

**Analysis and Design of Dominique’s Driving School System**

Lab 1

DMIT 2028 – Systems Analysis and Design II

Section: E01

Team Placeholder: Sheldon J Costello-Johnson, Andrew Ta, John Christopher C Bana, Jianhe Wang, Guillermo E MORONTA FERNANDEZ

Date: October 27, 2021

Instructor: James Thompson

Table of Contents

[Introduction 1](#_Toc86164448)

[Purpose 1](#_Toc86164449)

[Use Case Diagram 2](#_Toc86164450)

[Use Case Narratives 4](#_Toc86164451)

[Entity Relationship Diagram 19](#_Toc86164452)

[Business Rules 20](#_Toc86164453)

[Screens and Reports 21](#_Toc86164454)

[Class List Prototype 21](#_Toc86164455)

[Class List Prototype Writeup 22](#_Toc86164456)

[Narrative 22](#_Toc86164457)

[Vehicle Reservation Screen Prototype 25](#_Toc86164458)

[Vehicle Reservation Screen Prototype Write Up 26](#_Toc86164459)

[Narrative 26](#_Toc86164460)

[Vehicle Reservation Report 28](#_Toc86164461)

[Vehicle Reservation Report Narrative 29](#_Toc86164462)

[Summary and Recommendations 30](#_Toc86164463)

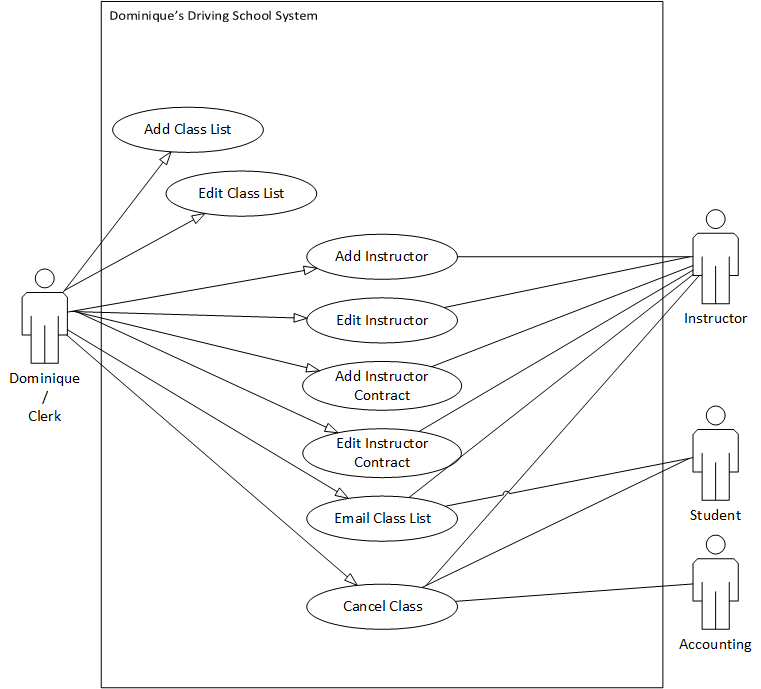
# Introduction

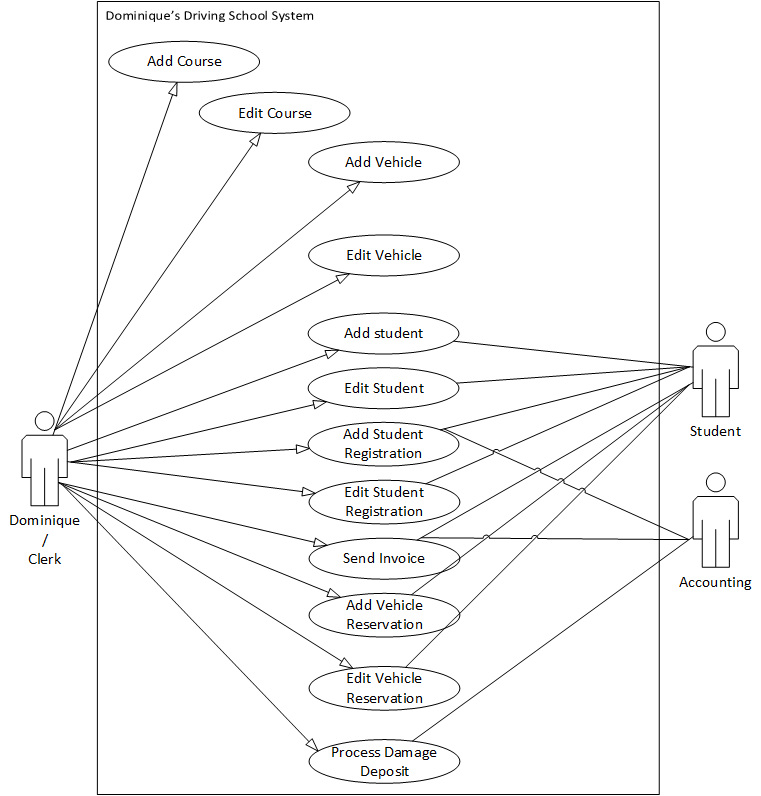
Dominique’s Driving School is a small local business that offers cars, trucks, and motorcycle classes to students. They plan on expanding their offerings next year by adding large-truck training as well. The company currently uses spreadsheets for much of its data management needs—a database application would be useful in order to more efficiently manage enrollees' records throughout different stages within each course's progression through our system.

# Purpose

The purpose of this report is to analyze and document Dominique’s Driving School system. The report features a use-case diagram that gives a visual representation of all the actors involved in the system, along with the processes that they interact with and the stakeholders that benefit from the process. Each process has a use case narrative associated with it that provides more detail about that process. An entity relationship diagram is provided, and the business rules are documented. Two prototypes are included in this report. The first is a prototype screen for the class list interface, and the second is a prototype of the vehicle reservation screen.

# Use Case Diagram



****

# Use Case Narratives

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Class List | |
| **Description:** | Add Class list | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | A Class List must not have duplicates or conflict with other existing classes. | |
| **Trigger:** | Actor needs to create a class without conflict with existing classes in the system, six weeks before the start of the semester. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Actor enters class information | 2. System returns confirmation of the class being added. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The new master class list is added | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Class List | |
| **Description:** | The class list information needs to be updated. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The master class list must exist | |
| **Trigger:** | A master class list information has changed and needs to be updated in the system | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The actor enters ClassID and edits the class information | 2. The system returns a confirmation that the class was being updated. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The master class list is updated | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Instructor | |
| **Description:** | Add instructor to the database | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The instructor does not already exist in database. | |
| **Trigger:** | Instructors are added to the database before they can teach a course. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Actor uses the system to check if the name of the instructor is already in the database.  3. Actor clicks register new instructor, fills in instructor information. | 2. system either shows if the instructor already exists or does not exist.  4. system makes a new entry into the database and gives the entry an instructorID.  5. system confirms final actions to the actor. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | Instructor is now added to the database. Asks if the actor wants to add register another instructor. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Instructor | |
| **Description:** | Edit instructor information in the database if it is not up to date. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The instructor tells the Clerk the information is incorrect or not up to date. The instructor must already exist in the database. | |
| **Trigger:** | The instructor tells the Clerk their personal information is incorrect on their form. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Actor goes into the system to see if the instructor is in the system.  4. Actor updates the fields regadring the instructor that needs to be changed. | 2. system confirms they are in the system.  3. system shows an update button.  5. system updates the database with this info  6. system confirms final actions to the actor |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | Instructor info is now up to date and accurate. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Instructor Contract | |
| **Description:** | Add a new contract to the database. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The instructor must exist in the system.  The class must exist.  The course must exist. | |
| **Trigger:** | Dom makes a master class list and needs to assign instructors to teach the courses. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Actor goes into the system and clicks ‘add new contract’.  3. Actor selects the instructor and adds the contract information.  7. Actor confirms. | 2. system shoes the new contract screen and shows a drop-down list of instructors.  4. System confirms that the instructor is available at the date and time inputted.  5. system adds contract to the database.  6. system asks actor if they want to send the contract to instructor.  8. System sends contract to instructor.  9. System confirms contract is sent. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | Instructor has to sign the contract before the deadline. | |
| **Outstanding Issues:** | Formatting the data before entering the database.  We assume the class info and course info are correct. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Instructor Contract | |
| **Description:** | An existing instructor contract within the system has its information updated. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | Instructor contract must exist in the database. | |
| **Trigger:** | Instructor information has been changed and needs to be updated on the contract. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The user searches the existing instructor contract information.      3. The user selects the contract and edits the information as needed. | 2. System returns a list of instructor contract information.      4. System updates the contract information.    5. System confirms the updates and returns a success message to the user. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The instructor contract is updated. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Email Class List | |
| **Description:** | Dominique emails each instructor on the Instructor file | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | Dominique has ‘fleshed out’ the ‘draft’ master class list | |
| **Trigger:** | A ‘draft’ master class ‘fleshed out’ by the actor | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The actor checks and selects a list of instructors.  3. The actor will send the copy to each instructor, asking if they’re interested in teaching the course on the ‘draft’ class list. | 2. The system displays the list of instructors  4. The system validates the email that it has been sent |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | Instructors receive the email | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Cancel Class. | |
| **Description:** | If six students or less are registered for a single class, the class will be canceled. Students, instructors, and accounting will be notified, funds will be returned to students via mail, and instructor contracts will be voided. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | Class must exist.  There must be less than six students registered for a class  The class has not been canceled. | |
| **Trigger:** | The master class list is checked each week for six or more students per class. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1.Dominique looks up master class list information.    3.Dominique updates the master class list and voids the instructor contract for the specific course. | 2.System returns the master class list.    4.System returns updated master class and voided instructor contracts.    5.System confirms cancelled class and voided instructor contract. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The class is cancelled. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Course | |
| **Description:** | Course needs to be created for offering new subjects. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The course cannot exist | |
| **Trigger:** | A new course needs to be created for a new subject. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Dominique accesses the course database.    3.Dominique creates and adds new course to course database. | 2.System returns course database.  4.System confirms new added course. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | New course is added. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Course | |
| **Description:** | An existing course within the system has its information updated | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The course being edited must exist within the system | |
| **Trigger:** | Course information has been changed and needs to be updated within the system | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The Clerk searches the existing course information.      3. The Clerk selects a course and edits the information as needed. | 2. System returns a list of course information.    4. System updates the course information    5. System confirms the course information has been updated. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The existing course has edited and updated information. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Student | |
| **Description:** | A student needs to be added to the system so that the student can register for a class. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The student cannot exist. | |
| **Trigger:** | A student wants to register for a class but does not exist in the system. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The user enters student information | 2. System returns confirmation of the student being added. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The new student is added | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Student | |
| **Description:** | A student’s information needs to be updated. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The student must exist | |
| **Trigger:** | A student’s information has changed and it needs to be updated in the system. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The user enters StudentID and edited student detail | 2. System returns confirmation of the student being updated |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The student is updated | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Student Registration | |
| **Description:** | Clerk registers student for a class. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The student must be registered in the database to register for any classes.  Class must exist.  Course must exist. | |
| **Trigger:** | A student asks to register for a class. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Actor selects the student              3. Actor adds student to class list | 2. System checks that they are registered in the database. Asks Student if their info is correct.        4. system prints out the invoice for the course for the student.    5. system sends an electronic copy of the invoice to accounting    6. system confirms final actions to the actor. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | Copy of invoice is sent to accounting and students must pay before they can start their course. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Student Registration | |
| **Description:** | The Clerk may edit a student’s registration from the existing registration database before beginning the registration process. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | Students must already exist within the registration database | |
| **Trigger:** | Student registration process. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. Clerk accesses the registration database for registration information.      3. Clerk updates registration information. | 2. System returns registration information from the registration database.      4.System confirms updated student registration information |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | Student registration information is updated. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Send Invoice | |
| **Description:** | An invoice is sent to both a student and accounting. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The invoice must be for a student that exists within the system and has been registered for a class. | |
| **Trigger:** | The student registration process is completed | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. With the registration process completed, the clerk prints an invoice for the student.  2. The clerk invokes the system to send an electronic copy to accounting. | 3. The System confirms that the invoice has been sent to accounting. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | An invoice is given to the student and an electronic copy is sent to accounting. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Vehicle | |
| **Description:** | A new vehicle is added to the vehicle inventory so that it is available for future reservations. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The vehicle must be a unique entry to the system. | |
| **Trigger:** | A new vehicle is acquired by the company and must be added to the vehicle inventory. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The Clerk enters vehicle information for a new, unique, entry. | 2. The system returns confirmation of the vehicle being added. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The new vehicle is added to the system. | |
| **Outstanding Issues:** | N/A | |

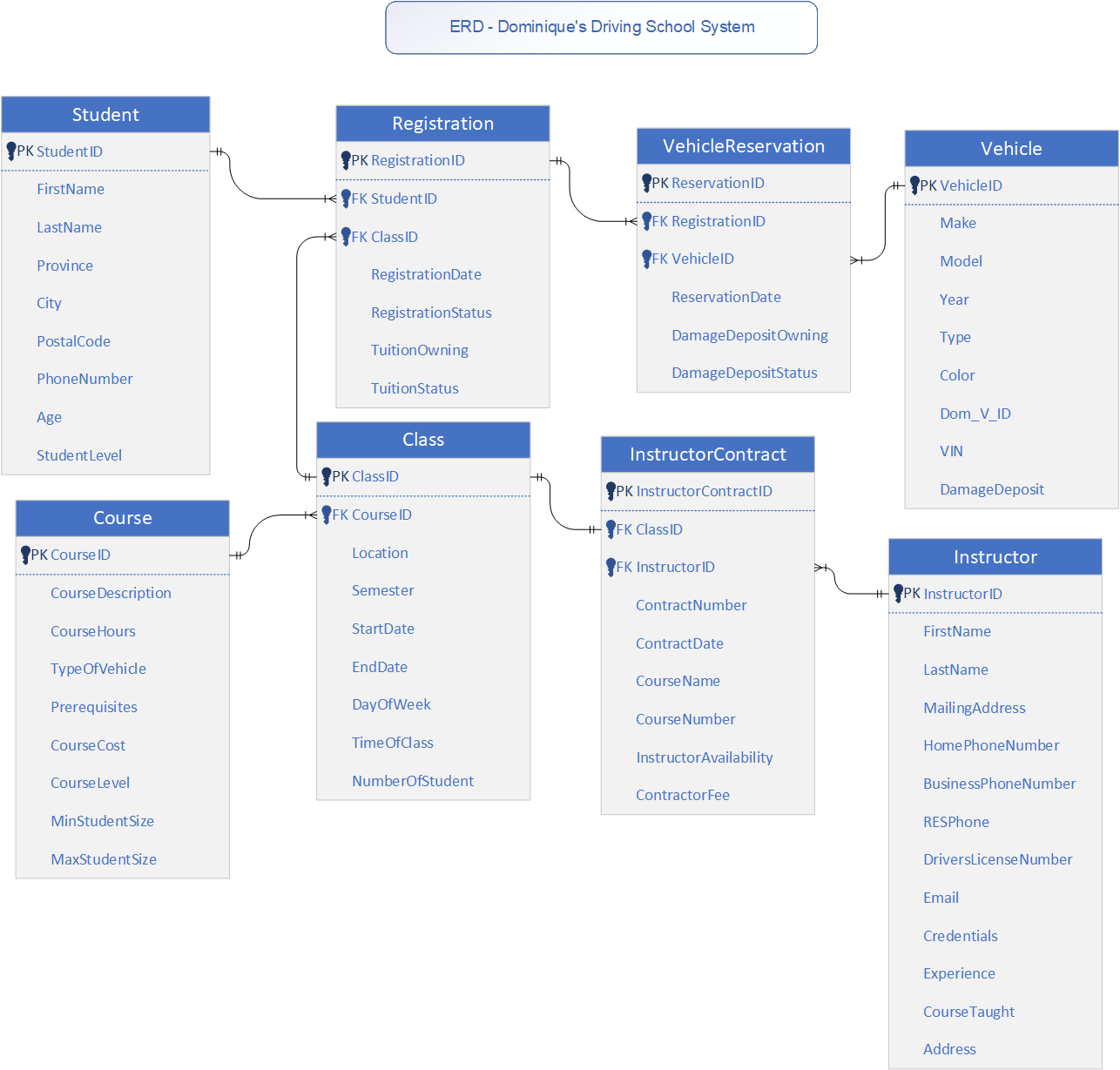
|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Vehicle | |
| **Description:** | An existing vehicle within the system has its information updated. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The vehicle being edited must already exist within the system. | |
| **Trigger:** | An existing vehicle is changed, and the system entry must reflect said changes. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The Clerk looks up an existing vehicle within the system.      3. The Clerk selects an entry and makes the necessary edits to the vehicle data. | 2. The system returns a list of vehicles in inventory.        4. The system returns confirmation of the vehicle data being edited. |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The existing vehicle has its information edited | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Vehicle Reservation | |
| **Description:** | A student is appended to vehicle data for reservation, availability allowing. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The vehicle must exist within the system and must be available for reservation at the time requested. The student requesting a vehicle reservation must exist within the system and must have paid their class fees in full to be eligible. | |
| **Trigger:** | A student makes a request to the Clerk to reserve a vehicle for a class. | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The Clerk does a search in the system for a student, to verify that their fees have been paid.    3. The Clerk selects the class, time, and student.    5. The Clerk verifies which vehicle is required for the student’s request and then reserves the vehicle for that student    7. The Clerk prints the vehicle reservation form for the student and forwards damage deposit information to accounting. | 2. The system returns the requested student information to the Clerk.    4. The system returns a class list alongside a list of vehicles that are eligible for reservation.      6. The system returns confirmation of the reservation being made |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The vehicle is reserved for the pertinent student during the requested time slot. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Edit Vehicle Reservation | |
| **Description:** | An existing vehicle reservation’s information is edited. | |
| **Actor(s):** | Dominique/Clerk | |
| **Preconditions:** | The vehicle reservation in question must exist within the system. The vehicle must exist within the system and must be available for reservation at the time requested. The student that requested a vehicle reservation must exist within the system and must have paid their class fees in full to have been eligible. | |
| **Trigger:** | Vehicle reservation information has changed or must be changed within the system | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The Clerk looks up an existing vehicle reservation within the system      3. The Clerk makes updates to the selected vehicle reservation form | 2. The system returns the requested vehicle reservation information      4. The system returns confirmation of the updated vehicle reservation |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The vehicle reservation’s information is edited. | |
| **Outstanding Issues:** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Damage Deposit Process | |
| **Description:** | Damage deposit information is received by accounting | |
| **Actor(s):** | Dominique/Clerk, Accounting | |
| **Preconditions:** | The student must exist within the system and must have made a vehicle reservation. | |
| **Trigger:** | A student completes the vehicle reservation process | |
| **Normal flow of events:** | **Actor action** | **System Response** |
| 1. The Clerk forwards damage deposit information to accounting      3. Accounting opens the damage deposit process, giving the status a value of ‘outstanding’ until the student pays. | 2. The system confirms that a student’s damage deposit information has been forwarded    4. The system confirms the inputted damage deposit information |
| **Alternate flow of events:** | N/A | |
| **Post-conditions:** | The vehicle reservation’s information is edited. | |
| **Outstanding Issues:** | N/A | |

# Entity Relationship Diagram



# Business Rules

* Instructors must sign a new contract every semester.
* Instructors must return the signed contract to Dominique before the semester commences.
* Dominique only uses instructors who are under contract with the company.
* A student must be in the database to register for any classes.
* IF a ‘new student’, the student is added to the database before beginning the registration process. For existing students, the Clerk updates student information, if necessary, before beginning the registration process.
* If fewer than six students register for a class, the class is cancelled.
* The maximum class size is 14.
* After the first day of class, students cannot attend class or reserve a vehicle until payment for the class has been received in full by Accounting.
* Students must pay for the course in full before being eligible to reserve a vehicle.
* A student must come to the office to reserve a vehicle
* All vehicles are reserved on a first-come basis for the semester and remain reserved for that student for that day and time slot for the entire semester.
* A damage deposit is required for each vehicle reserved.
* A damage deposit must be paid in full before the student can use a vehicle.

# Screens and Reports

## Class List Prototype

## Class List Prototype Writeup

**Purpose:** The purpose of this screen is to allow the user to search for a specific class by entering and selecting a course and then displaying the information. They are then able to print the information.

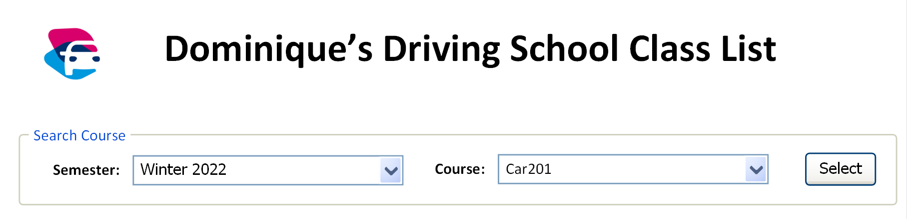
**Frequency:** The class list screen is used every time the Clerk or Dom needs access to the class list information.

**Target Audience:** This screen is used by Dominique and the Clerk. The printed report is used by the instructors.

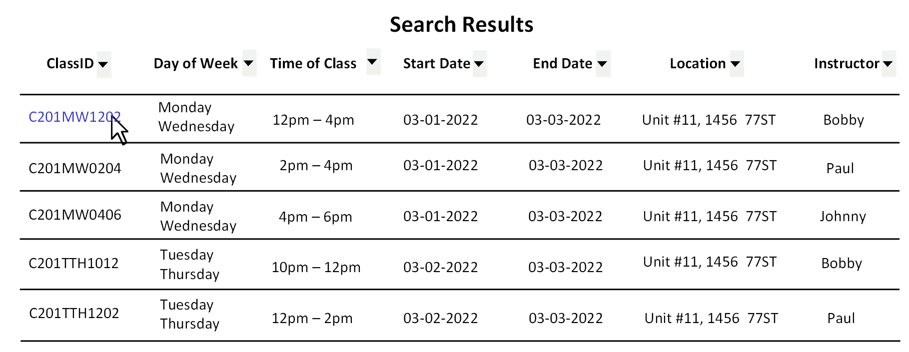
**Rationality:** The screen is divided into three different sections: search, search results, and class information. ‘Search’ contains two parameters that the actor can use to find the specific class. The search results section displays all the corresponding results. Clicking a specific row in the search results section will populate the class information section.

### Narrative

1. The Clerk searches for a course via the combo-dropdown box. It can be typed in as a partial search or selected through dropdown. Once the course they are looking for is in the search bar, they hit select.



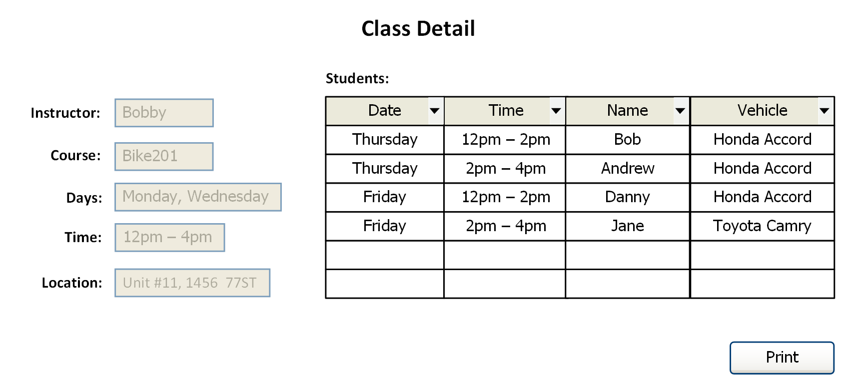
1. The search results are populated below the search area in the search results section. The results display all the classes that are teaching that specific course. The search results can be sorted by clicking the arrows by the column titles. The Clerk can then select a class in the search results table.



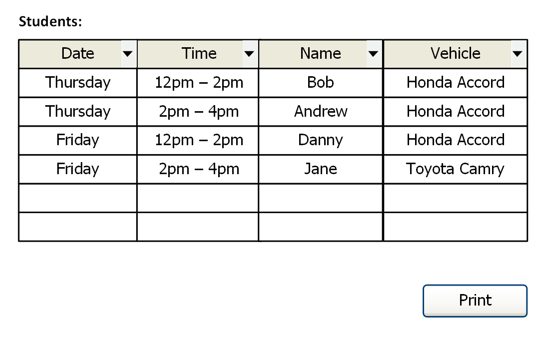
1. When the actor selects the specific class the weekly class schedule is generated to give a visual representation of the data, and the class detail section is populated with information related to the class.

Table

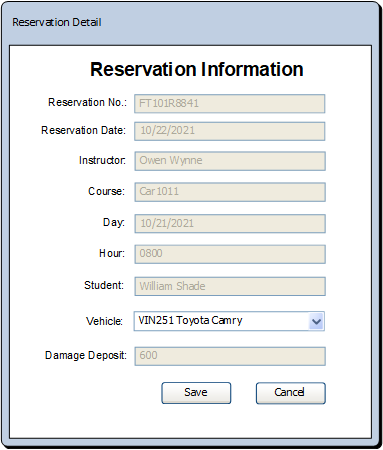
Description automatically generated



1. The Clerk can sort the student list by date, time, name, or vehicle via the arrows beside each column title. The class information can be printed by clicking the ‘print’ button.



## Graphical user interface, application, table Description automatically generatedVehicle Reservation Screen Prototype



## Vehicle Reservation Screen Prototype Write Up

**Purpose:** The purpose of the vehicle reservation screen is to allow the user to search for a vehicle reservation by Instructor, along with other optional parameters, to view and interact with the corresponding vehicle reservations.

**Target Audience:** This screen is used by Dominique or clerks.

**Frequency:** It is used every time that a vehicle reservation needs to be created or modified for a student. It is also used when a vehicle reservation report needs to be printed for an instructor.

**Rationality:** The screen layout for the Vehicle Reservation keeps Screen Flow design theory in mind. The screen sections are placed in order as they should be interacted with.

### Narrative

1. The Clerk can input parameters to search and display vehicle reservations based on instructors. The instructor is the only required parameter in the ‘Reservation Search’ section. The Clerk can search instructor by partial entry or by selecting them in the drop-down list. Other search parameters may be entered for additional filtering.

Graphical user interface, text, application, email

Description automatically generated

1. The search button will populate the ‘Reservation’ grid based on the search parameters; the clear button will clear the ‘Information Search’ parameters along with the ‘Reservation’ grid. The ‘Reservation’ grid can be further filtered with the arrows next to the column titles.
2. The user can print the ‘Reservation’ grid contents once populated or open a row within the grid by using the ‘open’ button. If a row in the grid is opened, an additional window will be brought up, displaying the relevant information for that reservation.

Table

Description automatically generated

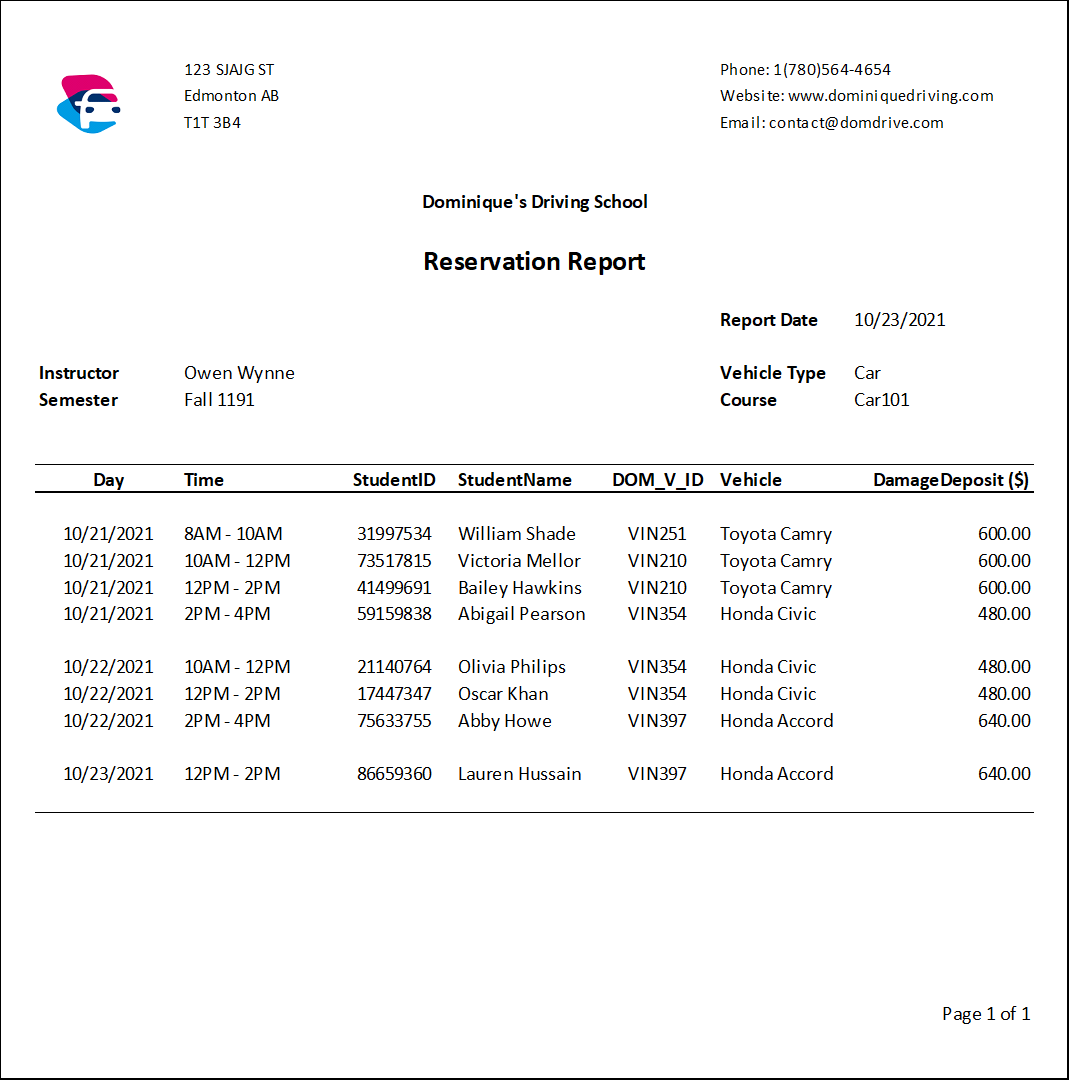
1. In this ‘Reservation Information’ window, a vehicle can be specified as being reserved for a class. Clicking the open button defaults to opening the first record in the vehicle reservation grid.

Graphical user interface

Description automatically generated

## 

## Vehicle Reservation Report



## 

## Vehicle Reservation Report Narrative

**Purpose:** The purpose of this report is to show the details and number of reserved vehicles that have been used all throughout the semester under each course.

**Target Audience:** This report is used by Dominique, clerks, and/or instructors.

**Frequency:** This report is printed every time the vehicle reservation details need to be reviewed for an instructor.

**Rationality:** Information on the report is placed by following common report creation conventions. The most important information is placed in easy to find areas. The report’s design focuses on report efficiency above all.

# 

# Summary and Recommendations

In this report, we designed the physical models and documents based on the existing Dominique’s Driving School system. Team Placeholder visualized the system by creating a Use Case Diagram, breaking up the individual processes into multiple use case narratives.

The team has also created two screens. One for the vehicle reservation window to view and update vehicle reservations. As well as a screen for viewing class-lists, filterable by the school’s instructors alongside a slew of other, relevant, parameters.