# BBM487 -Software Engineering Laboratory

## Library Book Loan System
## Coding Standards

**Group 4:**

**Ali Burak ÖZAKINCI - 21427162**

**Sabit ÇATALTAŞ - 21692544**

**Aishat Temitope OLOMOWEWE – 21603775**

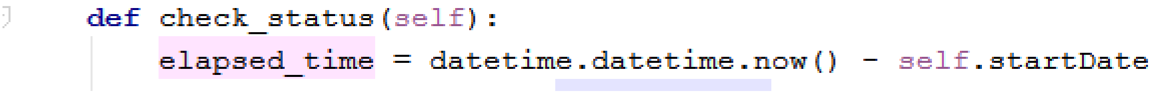
This document gives coding conventions for the Python code we used in this software.Pep-8 Python Coding Standard is followed mainly.

A style guide defines the consistency for the written code.Consistency in code provides consistency on whole project.Also another aspect we focused in this standards is readability.

Specifically ,this document covers naming conventions, indentation, statements, comments,tab-space preference and imports.

# Indentation

Use 4 spaces per indentation level.



```
def check_status(self):
    elapsed_time = datetime.datetime.now() - self.startDate
```

This rule is optional for continuation lines.

# Tabs or Spaces?

Spaces are preffered in project implementation.Python 3 does not allows mixture of tabs and spaces.Since Python 3.5 used in this project,space indentation is the way to choose.

# Maximum Line Length

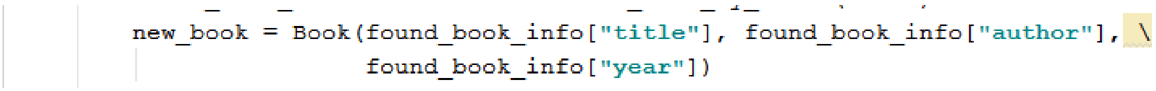
Maximum line length is 79 characters with respect to pep-8.

For comments and docstrings, the line length should be 72 characters.

Inline css statements of PyQT has no line length rule.

These line values are appropriate for working with split views or multi windows.

Backslashes should be used split long lines.



```
new_book = Book(found_book_info["title"], found_book_info["author"], \
                found_book_info["year"])
```

# Blank Lines

Top-level function and class definitions are surrounded with two blank lines.

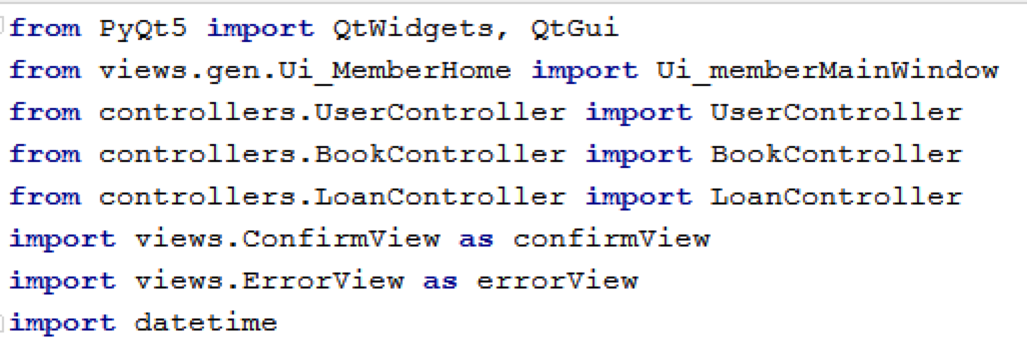
Method definitions inside a class are surrounded by a single blank line.

# Source File Encoding

All source files use UTF-8 encoding.

# Imports

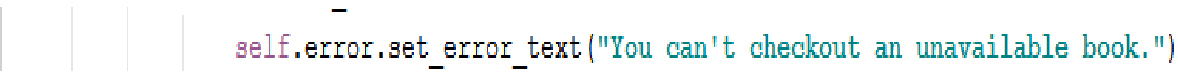
Imports should be on separate lines.



```
from PyQt5 import QtWidgets, QtGui
from views.gen.Ui_MemberHome import Ui_memberMainWindow
from controllers.UserController import UserController
from controllers.BookController import BookController
from controllers.LoanController import LoanController
import views.ConfirmView as confirmView
import views.ErrorView as errorView
import datetime
```

# String Quotes

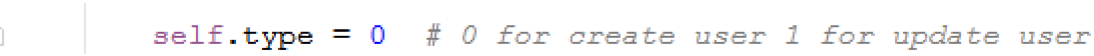
Double quotes or single quotes does not matter on the Python side but we stick to the double quotes in our project.



```
                self.error.set_error_text("You can't checkout an unavailable book.")
```

# Comments

Inline comments should be used starting a '#' and a space.

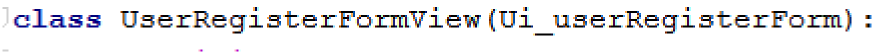


```
        self.type = 0  # 0 for create user 1 for update user
```

# Naming Conventions

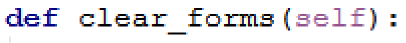
In general, for naming situations “lower_case_with_underscores” and “mixedCase” naming styles should be used.

Class names should use the CapitalizedWords convention.



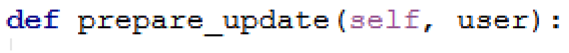
```
class UserRegisterFormView(Ui_userRegisterForm):
```

Function names should use “lower_case_with_underscores” style.



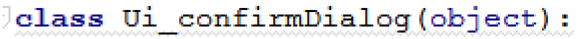
```
def clear_forms(self):
```

Self must be always used for the first argument to instance methods.



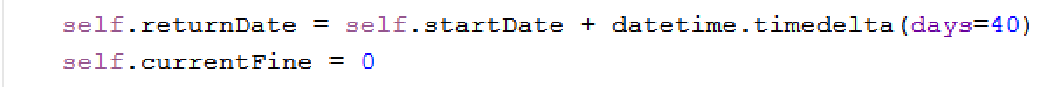
```
def prepare_update(self, user):
```

Generated interface files start with “Ui_”.



```
class Ui_confirmDialog(object):
```

Mixed case or “lower_case_with_underscores” style should be used on variable names.



```
self.returnDate = self.startDate + datetime.timedelta(days=40)
self.currentFine = 0
```