrice, stockGains

ViewHolder:

super(itemView)

//Connect variables to unique item id

itemView.setOnClickListener(this)

onClick:

if (mClickListener != null) mClickListener.onItemClick(view, getAdapterPosition())

onCreateViewHolder(ViewGroup parent, int viewType):

View view = mInflater.inflate(R.layout.stock\_layout, parent, false)

return new ViewHolder(view)

onBindViewHolder:

//Get inputs and bind them to front end

UserStockPosition userStockPositionAux = mData.get(position)

Double auxStockGains = StockOperations.CalculateCurrentOperationPosition(userStockPositionAux)

BigDecimal stockGainsBD = new BigDecimal(auxStockGains).setScale(2, RoundingMode.HALF\_UP)

double stockGains = stockGainsBD.doubleValue()

holder.stockName.setText(stockName)

holder.buyPrice.setText(buyPrice)

holder.sellPrice.setText(sellPrice)

// Setting the color of the text depending on Gains or losses

getItemCount():

return mData.size()

getItem(int id):

return mData.get(id)

setClickListener(ItemClickListener itemClickListener):

mClickListener = itemClickListener

interface ItemClickListener:

onItemClick(View view, int position)