School of Computing and Information Technology, University of Technology Jamaica

CIT2004: Object Oriented Programming

Chevannese Ellis – 2301109

Kehvoi Thompson– 2304224

Janae Bernard- 2209025

Karlicia Sutherland – 2302691

Kay-Ann Green – 2110490

Occurrence: UN1

Rorron Clarke

Due Date: November 30, 2024

**Table of Contents**

[Group Report 3](#_Toc184245221)

[Object Oriented Analysis 6](#_Toc184245222)

[Object Oriented Design 19](#_Toc184245223)

[User Manual 21](#_Toc184245224)

# **Group Report**

|  |  |  |
| --- | --- | --- |
| Group Member | Activity | Contribution |
| Chevannese Ellis | Object Oriented Analysis and Design | Identified classes, Main, Name, Date, Address and Ticket. Created OOA table and UML Diagram based on all members’ input. |
|  | Implementation of Add Ticket, View current ticketing information, Check status of a driver, View all outstanding tickets (sorted by parish) Methods. | Responsible for implementing all functionalities associated with selecting JCF Officer. |
|  | Implementation of Abstract Class | Implemented User as abstract class. The abstract method is called Info. This allows its child classes to inherit the method Info. |
| **Kehvoi Thompson** | Object Oriented Analysis and Design | Identified class abstract User. |
|  | Implementation of user-friendly interface of TIOCS | Created while loop to allow user(s) to interact with all options of the TIOCS. |
|  | User Manual | Implemented Manual for to demonstrate executable program. |
|  | Implementation of Driver Methods | Responsible for Implementing OptionDriver methods 4-5. |
| **Janae Bernard** | Object Oriented Analysis and Design | Identified methods StatusCheck. Additionally, identified class JCFOfficer. |
|  | Implementation of user-friendly interface of RPLS | Created while loop to allow user(s) to interact with all options of the RPLS. |
|  | Implementation of ViewApplicantPersonalInfo and ViewAllApplicantPersonalInfo | These methods allow a Processing Officer to view the personal information of applicant(s). |
| **Karlicia Sutherland** | Object Oriented Analysis and Design | Identified methods Create, Update, Delete, Reject and View all for Main class. |
|  | Implementation of Create, Update, Delete,View, View All Methods | Responsible for all options related to processing officer |
| **Kay-Ann Green** | Implementation of checkPastTickets, makeOnlinePayment, checkPastDueTickets  Methods | Responsible for Implementing OptionDriver methods 1-3 |

# **Object Oriented Analysis**

Based on the concerns about the persistently high rates of **motor vehicle** traffic deaths on Jamaican roads according to a **National Road Safety Council (NRSC)** review meeting in Kingston on January 11, 2024, PAHO, (2024). Another concern for this solution is the consistently high number of applications received yearly for Public Passenger Vehicle (PPV) licenses and the high rate of unpaid tickets by **PPV operators**. Your group is asked to create software solutions that will automate the *approval of applications* for **PPV licenses** based on specific criteria. Create another solution that will be used by the **Jamaica Constabulary Force (JCF)** to *issue tickets* to **road traffic offenders**, *record road accidents* and *store police records*.

* All the nouns in the text above are underlined, italicized, and bold however, National Road Safety Council is not a component of the system, so it is rejected.
* The class identified is JCF and its methods are issue tickets and store police records
* The next class identified was Driver as it is a suitable name to describe PPV operators and offenders.
* The other class identified is Ticket and the attribute is unpaid tickets.
* Another class Main was identified and will contain the method Approval of Application.

The solution will link the two systems using OOP principles that were taught in the module along with your research to implement a robust software solution. The new systems will allow the **JCF** to know the verified **address** of the **driver/owner**, and the location of the **vehicle** through a mandatory tracking system that all **PPV owners** must install and give access to the **Red Plate Licensing System (RPLS)**. When implemented it is the hope that the robust solution should reduce the high traffic-related deaths by keeping serial **offenders**, **criminals** and other unapproved **drivers** from obtaining an opportunity to operate a **Public Passenger Vehicle (PPV).** The **system** should be able *to produce as a report* the name of the **current** and **last driver**, the current and last owner of the vehicle, and the *total tickets outstanding* for the **applicant** and **current driver**.

* All the nouns in the text above are underlined, italicized, and bold however, Vehicle is not a component of the system, so it is rejected.
* The class Driver would also contain attributes: name, address. Applicant and criminal are as classes are rejected from system because there is already a class known as Driver.
* The class Main would contain the methods: produce as a report and total tickets outstanding. The attribute identified is currentDriver.
* The next class identified is Address.
* The other class identified is Name.

System 1: Red Plate Licensing System (RPLS)

The requirements of the system are as follows:

1. The **processing officer** can check the *status of an applicant*, through the **Ticketing Issuing and Offender Checking System (TIOCS)** and can do the following:

a. *Create an application*

b. *Update an application.*

c. *Delete an application.*

d. *Reject an application.*

2. The **processing officer** using the **applicant**’s TRN can see if an applicant has any outstanding tickets.

3. The **processing officer** can indicate that outstanding tickets must be paid before the **application** is processed.

* All the nouns in the text above are underlined, italicized, and bold, however Processing Officer is not a component of the system, so it is rejected.
* Class Driver contains attribute trn.
* Main class has additional methods which are the following: Create an application, Update an application, Delete an application, Reject an application, and Status of an Applicant.

When an **owner** applies for a *red plate license* the **system** will ask the following questions to ensure that the **applicant** is qualified to receive a *red plate permit:*

1. Did the **driver** cause any *accident(s)* within the last two years?
2. Does the **driver** have a negative *police record*?
3. Does the **driver** have any outstanding tickets?

Once any of the 3 conditions above is “Yes”, the applicant will have to change the proposed **driver** within 5 to 10 business days, otherwise, the application will be denied. However, if none of the conditions are “Yes” then, the permit will be issued to the **applicant**. The **applicant** should be provided with a *report of the status* of their **application** regardless of the outcome.

The applicant’s *Tax Registration Number (TRN)* should be used as the search key when searching for the applicant’s information. The output should include the following:

* The application's *full name, date of birth, TRN, current address, email address and contact number.*
* the *number of tickets outstanding*, the *amount for each ticket* and the *overall amount owing.*
* the reason the *police record* check was denied or approved.
* the reason the **driver** was denied and the following statement, “Application Denied based on the disqualification of the proposed driver, you have 5 to 10 business days to provide another driver”. The **application** is therefore denied pending the change of driver and/or the full payment of the **outstanding fees**.
* All the nouns in the text above are underlined, italicized, and bold, however red plate license and police record were rejected from the system due to having class Driver and Ticket carrying the same attributes.
* The class Driver also has attributes date of birth, email address, contact number.
* The class Ticket also has attributes number of tickets outstanding, the amount for each ticket, overall amount owing.

**Drivers** should be able to log on to the system using their *TRN* and be able to do the following:

1 Check for all their past tickets, which will show output in ascending alphabetical order based on all their issued date. The report should show the following information:

a. Ticket Issue Date

b. Payment Due Date

c. Ticket Number

d. Offence Code

e. Offence Description

f. Fine Amount

g. Ticket Status Description (i.e. paid, unpaid, warrant outstanding)

h. Court Date

i. Court Location

j. Total unpaid tickets

k. Total fine amount

2 *Make online payments* for tickets that are issued but not passed due.

3 *Check for past-due tickets*. Tickets which have passed the 21 days for payment will reflect a court location, date and time that the offender should appear in court to answer his/her charges.

4 *View ticket(s) payments that have not passed due based on their TRN*.

5 *Check where there is a warrant issued* for their *arrest* for not appearing in *court*, and it should show which *police station* they should turn themselves in.

* Class Ticket would contain additional attributes such as ticket issue date, payment due date, ticket number, offense code, offense description, fine amount, ticket status description, court date, court location, total unpaid tickets and total fine amount. Furthermore, Class Ticket would have methods: Make online payment, Check for past due tickets, and check where there is a warrant issued for arrest.
* Class JCF would contain attribute police station.

System 2: Ticketing Issuing and Offender Checking System (TIOCS)

This system should have the following basic details as defined below however, additional options can be added based on the group’s creativity. The system should be designed to be used by **the Jamaica Constabulary Force (JCF) Officers** to manage drivers on the roadways of Jamaica by using the law to issue cautions. **JCF Officers** can use the system to both add information for an offender and/or be able to check a driver's information in **RPLTS** for any outstanding tickets, or warrants. The **JCF officer** can *Add a new ticket*, check for any outstanding tickets, and *delete a ticket* that was added to the system based on their discretion. The Offender’s Driver’s License is used to do all processing. A driver’s license number is the same number that is used for TRN.

1. *Add*: This allows the **JCF officer** to add a new ticket for an **offender**. An offender might be an existing or a first-time offender.

i. The new ticket will have the following data:

a) Ticket number

b) Ticket Issue Date

c) Ticket Offence code

d) Ticket Offence description

e) Fine amount for the Ticket (The Fine should be automatically assigned based on the fine class)

f) Vehicle License Plate Number

g) Drivers TRN

h) Driver’s Full Name

i) Driver's Date of Birth

j) Driver's Address

k) Driver's Contact Number

l) JCF Officer’s Badge Number

m) JCF Officer’s Full Name

n) JCF Officer’s Assigned police station

* Ticket Class has the following methods: Add Ticket, Delete Ticket. Its attributes are license and warrant
* JCF Officer has the attributes Badge Number, Full Name, and Assigned police station

ii. A JCF Officer should be able to *view the current ticketing information that was added and verified with the driver before the final submission.*

iii. *Check: This allows a JCF Officer to check the status of a driver in the system.*

a) To verify if the driver has any unpaid tickets that have passed the 21-day stipulation.

b) Can view all offenders who have outstanding tickets pending including their address and TRN.

c) Should be able to *view all the outstanding tickets in a specific parish.*

iv. *View All:* This allows a **JCF Officer** to *view all outstanding tickets sorted by parish*.

a) This allows the JCF Officer to *view all the outstanding tickets in a specific parish*. The system should be able *to show who has the most outstanding tickets* and *produce a report showing each offender*’s full name, address, TRN, contact information, gender, and ticketing information (which includes each ticket number, date issued, whether it is paid or unpaid, and whether a warrant has been issued for an arrest).

b) The JCF Officer will have the option to update the current information that they are viewing. The information can be updated because information is missing or incomplete. The information can be both new information or old information that may include, the result of a police report, and any convictions etc.

* Class Driver has additional attribute such as gender.
* Class Ticket has additional methods such as ViewAllOutstandingParishTicket, and ViewCurrentTicketingInfo.

The Driver/Offender can use the RPLTS to check for the following:

1. any outstanding tickets,
2. any outstanding warrants.
3. due date for payment of outstanding tickets.
4. *view passed paid tickets.*
5. *pay for tickets that are not passed due online.*

3. The *online ticket payment system* will accept the following information and process the ticket:

i. Driver TRN

ii. Ticket Number

iii. Ticket amount

4. When the exit option is selected the application should close.

5. All committed changes made during the execution of the program should be stored and used to update the relevant files when the application terminates.

* The classes, methods and attributes were already identified previously

Program Requirements:

1.Perform an Object-Oriented Analysis (OOA) on both proposed solutions, that is the RPLTS and the TIOCS described above. The OOAs should clearly show the steps used to identify potential classes, attributes and methods. Based on the OOAs, create Object-Oriented Designs (OOD) using the Unified Modelling Language (UML) for the classes that were identified in the OOAs. The OOD should show appropriate UML diagrams for all classes and the class relationship diagrams, showing all existing relationships between the classes.

2. Using C++ or Java, implement both systems and allow them to be able to interact with each other as your group project.

**Users** are placed into the following categories:

a. **Processing Officer**

b. **Users**

i. **JCF Officers**

ii. **Drivers**

3. The Processing Officer (PO) maintain the overall system to include the Red Plate Licensing and Ticketing System (RPLTS). The PO is responsible for adding the PPV

* Class User was identified as the base class, and its child classes are JCF Officer and Driver
* Processing Officer, RPLTS, TIOCS are components of the Main Class
  1. PO can do the following functions: Add: This allows the PO to add a new driver to the system.
  2. Update: This allows the PO to update an existing driver’s information.
  3. View: This allows the PO to view a driver's driving record.
  4. View All: This allows the PO to view all drivers with a PPV license badge per parish with each driver badge's issued date and expiration date.
  5. Delete: This allows the PO to delete a Driver’s record.
* PPVLicense was identified as a class with attributes badgeNum, badgeIssueDate, badgeDueDate, and badgeParish

# Object Oriented Design

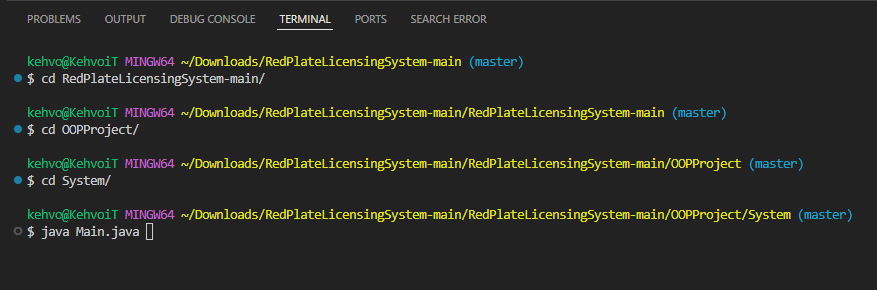
A diagram of a family tree

Description automatically generated

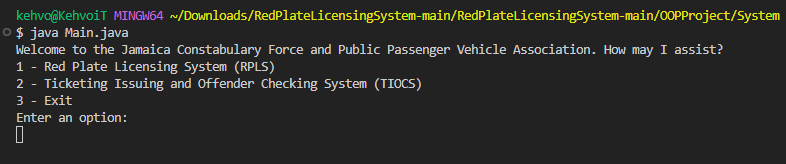
# User Manual

This program uses the common line interface.

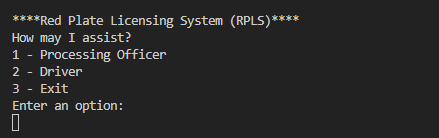
Start by typing Java Main.java in the terminal. This starts the program

Once the program is launched, the main menu should appear with the two systems that were to be implemented.

Use the number keys to select an option.



Using RPLS

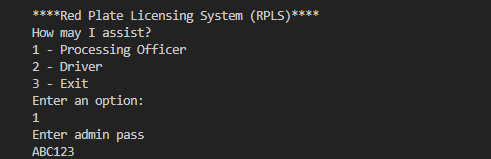


Select the option

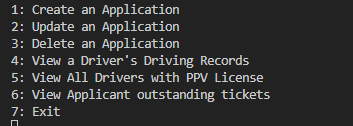
Processing Officer

Enter 1 to select Processing Officer

The passcode to access the Processing Officer Account is **ABC123**

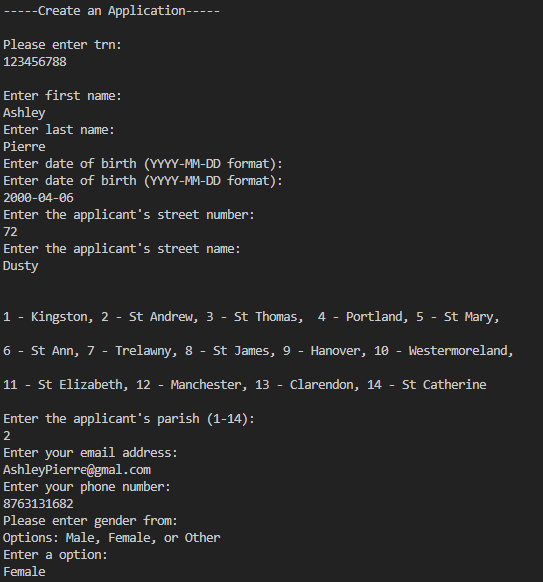


Once completed, a sub-menu only available for the Processing Officer Account will appear.

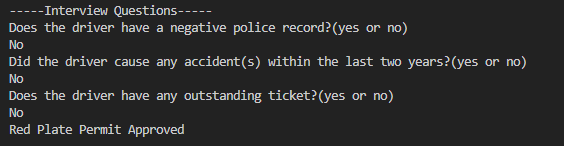


Creating Application Process

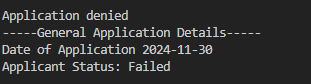
To create an Application select Create an application by pressing the corresponding number on the screen



Proceed to answer questions about the individual

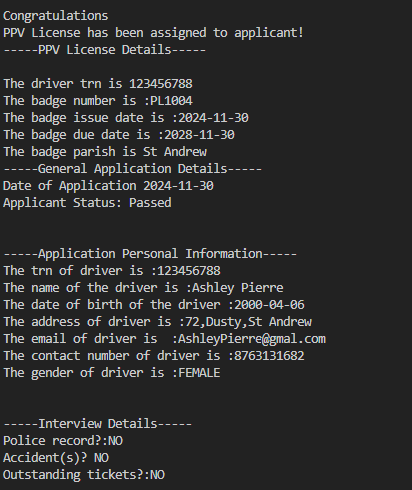


If all requirements are met the user’s request will be approved. However if all the requirements are not met the application will be denied



The information will then be stored for future use.

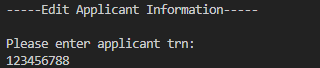
This is how to create an applicant successfully.



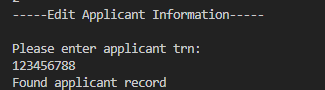
Updating Application Process

Let’s say there was a mistake in the applicant’s last name and we want to change it from Perre to Rodriguez. We would simply go to the update section To update an application, return to the Processing Officer menu and select the update option.

Once selected enter the applicant’s TRN

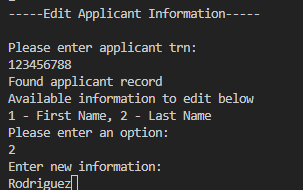


If the applicant’s information is found it will display a message saying a record has been found.

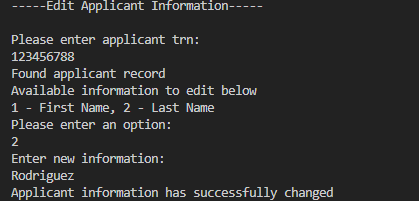


If there is a no record then -

Enter the select and enter the option you’d like to replace, in this case it would be the last name, then enter the information.



The information is then updated



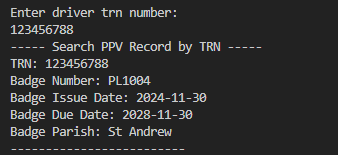
Viewing a Driver’s Driving Record

To view the record that was created.

Select the option from the menu .

Enter the TRN to search for the Driver’s record.

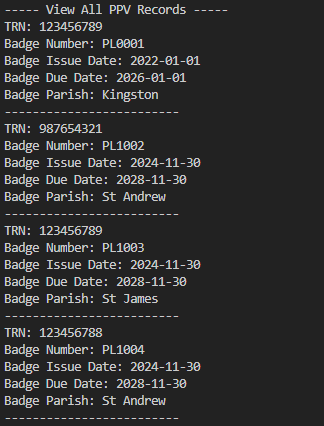
This TRN was just created. Here is the corresponding information.



View All Drivers with PPV License

To view all existing records that are stored in the system, select the option from the menu.

This will then print out all available records.

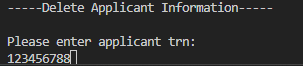


Deleting Application Process

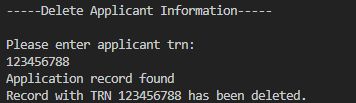
To delete an applicant’s record, select the option in the menu.

You will then be prompted to enter the TRN of the applicant you want to delete from the system.

For this example let’s use the applicant that was created above.



Once the TRN is entered the applicant’s information will be deleted.



Source: Trust me Bro.

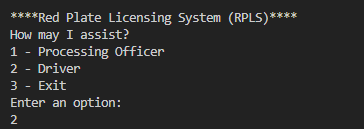
View Applicant outstanding tickets

To check if an applicant has an outstanding ticket, select the corresponding option from the menu. Once again you will be prompted to enter the TRN. The information will then be printed ont he screen.



Driver

Press 2 to select the Driver option in the RPLS menu.

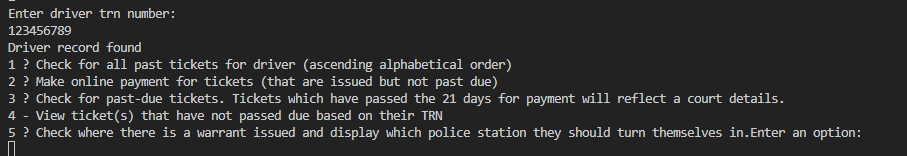


You will be prompted to enter the Driver’s TRN

If the record has been deleted or does not exist, you will receive a Driver record not found message.

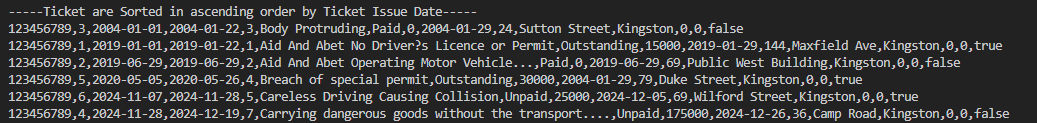


If the Driver is registered and is on the system the driver sub menu will appear.



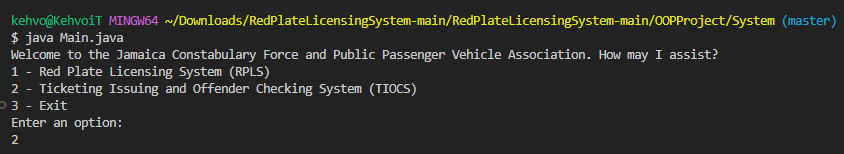
Checking All Past Tickets For Driver

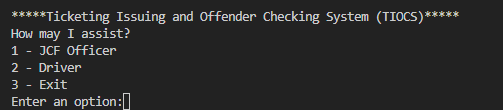
To check for all previous tickets for the driver select the option by pressing 1. The information corresponding to that driver will be printed on the screen.



**Using TIOCS**

Use the number keys to select an option.

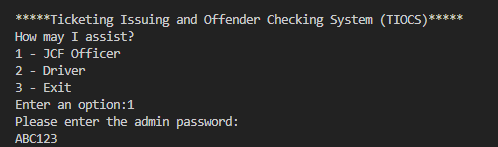




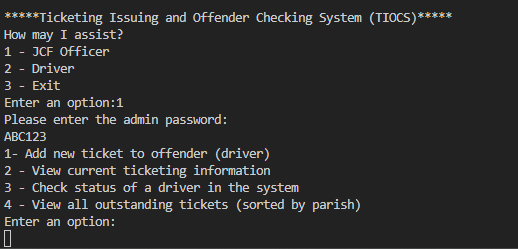
**JCF Officer**

Use the number keys to select the JCF Officer.

The passcode for JCF Officer is **ABC123**



This sub menu has several different options

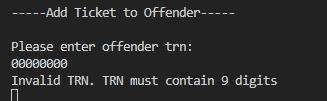


**Adding a ticket to Offender**

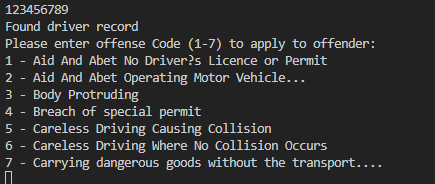
Start by Selecting the Add new ticket to offender option.

You will then be prompted to enter Offender TRN.

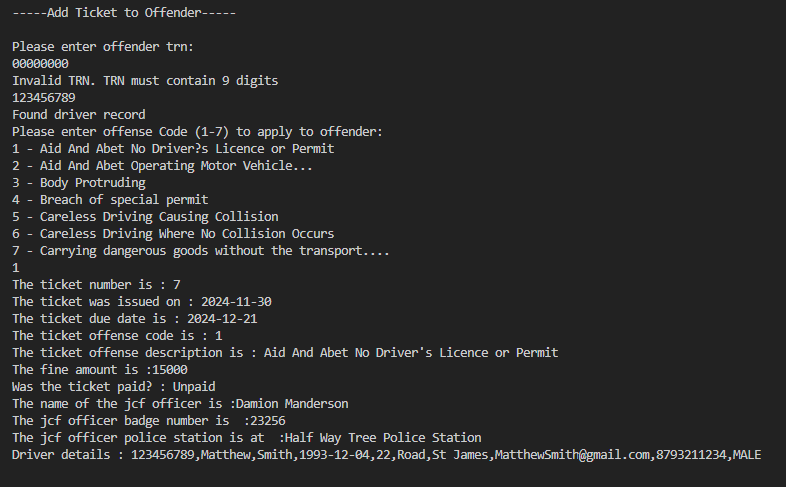
Ensure that the information being added is correct. If the request will be rejected.



Once a valid TRN is provided, you will then be able to select the type of offense based on the offense code.



Once the offense is selected the ticket will be written automatically.



Viewing Current Ticket Information

To view the latest ticket, select the view current ticketing information

