**CMPUT 291 Project 1 Design Report**

Team Members: Tian Qi Xiao,

Weijie Sun,

Qingdai Du

This database application system for an auto registration system, which have access to the data in the Oracle Server. By calling different functions in this program, user who is an officer can register a new vehicle and added it to the system, complete an auto transaction by entering required information and issue a driver licence with given information and photo; user who is a police officer can issue a traffic ticket with violation; and also, user can use three build-in search engine to perform three different searches. The program imported sys, cx\_Oracle, getpass, and randint(from random) to extend the function.

*main()*

It will connect to the oracle and ask user to choose a number, then the program will call different function to meet user’s desired request. To accomplish this goal, we first designed to ask user for Username and Password, if the Username and Password does not match, the program will print error message and return to the beginning of the loop to let user enter the Username and Password again; if we get the proper Username and Password, the program will successfully connect to the database and change the connection status to True to exit the loop. Then, the application will display welcome message and a list of numbers which have a corresponding function. If the user enter “1” as the input, the application will call function *New\_Vehicle(curs, connection)* to operate; if the user enter “2” as the input, the application will call function *Auto\_Transaction(curs, connection)* to operate; if the user enter “3” as the input, the application will call function *Driver\_Licence\_registration(curs, connection)* to operate; if the user enter “4” as the input, the application will call function *Violation\_Record(curs, connection)* to operate; if the user enter “5” as the input, the application will call function *Search\_Engine(curs, connection)* to operate. If the user wants to exit the system, they can simply type in “exit”, the application will be disconnected to Oracle database and be terminated. If the user did not type in a valid string, the system will print a warning message and ask user to try again.

*People\_Information(curs, connection, sin)*

It will takes all the detailed information of a person to register to the database. To accomplish this goal, the function first takes in three parameter, curs, connection and sin number, which allows the program to access the database. Besides the sin number, the function needs user to input the name, height (an integer between 0 and 1000), weight (an integer between 0 and 1000), eye color (a string variable), hair color (a string variable), addr (a string variable), gender (a string variable; check if the gender is a valid input), birthday (the date need to be exact 11 digit with desired format). After got all the information, the function will put them together as a new personal information profile and save in the database.

*New\_Vehicle(curs, connection)*

It will takes all the detailed information about a vehicle and personal information of its new owners, if it is not in the database. To accomplish this goal, the function first takes in two parameter, curs and connection, which allows the program to access the database. We first designed it to ask user to enter the valid serial number and check if it is not in the database (make sure it is a new vehicle). Then, we want user to enter the maker and model name, year (check enter a valid four digit year number), color and type id (make sure the type id is in the database). If any part of the input is not valid, the function will print error message and ask user to enter again. At the end, all the input information will be store together for a new vehicle. After, the program will record owner’s sin number and check if it is in the database, if not, it will call the function *People\_Information(curs, connection, sin)* to add this new person’s information. This function also will ask the user whether the person is the primary information. If any part of the input is not valid, the function will print error message and ask user to enter again. At the end, after all necessary information entered successfully, the program will back to the *main()* function and shows the main menu.

*Auto\_Transaction(curs, connection)*

It will takes all the information including the detail of the seller, the buyer, the transaction date, and the price to complete an auto transaction. To accomplish this goal, the function first takes in two parameter, curs and connection, which allows the program to access the database. We first designed it to get sin, vehicle id and transaction\_id from the database for further use. Then, it will ask user to input seller’s id and check if it is in the database, if not, it will call *People\_Information(curs, connection, sin)* to add this new person’s information. Same as buyer’s id, but the program will also check if seller and buyer are the same, if yes, then print error message and ask user to input again. The user need to provide vehicle id as well. If the vehicle is not in the system yet, it will call *New\_Vehicle(curs, connection)* function to register first and then continue the process by asking for date (which need to check if it is a valid number or in desired format) and price (which also need to check if it is a valid number) input. After recording all the information, the program will provide user an option of whether enter a transaction id (need to check if it is a new transaction id number) or let the system create one automatically (and randomly). Finally the system will ask if there’s more than one owner, if yes, it will add another owner to the transaction as well; if no or if finished adding another owner, the program will back to the *main()* function and shows the main menu.

*Driver\_License\_registration(curs, connection)*

It will takes all the information needed, including the detail of the personal information, a picture of the driver, issuing date, expiring date and the driving class, to issue a drive licence. To accomplish this goal, the function first takes in two parameter, curs and connection, which allows the program to access the database. We first designed it to get sin number and transaction\_id from the database for further use. The program will ask a sin number input from user and make sure it does not in the database, then use the sin to create a new personal information profile by calling *People\_Information(curs, connection, sin)*. Then the function will ask user to input a licence number (need to check if it is a new licence number), a drive class (need to check if the input is valid), a issuing date (the date need to be exact 11 digit with desired format), expiring date (the date need to be exact 11 digit with desired format) and a photo image of the driver. At the end, after all necessary information entered successfully, the program will back to the *main()* function and shows the main menu.

*Violation\_Record(curs, connection)*

It will takes all the information needed, including the detail information of the police officer, violation people, violation vehicle and ticket, to issue a traffic ticket. To accomplish this goal, the function first takes in two parameter, curs and connection, which allows the program to access the database. We first designed it to get the sin number, vehicle serial number, ticket number, ticket type, to issue a traffic ticket from the database for further use. The program will provide user an option of whether enter a ticket number (need to check if it is a new ticket number) or let the system create one automatically (and randomly). Violation number, as responding to people’s sin number, is required. If the input sin is not in the database, the program will ask user to register first by calling *People\_Information(curs, connection, sin)*. Then the user will enter the police officer’s number (need to check if the number is valid), choose a violation ticket type (need to check if the type is valid), violation date (the date need to be exact 11 digit with desired format), violation place (a string variable which cannot be too long), and description (a string variable which cannot be too long). At the end, after all necessary information entered successfully, the program will back to the *main()* function and shows the main menu.

*Search\_Engine(curs, connection)*

It will allows user to choose one of three build-in search engine to complete desired search action. This function takes in two parameter, curs and connection, which allows the program to access the database. The program will display four option: if the user input “1”, the program will call *search1(curs, connection)* function to list all basic information of the driver; if the user input “2”, the program will call *search2(curs, connection)* function to list all violation information of the driver; if the user input “3”, the program will call *search3(curs, connection)* function to display the vehicle history; if the user input “Exit”, the program will exit from the function and return to the *main()* function and shows the main menu. If any part of the input is not valid, the program will ask user to enter again.

*search1(curs, connection)*

It will list the name, licence number, address, birthday, driving class, driving condition and the expiring date of a driver by entering either a licence number or a given name. This function will also display all the result if a duplicate name is given. This function takes in two parameter, curs and connection, which allows the program to access the database. First, it will gets person’s name and driver licence information from the database for further use. The function will provide the user an option to choose either enter the name of the driver or enter the driver licence number. If the user choose to enter a name, the program will first check whether the name exist or not, if not, ask the user to re-enter it; if yes, pull out the corresponding personal information. If the user choose to enter a licence number, the program will first check whether the licence number exist or not, if not, ask the user to re-enter it; if yes, pull out the corresponding personal information. At the end the program will back to the *main()* function and shows the main menu.

*search2(curs, connection)*

It will list all the violation record of the driver by entering either a licence number or a sin number. This function takes in two parameter, curs and connection, which allows the program to access the database. First, it will gets person’s sin number and driver licence information from the database for further use. The function will provide the user an option to choose either enter the sin of the driver or enter the driver licence number. If the user choose to enter a sin number, the program will first check whether the person exist or not, if not, ask the user to re-enter it; if yes, pull out the corresponding personal information. If the user choose to enter a licence number, the program will first check whether the licence number exist or not, if not, ask the user to re-enter it; if yes, pull out the corresponding personal information. At the end the program will back to the *main()* function and shows the main menu.

*search3(curs, connection)*

It will print out the vehicle\_history, including the number of times that a vehicle has been changed hand, the average price, and the number of violations it has been involved by entering the vehicle's serial number. This function takes in two parameter, curs and connection, which allows the program to access the database. First, it will gets vehicle serial number from the database for further use. This function will ask user for the serial number of a certain vehicle and check if the input serial number is in the database, if not, the function will ask user to re-enter it; if yes, the program will continue. Then it will drop and create a new view, vehicle\_history, to record all desired information, including the number of times that a vehicle has been changed hand, the average price, and the number of violations it has been involved. Then it will group them all together and displays on the screen for the user. At the end the program will back to the *main()* function and shows the main menu.