FUNDAMENTALS IN SOFTWARE DEVELOPMENT

CT010-3-1-FSD

GROUP ASSIGNMENT

Intake Code: UC1F1604SE

Lecturer: SUMAIRA MUHAMMAD HAYAT KHAN

**PARKING MANAGEMENT SYSTEM**

**AHMED ABDULLA MAJUDHU** (TP041663)

**MD EASIR ARAFAT** (TP039768)

# Table of Contents

[Table of Contents 1](#_Toc457207559)

[Workload Matrix 3](#_Toc457207560)

[Design of the Program 4](#_Toc457207561)

[Main loop of the program 4](#_Toc457207562)

[Shelve database layout 6](#_Toc457207563)

[Login screen 7](#_Toc457207564)

[Main Menu 7](#_Toc457207565)

[Register new Student 8](#_Toc457207566)

[Cancel a student 9](#_Toc457207567)

[Update student details 10](#_Toc457207568)

[Display report 11](#_Toc457207569)

[Display student details 11](#_Toc457207570)

[Students by expiry date 12](#_Toc457207571)

[Display grid 12](#_Toc457207572)

[User menu 13](#_Toc457207573)

[Logout 14](#_Toc457207574)

[Exit 14](#_Toc457207575)

[Test Plan 15](#_Toc457207576)

[Sample Output 16](#_Toc457207577)

[Registering a new student: 16](#_Toc457207578)

[Update student 16](#_Toc457207579)

[Display report 16](#_Toc457207580)

[Display Student Details 17](#_Toc457207581)

[Students By Expiry Date 17](#_Toc457207582)

[Display grid 18](#_Toc457207583)

[List Registered Users: 18](#_Toc457207584)

[References 19](#_Toc457207585)

# Workload Matrix

|  |  |  |
| --- | --- | --- |
| **Name** | **Percentage** | **Signature** |
| Ahmed Abdulla Majudhu | 50% |  |
| Arafat | 50% |  |

# Design of the Program

## Main loop of the program

Start

Print menu

Read choice

If  
choice == 1

If  
choice == 2

If  
choice == 3

No

No

New\_student()

Cancel\_student()

Update\_student()

Yes

Yes

Yes

Read username, password

If  
username and password is   
correct

Yes

No

No

No

If  
choice == 4

Print\_report()

Yes

No

A

C

B

If  
choice == 5

If  
choice == 6

If  
choice == 7

If  
choice == 8

No

No

No

Student\_details()

Students\_by\_expiry

Print\_grid()

If  
choice == 9

If  
choice == 0

No

Stop

Yes

Yes

Yes

Yes

No

Yes

Print User Menu

Read choice

If   
choice == 1

If   
choice == 3

If   
choice == 2

If   
choice == 4

No

No

No

No

List\_users()

New\_user()

Del\_user()

Change\_pass()

No

A

C

B

## Shelve database layout

When the program is started it first loads the shelve database to a variable named ‘db’ and checks if the required dictionaries are defined in the database. If the values are not found it is initialized as empty dictionaries.

The ‘users’ dictionary contains the list of users. Username is set as the key and the value is a dictionary with keys fname, lname, pswd\_hash and salt. ‘fname’ is the first name of the user and ‘lname’ is the last name of the user. The user’s password is stored in an encrypted form by using the pbkdf2\_hmac function from the hashlib library. The hash type use is SHA512 and number of iterations is 100000. It may be changed by changing the variables hash\_rounds and hash\_name respectively. The database must be reset after changing these variables. The program will not work correctly if it does not match with the database values.

The ‘students’ dictionary contains the list of students currently registered for a parking space. The StudentID is the key and the value is a dictionary with the student’s first name, last name, contact number, email address, car number and the date of registration. The keys for each value are studentid, . For example, the car number of student with fname, lname, contact, email, carnumber and regdate, respectively.

The ‘parking’ dictionary contains the information of each parking space. It is organized as levels and spaces where all levels contain the same number of parking spaces. The number of levels and spaces may be set from the variables ‘parking\_levels’ and ‘parking\_spaces’, respectively. The database must be reset after changing any of these values. The program will not work correctly if they do not match with the database. The status, either assigned or available and the student ID of the student registered for that space is stored for each space. The keys used are ‘status’ and ‘studentid’ respectively

## Login screen

After the database is initialized and at least a single user exists the login function is called. The user is prompt for his username and password. The hash for the password entered is compared with the hash stored in the database for the specified username.  
If the two hashes match the menu function is called. If the password does not match the login screen is displayed again. The login screen is displayed until a correct username and password combination is entered. The system allows an unlimited number of login attempts and hence it is the user’s responsibility to use a password that cannot be guessed easily.

## Main Menu

The main menu displays all the functions that can be performed with the program. It will be displayed when the function menu is called. This function will be called at the end of all other functions, successful or not, except for logout and exit. Hence the program loops displaying the main menu. Each option is given a number. The number is to be entered by the user to perform that function. The functions with their number is as follows:

1. Register new Student  
2. Cancel a student  
3. Update student details  
4. Display report  
5. Display student details  
6. Students by expiry date  
7. Display grid  
8. User menu  
9. Logout  
0. Exit

## Register new Student

Read studentid, student\_details

Students[studentid] = student\_details

Save Students

Print Student registered successfully

This option calls the function new\_student. This function allows to register a new student in the parking system. The user is prompt to enter the details of the student, which are Student ID, first name, last name, contact number, email address and the car number of the student. The current date is used as the date of registration. It will check if the Student ID is already registered and will show and error if so. It will search for an available parking space and the student will be assigned a parking space if one is to be found. The status of the parking space will be set to assigned. If the all the parking spaces are already occupied, it will show an error and the student will not be registered. The student is successfully registered and saved in the database only if there is an available parking space.

## Cancel a student

This options calls the function cancel\_student. The user is prompt to enter the Student ID of the student to cancel. It will remove all the information of the student from the database and the parking space given to that particular student will be set as a free space.

Read studentid

Del Students[studentid]

Save Students

Print Student deleted successfully

## Update student details

This option calls the function update\_student. It allows to change the car number, contact number or the email address of the student. It also allows to renew the registration date. The registration will be set to the current date if renewed. It will show an error if the Student ID is not registered.

Read Choice

If  
choice == 1

If  
choice == 2

If  
choice == 3

Yes

Yes

Yes

No

No

No

Read carnumber

Read mobileno

Read email

Student[‘carnumber’] = carnumber

Student[‘mobileno] = mobileno

Student[‘email] = email

Save Student

## Display report

Print Students

The option calls the function print\_report. It shows the list of all currently registered students as a table. Student ID, car number, parking space ID, registration date and expiry date of all the users will be displayed.

## Display student details

This option calls the function student\_details. It shows all the information for a specific student. The user is prompt to enter the student ID. It will show the Student ID, first name, last name, contact number, email address, car number, parking space ID, registration date, expiry date and the number of days left for the student’s parking space to expire.

Read studentid

Student\_details = Students[studentid]

Print student\_details

## Students by expiry date

This option calls the function students\_by\_expiry. It displays all the students who have less than a certain amount of days for their parking space to expire. The user is prompt to enter the number of days. It displays the student ID, expiry date, number of days left, parking space id and the name of the student.

Read days

Student\_details = next student

If  
student\_details != null

If

student\_details[exipry] – today < days

Print stundent\_details

Yes

Yes

No

No

## Display grid

This option calls the function print\_grid. It displays all the parking spaces as a grid with the parking space ID and the student ID if a student is registered on that space. The spaces in each row may be adjusted by changing the spaces\_per\_row variable in the function. It uses the box drawing characters from Unicode to draw the grid.

## User menu

Read choice

If  
choice == 1

If  
choice == 2

If  
choice == 3

Print Users

Read username

Users[username] = [firstname, lastname, password]

Read firstname, lastname username, password

Del users[username]

Yes

Yes

Yes

No

No

No

This option displays allows to list all the registered users, create a new user, delete an existing user or change the password of an existing user. Any user may delete a user. To change the password the old password must be entered correctly. The functions list\_users, new\_user, del\_user and change\_pass are called respectively.

## Logout

This option calls the function login. The user will be logged out and instead of the main menu it will display the login screen.

## Exit

This option calls the save\_and\_exit function. It closed the database and exits the program.

# Test Plan

|  |  |  |
| --- | --- | --- |
| **Function** | **Expected output** | **Actual Output** |
| Register new student | Ask for new student details, new student saved in the databse | As expected |
| Update student | Ask for student ID and what to update, student details saved with new information | As expected |
| Delete student | Ask for student id, remove all student details from the databse | As expected |
| Display report | List all student details | As expected |
| Student details | Ask for student ID, print all details of student | As expected |
| Students by expiry | Ask for how many days, print details of all students who have less than that amount of days to expire their registration | As expected |
| Display grid | Prints parking space id followed by student id of the whole parking system | As expected |
| List users | Prints full name and username of registered users | As expected |
| Create new user | Ask for name, username and password, then create a new user | As expected |
| Delete existing user | Ask for username | As expected |
| Change password | Ask for old password and new password, change password only if old password matches the stored password | As expected |
| Logout | Log the user out of the system and display the login screen | As expected |
| Exit | Save all information and close the program | As expected |

# Sample Output

## Registering a new student:

The user is prompt to enter the details. After succeful registration the parking space id will be printed as on the last line.

Enter studuent details

Enter StudentID: TP012345

Enter First Name: Oliver

Enter Last Name: Jack

Enter mobile number: 0123456789

Enter e-mail address: oliverjack@mail.com

Enter car number: OJ12345

New student TP012345 registered successfully with parking space L101

## Update student

The user is prompt ot enter the student ID. After successful update the new information will be printed as on the last line.

Enter studentid to update: TP012345

Oliver Jack (TP012345)

1. Update car number

2. Update contact number

3. Update email address

4. Update registration date to today

1

Enter new car number: MN6789

Car number of student TP012345 changed to MN6789

## Display report

Few information is printed with display report.

StudentID | Car No. | Parking | Registerd | Expiry

==========|=========|=========|===========|=========

TP012345 | OJ12345 | L101 |2016-02-12 |2016-06-11

TP011111 | AA12342 | L102 |2016-03-18 |2016-07-16

TP022222 | BC87657 | L205 |2016-04-02 |2016-07-31

TP033333 | JK34913 | L213 |2016-04-21 |2016-08-19

TP044444 | KF09487 | L306 |2016-01-23 |2016-05-22

TP055555 | KJG9824 | L315 |2016-05-29 |2016-09-26

## Display Student Details

All information of the student is printed

Enter studentid: TP012345

Student ID: TP012345

First Name: Oliver

Last Name: Jack

Contact No.: 0123456789

E-mail: olverjack@mail.com

Car Number: OJ12345

Parking: L101

Registerd on: 2016-02-12

Expire on: 2016-06-11

Days left: 23 days

## Students By Expiry Date

User may enter any number of days to filter the output. A number of 0 may be entered to displayed students who have their parking already expired.

Students who have less than how many days? Enter a number: 30

StudentID | Expiry | Days | SpaceID | Student Name

========================================================

TP011111 | 2016-03-18 | 13 | L102 | Emma Noah

TP022222 | 2016-03-25 | 19 | L205 | Liam Benjamin

TP044444 | 2016-03-30 | 24 | L306 | Jacob William

## Display grid

If a space is assigned, the student id is printed. If the space is available it is left blank.

Level: 1

┌────────╥────────╥────────╥────────╥────────┐

│ L101 ║ L102 ║ L103 ║ L104 ║ L105 │

│TP012345║TP011111║ ║ ║ │

╞════════╬════════╬════════╬════════╬════════╡

│ L106 ║ L107 ║ L108 ║ L109 ║ L110 │

│ ║ ║ ║ ║ │

╞════════╬════════╬════════╬════════╬════════╡

│ L111 ║ L112 ║ L113 ║ L114 ║ L115 │

│ ║ ║ ║ ║ │

└────────╨────────╨────────╨────────╨────────┘

Level: 2

┌────────╥────────╥────────╥────────╥────────┐

│ L201 ║ L202 ║ L203 ║ L204 ║ L205 │

│ ║ ║ ║ ║TP022222│

╞════════╬════════╬════════╬════════╬════════╡

│ L206 ║ L207 ║ L208 ║ L209 ║ L210 │

│ ║ ║ ║ ║ │

╞════════╬════════╬════════╬════════╬════════╡

│ L211 ║ L212 ║ L213 ║ L214 ║ L215 │

│ ║ ║TP033333║ ║ │

└────────╨────────╨────────╨────────╨────────┘

Level: 3

┌────────╥────────╥────────╥────────╥────────┐

│ L301 ║ L302 ║ L303 ║ L304 ║ L305 │

│ ║ ║ ║ ║ │

╞════════╬════════╬════════╬════════╬════════╡

│ L306 ║ L307 ║ L308 ║ L309 ║ L310 │

│TP044444║ ║ ║ ║ │

╞════════╬════════╬════════╬════════╬════════╡

│ L311 ║ L312 ║ L313 ║ L314 ║ L315 │

│ ║ ║ ║ ║TP055555│

└────────╨────────╨────────╨────────╨────────┘

## List Registered Users:

The full name and the username of each user is printed.

Darrell Daluz (PKO1234)

Harris Goris (PKO1111)

Hassie Cloutier (PKO2222)

Orlando Melara (PKO3333)

Dora Eury (PKO4444)

# **References**

Pyhton Software Foundation, 2016. *12.3. shelve — Python object persistence.* [Online]   
Available at: https://docs.python.org/3/library/shelve.html  
[Accessed 21 July 2016].

Python Software Foundation, 2016. *15.1.2. Key derivation.* [Online]   
Available at: https://docs.python.org/3/library/hashlib.html#key-derivation  
[Accessed 21 July 2016].

Python Software Foundation, 2016. *16.9. getpass — Portable password input¶.* [Online]   
Available at: https://docs.python.org/3/library/getpass.html  
[Accessed 21 July 2016].

Python Software Foundation, 2016. *4.10. Mapping Types — dict.* [Online]   
Available at: https://docs.python.org/3/library/stdtypes.html#dict  
[Accessed 20 July 2016].