**TRAVEL MANAGEMENT SYSTEM**

**Bhanu Prakash Pakanti**

**Q102147207**

**Leader : Dr. Taiwo Ayodele**

**Software Design and Development**

**Table of Contents**

[Introduction 3](#_Toc164937063)

[Problem Scenario 3](#_Toc164937064)

[Project Objectives 3](#_Toc164937065)

[Functional Requirements 4](#_Toc164937066)

[System Architecture 4](#_Toc164937067)

[Implementation Details 6](#_Toc164937068)

[Demo and User Interaction 8](#_Toc164937069)

[Future Enhancements 9](#_Toc164937070)

[Conclusion 9](#_Toc164937071)

[References 11](#_Toc164937072)

# Introduction

The Travel Management System exercise offers a reliable solution to the indication of trouble that Solent Trips, a fictional travel company, faces in its paper based administration system. The project will attempt to simplify the communication, minimize the documentation, and to aggregate trip management procedures by means of the digital solution. The purpose of this report is to give a deeper insight into the issues encountered by Solent Trips, and to show why a digital transformation is crucial. And especially provides a summary of report structure, which outlines the main parts discussed so far. Through assessing the specified functional requirements, system architecture, implementation details, user interaction, and further areas for improvement, it is intended to enable a deep-rooted understanding of the Travel Management System project that will lead to increased efficiency and effectiveness within Solent Trips. The Travel Management System project is tackling this change by means of joint teamwork and developing a new level of executive examples.

# Problem Scenario

The imaginary travel agency, Solent Trips, is paper based travel management system, and also there are communication issues between staff teams and travellers. This result causes the deviations of planning, gaps, and disconnections in traces of travel and the passengers information. The current system is not efficient and error prone, because of it offline management it faces many challenges. And is looking for the digital solutions where the management of trips and travellers is more efficient and can be accessed via console application with database connected which serves the central system of the TMS. To enanble and develop this system we have implemente this Project and this report discuss of it

# Project Objectives

The main objective is to provide a centralized platform for managing all aspects of a trip, ensuring a smoother and more organized process for both staff and travellers. The system incorporates various functional requirements, including trip management, traveller management, trip leg management, user management (Trip Coordinator, Trip Manager, and Administrator), and reporting and analytics. The modular architecture, object-oriented design, and clear documentation ensure maintainability and extensibility for future enhancements, such as integration with external systems, advanced reporting features, and development of web-based or mobile interfaces.

# Functional Requirements

**Trip Management:** Create, view, update, and delete trips; assign a trip coordinator and define trip detailing.

**Traveller Management:** Include the functionality to view, create, edit, and delete traveller profiles, capture all essential travel info and ensure the provision of a valid government ID.

**Trip Leg Management:** Specify the trip legs and their respective information such as the origin location, destination, transport provider, means of travel, and cost.

**Trip Coordinator:** Control passengers, revise travel legs, compose the route chart and handle the sum and payments.

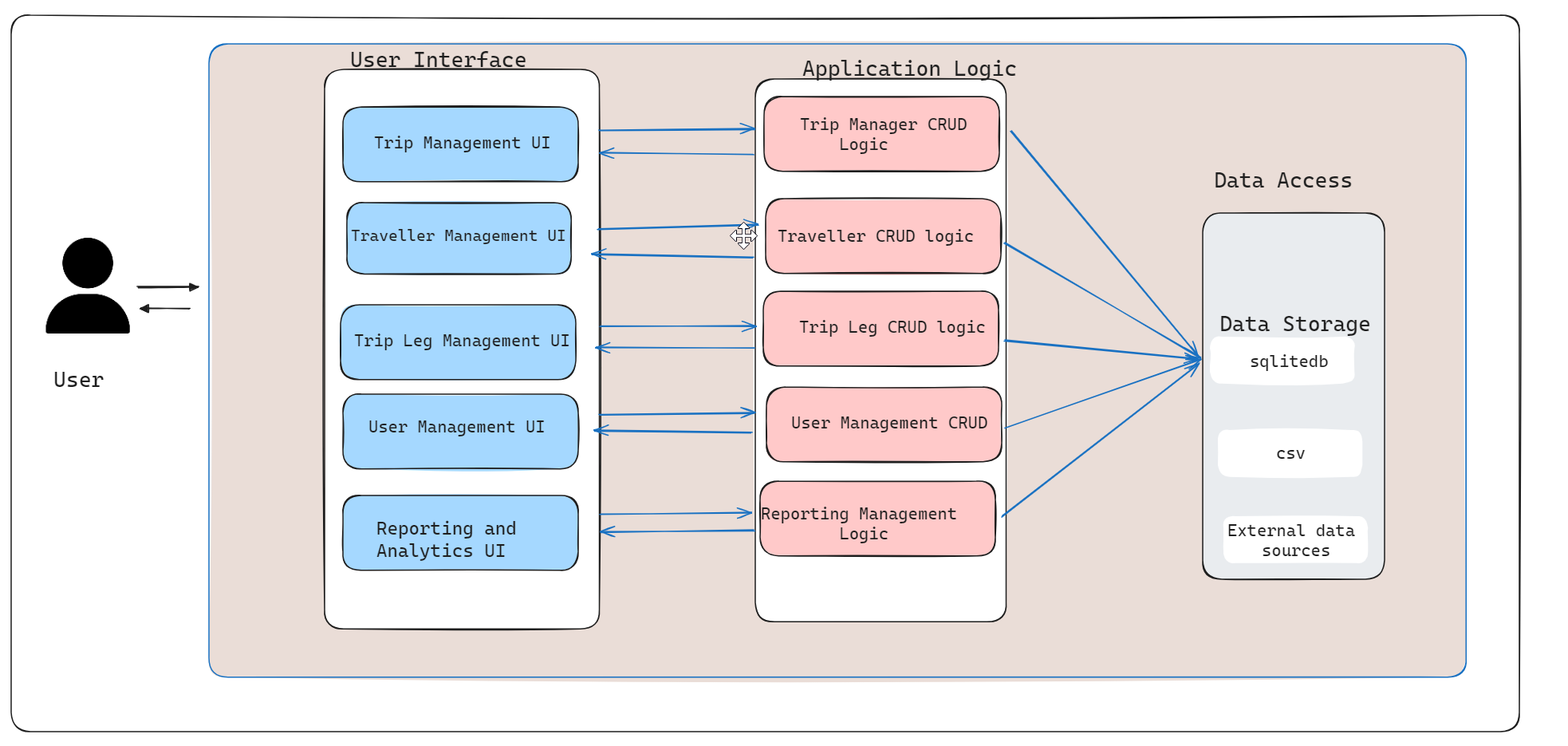
**Trip Manager:** Perform the duties of a Trip Coordinator, supervise the appointed trip coordinators and generate all necessary invoices.

**Administrator**: Execute the duties of a Trip Manager, have the full authority to hire Trip Managers, retrieve invoices and payments, and generate reports.

**Reporting and Analytics:** Develop the module that should render the reports of trip performance, financial summaries and passengers data.

# System Architecture

The Travel Management System is built in the modular architecture(Lac, É., et el.,), as illustrated in Figure 1, which means the system has a structured and organized method to deal with the different aspects that start with travel planning and ends with the execution part of the plan. This modular design makes the system grow and link-able, hence [without the complete severance of the entire system] a new element can be linked and the system evolved, extended or modified as per the author (Moore, A.D)



**Figure 1: System architecture in detail**

(Source: Excalidraw)

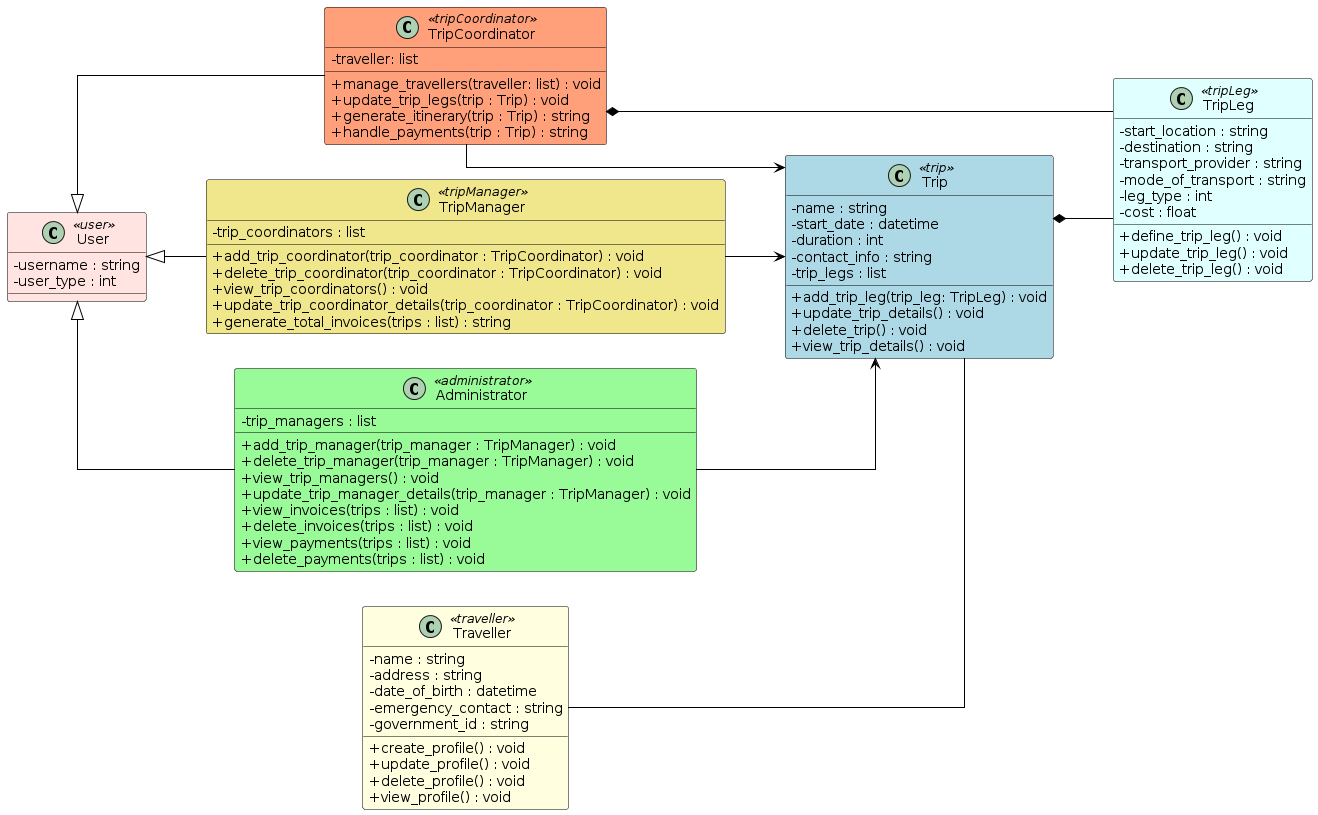
This framework captures the main blocks of the system, each serving a particular purpose. The Task Management has a feature to create, view, edit and delete trips, as well as appointing trip coordinators, specifying trip details like name, start date, duration, and contact information. Here, this vital module is the primary one that leads to the heart of the system, it helps to maximize trip metadata management. Traveller profile can be created the Adminstrator who can perform all the actions for the traveller details like CRUD operations.

The Trip Leg Management comprises of underlying attributes which are fundamental to a trip is its definition of the different legs or segments of a journey. For the start location and destination it permits setting only. Then, it offers to distinguish between transport providers or selecting mode of transport and respective costs per each trip. This section also allows the management of different types of legs, consisting of accommodations, sightseeing, as well as transfer points, offering the convenience of knowing an itinerary in one page. The User Management section in the system holds the task of managing the different user roles available in the system, which include Trip Coordinators, Trip Managers, and Administrators. That particular function has its own specific security permissions and permissions , which allow to apply proper access control and to segregate who does what duty. Trip Coordinators can manage passengers and add or remove trip legs to their assigned trips, while Trip Managers have more responsibilities, they not only generate Trip Coordinator profiles and manage the trips that they oversee but also invoice the trips they manage.

Reports and Analytics igives a very detailed account on the performance of your trips as well as financial reports. Here, the TMS sticks to the module architecture Not only does clean separation of concerns and clear defined boundaries between components streamline them to improve the look and feel, but it also promotes forward-thinking enhancements like integrating with external systems, implementing advanced reporting features or creating web-based or mobile interfaces without disrupting the existing functionalities.

# Implementation Details

The Travel Management System (TMS) for Solent Trips implementations uses Python programming language (Python) and multiple libraries to code a functional software solution that covers all functions requirements as mentioned for this assement where such application development is mentioned in the paper (Fitzpatrick, M., 2021). The implementation includes the industry's best practices for software development, following the principles of modulurity, code reuseability, and maintainabilty. The UML Diagram is first created to Analyse the relationship Between the class and its Atributes to understand the application attributes which helps in the development More easily as per the Figure 1. This architecture enables and assures the role-based approach of passenger trip management. In the UML connected diagram the implementation details of the travelers manage their accounts and book their trips whereas the Trip Organizers create schedules. Trip Coordinators are involved in the minutiae of each trip phase. Passengers management and the trips fall within role of the Trip Manager. The last, Administrators take care of managing user account and invoices. This distributed structure ensures appropriateness of the trip management by assigning specials tasks to the certain positions in the group.

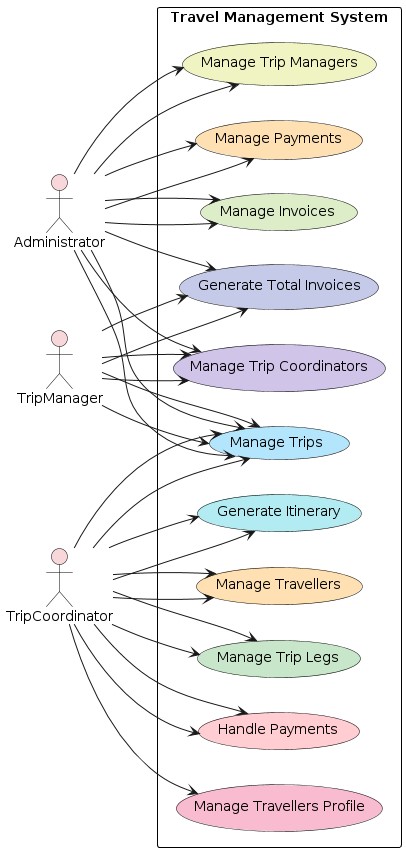


**Figure 2: UML Diagram**

(Source: PlantUML)

The Travel Management System is the end-product of using Python with Python's wide capability and extensive libraries for development of software. The architecture is object-oriented type (Figure 2) and wide-ranging functionality is supported by the object-based approach as well by the modularity, maintainability, and extensibility. The classes and their relations are thought out and developed so that the subset of functionalities can be routinely used, code can be cleanly reusable and the enhancement of the system will be facilitate. In addition, the functionality of the system is explained through the Sequence Diagram which is (Figure 3) depicts all the interactions between the system objects and how the control flow is maintaned within the system. The given model will explain the system's performance, interaction patterns, and order of the operations conducted by individual components so that the project is not messy and confusing.

**UseCase Diagram**

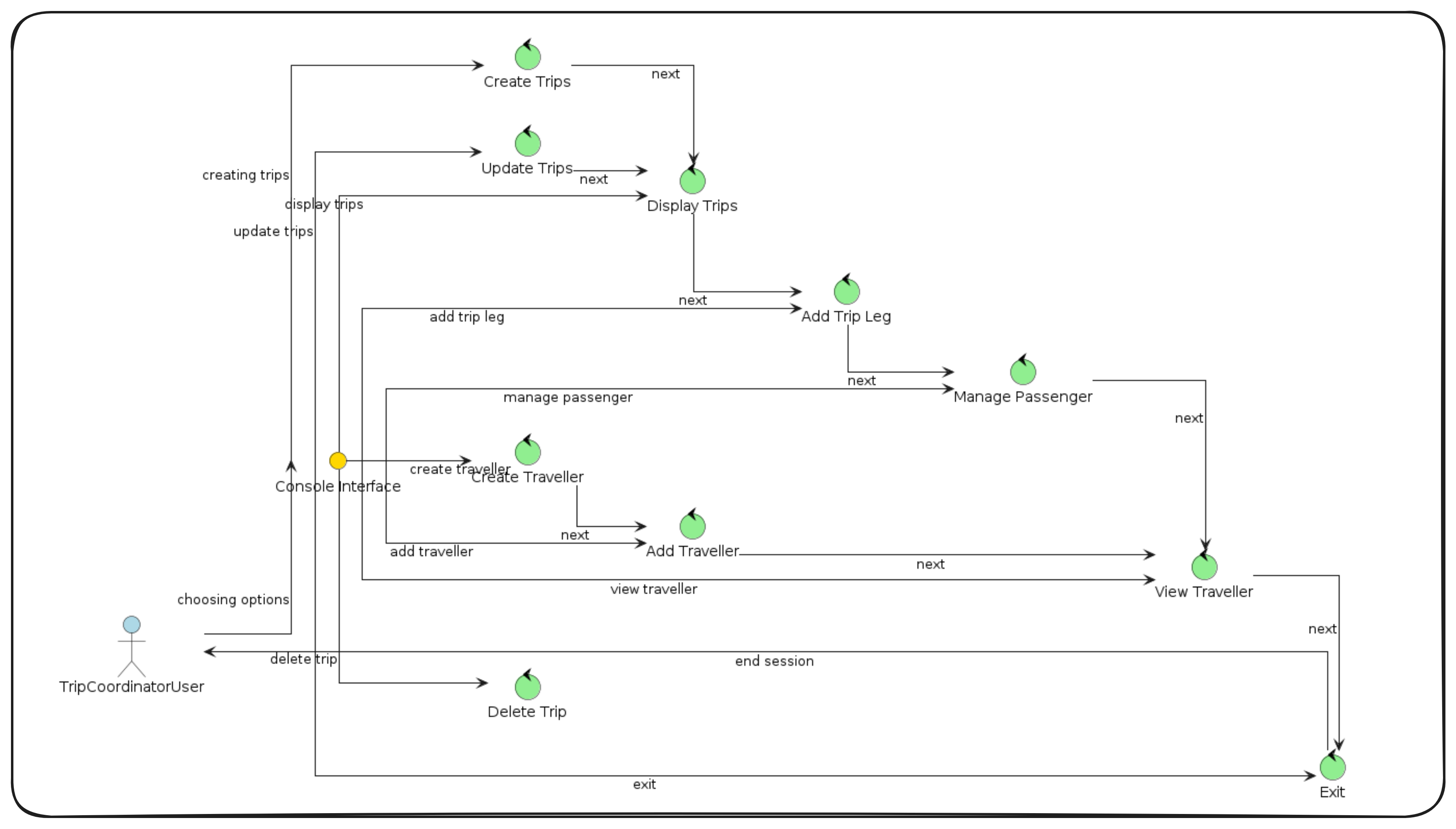


**Figure 3: UseCase Diagram**

(Source: PlantUML)

TMS implementation will be based on PyCharm(Van Horn II, B.M. and Nguyen, Q., 2023), an IDE that aiding in Python development with its powerful tools and features. Python 3.8+ is considered as it is easy to understand, highly readable, and has abundant packages. In the Main.py file provided all the required classes function in order to achieve the digitalization of the application in regard the TMS. And also to test the functionalities we have written Unit Test Cases.

**Robustness Diagram**



**Figure 4: Robustness diagram**

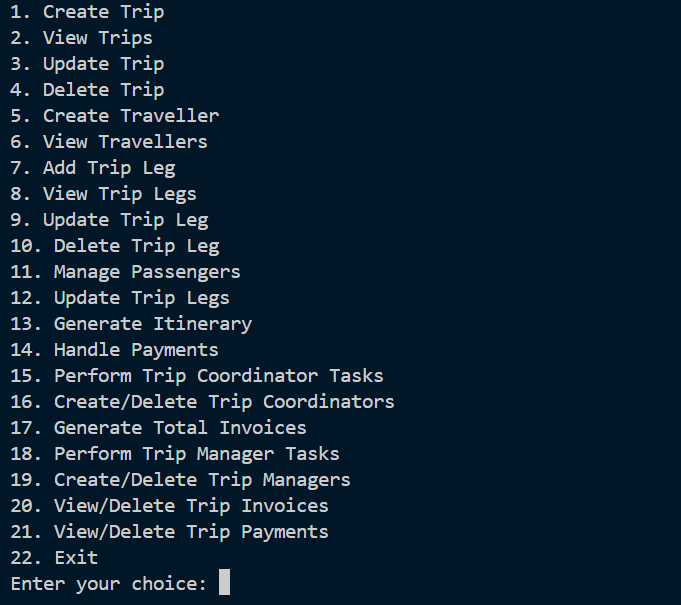
(Source: Created in Plantuml,Excalidraw)

The system is designed using double-layered user structure with regard to trip management. Trip Coordinators are the most authorised people. They are able to create and administrate itineraries, manage passenger information and invoices. In fact, they develop an itinerary and carry out the whole trip management process.

The TMS implementation is composed of libraries and modules that operate over various functionalities. The `csv` library is used for importing and exporting the information about the trip and the travellers in CSV files if in cases needed.

# Demo and User Interaction

The Travel Management System (TMS) for Solent Trips demonstration involves sharing the system's maximum functionalities and user interactions. In the Developed users who are Administrators, TripCoordinators, TripManager will run the application, since this is console built application ther will be technical Knowledge Needed, Similar to the console application built now, we can achieve the same in the GUI using Tkinter as per the author Amos et el., we need to run the main.py file it will display menu options in the console, we select the option as per our Requirements and necessary and perform the activity. Below is a Screenshot attached as figure 4 from the pycharm Console of the working Application with the menu. Now the User can perform the CRUD operation as per they permission and limition given which are mentioned already in the class diagram in figure 2 and the sequence diagram figure 3.



**Figure 4: Demo Implementation**

(Source: Implemented in PyCharm)

# Future Enhancements

# The current structure of TMS addresses the basic needs, nevertheless, there are some possibilities for improvements in the future. Some possible improvements include:

1. Implemented the web based application where travellers where it will lessen the burden of the Administrator to update details
2. More Libraries to show the Dashboard activities of the operations in TMS

# Conclusion

The Travel Management System is the unit that integrates all the elements problems of the current paper-based system and therewith it provides a comprehensive solution to Solent Trips. This system helps to streamline process of trip organizing, as it takes an advantage of Python and corresponding engineering practices to offer a single platform for journey information management, travellers and other details. The modular architecture, object-oriented design, and properly writing comments with the understandable descriptions ensure maintainability and extensibility of the application running in tF[[1]](#footnote-1)he future as well, needs any kind of changes.

# Artefacts <https://github.com/bhanuprakashpakanati/travel-management-system.git> <https://github.com/bhanuprakashpakanati/COM714.git>

# References

Amos, D., 2020. Python gui programming with tkinter. *Tersedia: https://realpython. com/python-gui-tkinter*, p.71.

Moore, A.D., 2021. *Python GUI Programming with Tkinter: Design and build functional and user-friendly GUI applications*. Packt Publishing Ltd.

Fitzpatrick, M., 2021. *Create GUI Applications with Python & Qt6 (PySide6 Edition)*. Martin Fitzpatrick.

Cynthia, J., Mohankumar, T., Arjun, T. and Naveenkumar, C., 2021, December. Development of Python based UI application for tele-operated vehicles. In *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)* (pp. 383-388). IEEE.

Lukasczyk, S., Kroiß, F. and Fraser, G., 2020. Automated unit test generation for python. In *Search-Based Software Engineering: 12th International Symposium, SSBSE 2020, Bari, Italy, October 7–8, 2020, Proceedings 12* (pp. 9-24). Springer International Publishing.

Martin, H., Hong, Y., Wiedemann, N., Bucher, D. and Raubal, M., 2023. Trackintel: An open-source Python library for human mobility analysis. Computers, Environment and Urban Systems, 101, p.101938.

Van Horn II, B.M. and Nguyen, Q., 2023. *Hands-On Application Development with PyCharm: Build*

1.Koç, H., Erdoğan, A.M., Barjakly, Y. and Peker, S., 2021, March. UML diagrams in software engineering research: a systematic literature review. In *Proceedings* (Vol. 74, No. 1, p. 13). MDPI.

Yanis, R.Z.I., Priyadi, Y. and Puspitasari, S.Y., 2022, November. Measurement of similarity between use case description and sequence diagram in software requirement specification using text analysis for dtrain application. In *2022 2nd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)* (pp. 328-333). IEEE.

Arifin, M.N. and Siahaan, D., 2020. Structural and Semantic Similarity Measurement of UML Use Case Diagram. *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, *11*(2), p.88.

Lac, É., Spiegeleer, G., Delsalle, A., Collonval, F., Lê, D.T. and Malandain, M., 2024. CoSApp: a Python library to create, simulate and design complex systems. *Journal of Open Source Software*, *9*(94), p.6292.

1. [↑](#footnote-ref-1)