proposal.md

# COMS 3101 Python Project Proposal

Wode "Nimo" Ni UNI:wn2155

## 1. Purpose

In this project, we will implement a lightweight, easy-to-use, command-line GTD software, DonePy. The primary purpose of the project is to demonstrate our understanding off the Python language and showcase some features of the language. Also, the author has a particular interest in command-line GTD programs, because he often procrastinate, and work on the terminal all the time.

## 2. Features

DonePy is a command-line to-do management system. It is able to create, search, retrieve, and modify to-do items. The items can be associated with customizable tags and due times. Also, using JSON files, DonePy will support import and export of the database, allowing the user to migrate his/her data across machines, or back it up on the cloud.

To add more complexity to the system, we will allow subtasks inside of a task. For example, a large project for Programming languages and translators can be broken down into smaller milestones such as (1) Build the parser; (2) implement codegen; (3) test etc.

## 3. Correct Behaviors

- Sample usage of the program: a simple case

```
# View the current todos
$ donepy -v
Hello Nimo, you currently have 3 tasks to do:
    - [1] Write the project proposal for python
    - [2] Do OS Homework
    - [3] Think about why you ended up doing OS and PLT together and regret it for 10 minutes
# Add in a todo item
$ donepy -a
Please enter the title:
Go to sleep earlier today
Add additional info? [y/n] n
# The new list
$ donepy -v
Hello Nimo, you currently have 4 tasks to do:
    - [1] Write the project proposal for python
    - [2] Do OS Homework
    - [3] Think about why you ended up doing OS and PLT together and regret it for 10 minutes
    - [4] Go to sleep earlier today
# Mark an item as done
$ donepy -d 1
Done: Write the project proposal for python
$ donepy -v
Hello Nimo, you currently have 4 tasks to do:
    - [1] Do OS Homework
    - [2] Think about why you ended up doing OS and PLT together and regret it for 10 minutes
    - [3] Go to sleep earlier today
```

## 4. Tests

The program will be mostly interactive. Therefore, the majority of the tests can be done manually by following a sequence of commands and see if we can reproduce the desired output. For import/export functionalities. We will generate a ground truth JSON file and `diff` it with the output of the program.