Code Review

During the development process the team came across inconsistencies with the codes formatting. This document is a record of those inconsistencies.

Code reviewed	Issue(s)	Action	Ammended by
Authorisation class – authorisation check method	Variables in Authorisation class are not camel case	Amend variable names so that they comply with the camel case code etiquette.	Bram
Document class – .print() method	The methods print layout was inconsistent in comparison to the User class .getUserInfo() method	We changed the .print() method so that information in displayed the same format as .getUserInfo(), instead of on one line	Sam
Document class	Most comments missing from Document class	Applied comments to relevant methods	Max
Yuconz_project_app class	Comments missing for .createPersonalDetails()	Applied comments to relevant methods	Latifah
	Comments missing for .ammendPersonalDetails()		
	Comments missing for .readPersonalDetails()		

Evidence

```
*/
public boolean authorisationCheck(User CurrentUser, String UserIn, String Action) {
    switch (Action) {
        case "readPersonalDetails" :
            if(CurrentUser.getDepartment().equals("Human Resources") || CurrentUser.getDepartment(
                return true;
            }
            break;
        case "createPersonalDetails" :
            if(CurrentUser.getDepartment().equals("Human Resources")){
                return true;
            }
            break;
        case "amendPersonalDetails" :
            if(CurrentUser.getDepartment().equals("Human Resources")){
                return true;
            }
```

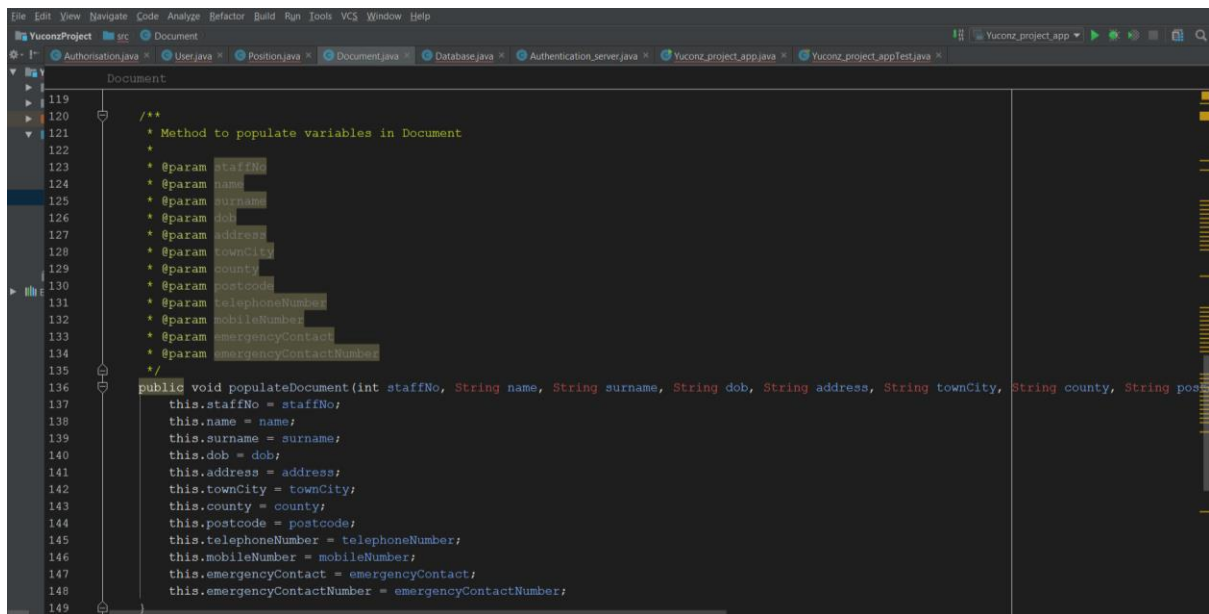*Figure 1: Authorisation class with capitalised variables*

BEFORE

AFTER

```
public void print(){
    System.out.println(staffNo + " "   + name + " "   + surname + " "   + dob + " "
+ address + " "   + townCity + " "   + county + " " + postcode + " "   + telephoneNumber
+ " "   + mobileNumber + " "   + emergencyContact + " "   + emergencyContactNumber);

    System.out.println(
                    "Staff No: " + staffNo + "\n" +
                    "Name: " + name + "\n" +
                    "Surname: " + surname + "\n" +
                    "Date Of Birth: " + dob + "\n" +
                    "Address: " + address + "\n" +
                    "Town/City: " + townCity + "\n" +
                    "County: " + county + "\n" +
                    "Postcode: " + postcode + "\n" +
                    "telephone Number: " + telephoneNumber + "\n" +
                    "Mobile Number: " + mobileNumber + "\n" +
                    "Emergency Contact Name: " + emergencyContact + "\n" +
                    "Emergency Contact Number: " + emergencyContactNumber + "\n"
        );
    }
}
```

IN COMPARISION

```
    public String getUserInfo()
    {
        return ("User ID:        " + getUserID() + "\n" +
                "Username:       " + getUsername() + "\n" +
                "Name:           " + getName() + "\n" +
                "Position:       " + getPosition().getPositionName() + "\n" +
                "Supervisor:     " + getSupervisorID() + "\n" +
                "Subordinates: " + getSubordinatesString() + "\n" +
                "Department:     " + getDepartment()
                );
    }
```

Figure 2: print formatting has changed to be identical to .getUserInfo()

Figure 3: Adding comments to method .populateDocument() in Document class

```java
/**
 * prints personal details of UserIn from a Document if authorised
 * @param userIn
 * @return
 */
public boolean readPersonalDetails(String userIn){
    //run authorisation method with readPersonalDetails as an action
    if(authorisation.authorisationCheck(currentUser, userIn, action: "readPersonalDetails")){
        if(database.checkExists(userIn)) {
            Document doc = database.fetchPersonalDetails(userIn);
            doc.print();
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

```java
/**
 * adds a new user record to database is authorised
 * @param userIn
 * @return
 */
public boolean createPersonalDetails(String userIn){
    //run authorisation method with createPersonalDetails as an action
    if(authorisation.authorisationCheck(currentUser, userIn, action: "createPersonalDetails")){
        if(!database.checkExists(userIn) && database.checkExistsEmployee(userIn)) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

```java
/**
 * Adjusts an existing user record if authorised
 * @param userIn
 * @return
 */
public boolean amendPersonalDetails(String userIn){
    //run authorisation method with amendPersonalDetails as an action
    if(authorisation.authorisationCheck(currentUser, userIn, action: "amendPersonalDetails")){
        if(database.checkExists(userIn)){
            Document doc = database.fetchPersonalDetails(userIn);
            doc.print();
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

Figure 4: Added comments to methods in Yuconz_project_app

References

https://git.cs.kent.ac.uk/co510/26c/commits/master

https://git.cs.kent.ac.uk/co510/26c/commit/f3d8fb17e52e57b3b30b711e795bace7d6df6a59

https://git.cs.kent.ac.uk/co510/26c/issues/1