Project Scope Document

Project Title: Uber & Lyft Trip Logger (Desktop Version)

Project Owner: Lester S. Carcamo

Prepared For: Project Stakeholders

Prepared By: [Your Name or Team Name]

Date: May 31, 2025

# 1. Purpose

The purpose of this project is to develop a desktop-based Python application that enables the structured logging, management, and reporting of Uber and Lyft trip data—including mileage, revenue, and business-related expenses—from an operator-owner perspective. The application is designed to assist independent rideshare drivers who manage their own vehicle operations and financial records, providing them with a practical and self-contained tool for daily operational oversight and long-term financial tracking.  
  
The system will support detailed expense tracking across common categories such as fuel, maintenance, carwash services, and insurance. It will also implement a receipt management workflow that ensures each expense entry is linked to a properly named and stored supporting document. This supports compliant recordkeeping practices suitable for audits, tax preparation, or business performance analysis.  
  
The application will feature a modular codebase, a graphical user interface for data entry and reporting, and will be delivered as a standalone executable for easy distribution and use. It will also include comprehensive documentation for both end-users and developers to support usability, maintenance, and future extensibility.

# 2. Scope of Work

## 2.1 In-Scope Features

The following features and deliverables are included within the scope of Version 1:

Application Functionality:  
- Desktop GUI using Python and Tkinter for manual input of:  
 - Uber/Lyft trip data (date, time, miles, balances, notes)  
 - Business expenses (gasoline, carwash, maintenance, insurance, etc.)  
- Automatic calculation of revenue per trip by computing balance deltas  
- Receipt management:  
 - File selection via GUI  
 - Standardized filename generation (e.g., EXP00023\_gasoline\_2025-06-01.pdf)  
 - Copy/rename of selected receipt into organized folders  
 - Database references to receipt\_folder and receipt\_filename for lookup  
- SQLite database for persistent storage of trip and expense records  
- View, edit, and export functionality for both trip and expense data  
- Reporting panel summarizing revenue vs. expenses over user-defined ranges  
- Excel export of filtered trip and expense data using openpyxl or equivalent  
  
Application Packaging:  
- Creation of a standalone executable using PyInstaller (or similar)  
- Packaged application to include all necessary dependencies  
- Delivered as a single zip or installer file for end-user distribution  
- Inclusion of startup routines to create default directories (/Receipts, /Database) on first launch  
  
Documentation:  
- Developer documentation:  
 - Application architecture and file/module organization  
 - Database schema and logic explanations  
 - Packaging process and Git usage  
 - Maintenance instructions for future development  
- End-user documentation:  
 - Installation and usage guide  
 - Step-by-step instructions for all key workflows (trips, expenses, receipts, reports)  
 - Backup and file organization best practices  
  
Development Environment and Version Control:  
- Development to occur in a structured project folder with named modules (e.g., \_1000\_main.py)  
- Source code to be version-controlled using Git  
- Hosted on GitHub (or equivalent) in a private repository  
- .gitignore to exclude local-only assets (e.g., receipt images, SQLite data)  
- GitHub to serve as the official code repository, version history system, and cloud backup

## 2.2 Out-of-Scope Items (Version 1)

- Web-based or mobile application versions  
- API integrations with Uber or Lyft platforms  
- Real-time GPS tracking or location services  
- Multi-user access, authentication, or remote database functionality  
- Receipt image OCR or content parsing  
- Automatic syncing with accounting or tax software

# 3. Deliverables

- Fully functional modular Python application source code  
- Local SQLite database schema and initialization logic  
- GUI components for trip input, expense input, reporting, and file selection  
- Excel export capability for filtered datasets  
- Packaged executable for Windows desktop users  
- Organized receipt folder structure  
- Developer documentation (markdown and/or PDF)  
- End-user guide (PDF)  
- GitHub repository with complete version history

# 4. Success Criteria

- The application allows entry of multiple trips and expenses per day  
- Revenue is correctly computed based on input balances  
- Expense entries are linked to externally stored receipts with predictable filenames  
- All application data is persisted across sessions and stored in SQLite  
- Reports and Excel exports match user expectations and filters  
- The packaged application runs without requiring separate Python installation  
- Developer and user documentation are complete, clear, and up-to-date  
- The GitHub repository includes commit history, documentation, and packaging instructions

# 5. Constraints and Assumptions

- Development and testing will be performed on Windows-based systems  
- Receipt files are managed by the user outside the application and stored in local folders  
- Application assumes single-user operation with no login or cloud syncing  
- GitHub will be used as the primary code repository and version control platform