### User Story 1: Add a Valid Person.

As a System Admin, I want to add a new client to the system, so that they can access services such as updating their personal details, booking tests, and using other features offered by the platform.

**Acceptance criteria:**

1. All required fields must be provided in the correct formats before adding the client to the platform.
2. RoadRegistry staff member can access the “Add New Person” form.
3. The RoadRegistry system should display a confirmation message on successful onboarding of new client.

### User Story 2: Update Personal Details.

As a RoadRegistry client, I want to update my personal details (such as my home address), so that my records on the platform remain accurate and up to date.

**Acceptance criteria:**

1. The RoadRegistry client can access the “Update Personal Details” section of the platform.
2. The user must be logged into the system before being able to access the “Update Personal Details” section of the platform.
3. The system must validate that any changes to personal details match the correct format for each field in question (email must be valid format; address must be valid address in Victoria, etc.).

### **User Story 3: Reject Invalid Person**

**As a System Admin**, I want the system to reject invalid person entries so that incorrect or unsafe data is not saved in the platform.

**Acceptance Criteria:**

1. The system returns *false* if the person ID is not in the correct format.
2. The system does not insert person data into the TXT file if the address is not in the required format.
3. The system ensures that the birthdate must follow the *DD-MM-YYY* format, or the addition fails.

### **User Story 4: Validate Update Logic**

**As a System Admin**, I want to ensure that updates are only accepted when all rules are satisfied so that the system enforces data integrity.

**Acceptance Criteria:**

1. All format checks from the *addperson* function are re-validated during updates.
2. Any violation of conditions like changing both ID and birthday results in *false* and no file changes.
3. Valid updates modify the correct line in the TXT file.

### **User Story 5: Add Demerit Points**

**As a Police Officer**, I want to add demerit points to a person’s record so that traffic offenses are recorded and can affect driving status.

**Acceptance Criteria:**

1. Demerit points must be whole numbers between 1 and 6.
2. The offense date must be in the forma *DD-MM-YYYY*.
3. Points are appended to the person’s record in the TXT file.

### **User Story 6: Suspend Underage Drivers**

**As a Police Officer**, I want the system to suspend drivers under 21 if their total demerit points within 2 years exceed 6 so that young drivers are penalized correctly.

**Acceptance Criteria:**

1. If a person is under 21 and has >6 demerit points in 2 years, *isSuspend* becomes true.
2. Date comparisons are checked to ensure the offenses are within a 2-year window.
3. Suspension status is updated in the TXT file.

### **User Story 7: Suspend Drivers 21+**

**As a Police Officer**, I want the system to suspend drivers over 21 if their demerit points exceed 12 within 2 years so that the law is applied equally.

**Acceptance Criteria:**

1. If the total demerit points in 2 years exceed 12 for drivers over 21, the person is suspended.
2. Suspension status is calculated based on offense dates and total points.
3. Updated suspension status is saved in the TXT file.

### **User Story 8: System Logging and Feedback**

**As a Developer**, I want the system to return messages and modify the TXT file only when operations are valid so that debugging and auditability are easier.

**Acceptance Criteria:**

1. *addperson* and *updatePersonalDetails* return *true* only when the file is updated.
2. *addDemeritPoints* returns *“Success*” or “*Failed”* based on rule validation.
3. All methods modify the correct person entry and append or update the TXT file accordingly.