Project Report

Bicycle-Management System

Arir Allana, Aiza Kuzovenko, Mavzuna Azizova

University of Central Asia

2018

**Introduction**

This system was designed for managing bicycle borrowing in our campuses. Our program was coded using Python. There are three functions: register, borrow and return. For the interface we used tkinter module, so the program looks more appealing to the user. Bicycle database keeps the information about number of bicycles available. Student database will have ID information and history of borrowing bicycles. To measure time, we will use a loop and built-in Python date and time functions. To implement it, we are going to write a proposal to assistant manager of student life team (Shabana Manji) to approve our idea. If approved, we will install it to the security guard’s computers who gives permission to borrow bicycles. We hope that this system would be implemented in our campus.

**Background/Problem**

Khorog campus obtained 4 bicycles for entertainment and recreation. As we have 109 students and 12 hours borrowing time per day for each bicycle which will give us only 3 hours of bicycle use per week for each student. Our project aims to deliver a system for managing of the borrowing of these bicycles, because at present, bicycles are not issued as there is no management system. This program would make it easier for keeping track of allotted time and make students more responsible. More people would have access to bicycles, use of bicycles would be fairly distributed.

**Methods/Instruments**

The interface of the program was made using tkinter module *(from tkinter import* \*). The program window is divided into 2 columns and 13 rows. Program consists of window title, labels, fields and buttons. Window title was created using *window.title(),* feature of tkinter. Labels are assigned to all text and pictures that user sees in the program: title, names of fields, created using Label(). Example of the label we used: *Label(window, text="NEW USER:", font="verdana 18 bold") .grid(row=2, column=0, sticky=W) –* so, in brackets we give text and its parameters; also we set the location of the label using .grid() method: row, column and alignment (E/W/S/N go for right, left, up and down). Apart from labels the program has fields or entries from the user, created using Entry(). Example of the email field: (*email = Entry(window, width=35, bg="white", textvariable=StringVar());email.grid(row=11, column=0, sticky=W)).* Also, in brackets we give parameters like width and color, type of the input (string or integer), and use .grid to set the location. Finally, there are 4 buttons: Borrow, Return, Register and Exit. Example of Exit button: (*Button(window, text="EXIT", font="verdana 12 bold", fg="red", width=10, height=2, command=quit) .grid(row= 14, column=1, sticky=W)).* We also set up all the parameters such as text font, color; width, height and alignment of the button. Buttons also have events: “Button-1” – left button of the mouse click, “Button-2” – roller of the mouse, “Button - 3” – right button of the mouse or “exit” – general command (Events and Bindings, 2018). At the end of the code we have *window.mainloop(),* so program appears on the screen when user runs it and keeps running till the user exits it (How do I close a tkinter window?, 2018).

All buttons clicked are events which call a function. Our program has three functions: Register, Borrow and Return. Register function gets value the user inputs into 5 fieds (first column of the program). We used *.get* to take user’s input and assigned to variables (name, ID, password, email and graduation year). Function also has if condition to check if user left one of the fields empty (User will receive a message: 'Missing Information', 'Please enter all fields.'). Else the function will create a list of entries in users[] and sign a possession (ID + “a”), which means that student haven’t borrowed a bicycle yet. Then, it appends a list into “users.txt” and prints out a message of successful registration (Print output in GUI interface Tkinter Python, 2017). Finally, it removes all inputs in fields of the program. Second function is Borrow. Firstly, it takes the entries (ID and password) and current date/time from the event (user clicked a button) (How to get the current time in Python, 2018). Function has if-condition to check if this user exists (refers to created “user.txt” file), and if there are any bicycles available (The amount of bicycles is stored in “bikecount.txt” file should be more than 0) and if all conditions are true, it replaces “a” with “b” in the list “users.txt”– which means that now this user has borrowed a bike. After that, it deletes one bicycle in “bikecount.txt”, stores the time of the bike borrowed (convert time to the string and stores ID and time in “time.txt” file). Finally, it prints out a message of successful borrowing and deletes all entered information from the fields (How to clear the Entry widget after a button is pressed in Tkinter?, 2017). Our bike variable is *global*, because it is defined in another file (Geeksforgeeks, 2018). The last function is Return. Firstly, it checks if this user filled all fields, if this user exists and if this user borrowed a bicycle. If all conditions are true, it takes time (same mechanism as in Borrow function) and calculates the difference between borrow and return time (Python - How to calculate the time interval between two time strings, 2018). After that it compares resulted time with 2 hours, to know if the user went over allowed time. Finally, it returns a message (content of the message is different when the user returns the bicycle on time and when not) and adds one bicycle back to the “bikecount.txt” file.

**Conclusion:**

Our project was written in Python, so it can run on any platforms. It has three main functions and interface. Our program is managing bicycle borrowing – storing information about bikes, students and time of borrowing. As our program can be useful in campuses, we hope that it will be implemented.

# **References:**

*Events and Bindings*. (2018, 10 11). Retrieved from Effbot: http://effbot.org/tkinterbook/tkinter-events-and-bindings.htm

*Geeksforgeeks*. (2018, 11 10). Retrieved from Global and Local Variables in Python: https://www.geeksforgeeks.org/global-local-variables-python/

*How do I close a tkinter window?* (2018, 04 04). Retrieved from Stackoverflow: https://stackoverflow.com/questions/110923/how-do-i-close-a-tkinter-window

*How to clear the Entry widget after a button is pressed in Tkinter?* (2017, 05 23). Retrieved from Stack Overflow: https://stackoverflow.com/questions/2260235/how-to-clear-the-entry-widget-after-a-button-is-pressed-in-tkinter

*How to get the current time in Python*. (2018, 11 11). Retrieved from Stack Overflow: https://stackoverflow.com/questions/415511/how-to-get-the-current-time-in-python

*Print output in GUI interface Tkinter Python*. (2017, 03 06). Retrieved from Stack Overflow: https://stackoverflow.com/questions/42828416/print-output-in-gui-interface-tkinter-python

*Python - How to calculate the time interval between two time strings*. (2018, 03 12). Retrieved from Stackoverflow: https://stackoverflow.com/questions/3096953/how-to-calculate-the-time-interval-between-two-time-strings