Maintenance Plan

Introduction

The postgraduate management system (PGMS) was developed using Django, an open-source, free high-level web framework written in Python. The Django version we used for this project is Django version 1.11.3, which was the latest version during the start of the development process. The version of Python we used is Python 3.6.1. The detailed documentation of the framework can be referred from here: https://docs.djangoproject.com/en/1.11/

Purpose

The Purpose of this post maintenance plan in to ensure that there is an on going support for the post graduate management system (PGMS) and to ensure that the system will be manage properly even when it is pass on to the next person

Design Pattern

Django web framework at it's core uses the Model–view–controller(MVC) architecture. It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database ("Model"), a system for processing HTTP requests with a web templating system ("View"), and a regular-expression-based URL dispatcher ("Controller").

Database Maintenance

With this architecture, developer was saved from the hassle of writing SQL query, any database table can be created easily by creating Python classes. Besides, Django supports multiple database engine, developer can create data models without worrying the database engine the website uses. The database we currently uses is the lightweight SQLite. However, it can be easily changed to other server like MySQL or PostgreSQL by changing the DATABASE settings in settings.py to have a more powerful and robust database system.

Documentation

The system is broken down into different modules and are handled by different member from the development team. The ERD diagram mapped out the overall entity classes in the system, but the implementation of the entity classes are split between modules as well. All entity classes are stored in the models.py file in every module's folder. For each modules, the user interface (HTML files) are stored in the template folders in each module. The logic behind each page are stored in the views.py file in each module, and are documented as a sequence diagram for each functions.

General Recommendations

An effective operation, inspection, and maintenance program includes:

- Check for outdated content and update accordingly to the stakeholders requirement. All of those includes:
    - Users status

        - Set student to inactive if student have not pay their bills or for valid reason
        - Set lecture to inactive if they are not teaching anymore
    - Subject
        - Add subject
        - Delete subject
        - Update subject details

Further Enhancement/Improvement On Current System

Every system goals should be to improve from time to time to create a more efficient and valuable system that is able to meet the stakeholders goals. The following shows the improvement that can be make according to module and GUI as a separate module:

- Application Module
    - Can be further improve after admin approval an application, the system will automatic sent an email to the applicator.
- Appointment Module
    - Can be further improve to allow student to make appointment from own appointment page by searching for lecturer from the page itself as part of the form on appointment page.
- Transaction Module
    - Student able to pay from the transaction page instead of just verifying the payment made.
    - Administrator can have notification showing that someone had make payment / had verified transaction.
    - Transaction could have a deadline for the student to pay, if not they will be prevented to login the system.

- Login Module
    - Users can access their account through user authorization (email, contacting administrator, or both)
- Subject
    - Timetable scheduling can be done for subject.
- Profile Module
    - More interaction that can be done between user eg. simple chat system

Additional module / function that can be added to the current system

Stakeholders requirement changes from time to time. Therefore, time to time, the system may need to add new module and new functional as well as non functional requirements. The following shows the module / function that can be added to the system:

- Dashboard
    - Lecturer's reminder and materials of the course can be upload, and appear at Home page
    - Download materials from the dashboard