**Personal Code documentation**

1. TaiKhoan
   1. Why does TaiKhoan declare NhanVien inside it and not the other way around?
      1. This is done because NhanVien might be declared in other classes such as SanPham and Kho. As these other classes do not need the TaiKhoan part of NhanVien, hence, declaring TaiKhoan inside NhanVien is impractical and is a redundancy.
   2. Why when deleting NhanVien inside the GUI doesn’t delete the NhanVien inside the SQL Table?
      1. Because we assume on default case that deleting a row inside said GUI is equal to firing that specific NhanVien, which we will set their status to “KhongConHoatDong” instead of prematurely deleting them to preserve their data. Yes, this will in fact clog the TaiKhoan table but it’s impractical to assume that a company can reach over 1 million users, so duplicate username/password will not be a problem soon.
   3. Why should we save the NhanVien’s IMAGE\_PATH to the database and not the image itself?
      1. This is to prevent the database from overloading with images. So, instead of directly saving the images to the database, I implemented a cache folder to save that image and call it from the IMAGE\_PATH whenever the program starts.
2. PanelCapNhatSinhVien
   1. filterTable(real-time searching function)
      1. How it works:
         1. It creates a temporary DefaultTableModel and parse the data of tblTaiKhoan into it.
         2. It then removes all the rows inside it IMMEDIATELY, then, it checks if the JTextField is empty, if the JTextField is empty, it loads the data by using a for-loop. If it’s not empty, then it uses containsSearchText() to return any row that has a matching String (non-case-sensitive).
3. WindowLogin\_GUI
   1. Why is there no regex check/validateData() inside the class?
      1. This is to prevent hackers who have never used the application before to guess pre-requirement for such username/password. E.g if we were to show a warning that password requires 8 or more characters, the hackers will immediately eliminate other possibilities.
4. SwingSidebarHelper
   1. Why is the class a separate function and not directly implemented inside WindowMain\_GUI?
      1. This is to prevent WindowMain\_GUI from being bloated, plus, the class can be reused on the future.
   2. At line 54, why is the event declared directly instead of calling a separate @Override ActionListener?
      1. This is due to SwingSidebarHelper being a static class, yes, it can still be declared using menuItem.this(); But, it’s the only variable whose ActionListener needs to be implemented.
   3. At line 61, why do we call xPanel.refreshData()? This is to make sure that the dynamic array being used by any 2 or more given panels completely synchronize after switching to one another. RefreshData() is called within the GUI class.
5. Database initialization
   1. Why are there 2 SQL files for createDatabase and useDatabase?
      1. This is because if we put createDatabase and useDatabase inside a same SQL, when we execute it at runtime JDBC will perform both queries and throw and error.
6. SQLs
   1. insertTaiKhoan
      1. IF NOT EXISTS ( SELECT 1 FROM taiKhoan WHERE tenDangNhap = ? OR maNhanVien = ?)
         1. This sql checks if the taiKhoan/nhanVien has already existed or not. If it doesn’t exists, it will perform adding the taiKhoan/nhanVien normally.
   2. deleteTaiKhoan
      1. DELETE FROM taiKhoan WHERE maNhanVien = ?;
      2. UPDATE nhanVien SET trangThai = 'KhongConHoatDong' WHERE maNhanVien = ?;
         1. We use this query to Set the trangThai of NhanVien to ‘KhongConHoatDong’ instead of actually deleting it to keep records of fired employees.
7. DAO
   1. JDBC
      1. stmt.executeQuery() => For SELECT stuff (often pair with rs.next());
      2. stmt.executeUpdate() => For INSERT/UPDATE/DELETE;
   2. Constructor
      1. Inside the constructor, we should always call
      2. public DanhSachTaiKhoan\_DAO() {
      3. conn = ConnectDB.getConnection();
      4. }
      5. to get
      6. connection = DriverManager.getConnection(url, user, password);
      7. inside ConnectDB