Field Testing and Evaluation of the Transit Integrated Monitoring System

Final Report
Transit IDEA Project 19

Prepared for

Transit IDEA Program
Transportation Research Board
National Research Council

Prepared by
Manuel D. Rossetti
University of Virginia

August 13, 1998

INNOVATIONS DESERVING EXPLORATORY ANALYSIS (IDEA) PROGRAMS MANAGED BY THE TRANSPORTATION RESEARCH BOARD

This Transit IDEA project was funded by the Transit IDEA Program, which fosters development and testing of innovative concepts and methods for advancing transit practice. The Transit IDEA Program is funded by the Federal Transit Administration (FTA) as part of the Transit Cooperative Research Program (TCRP), a cooperative effort of the FTA, the Transportation Research Board (TRB) and the Transit Development Corporation, a nonprofit educational and research organization of the American Public Transportation Association (APTA).

The Transit IDEA Program is one of four IDEA programs managed by TRB. The other IDEA programs are listed below.

- NCHRP Highway IDEA Program, which focuses on advances in the design, construction, safety, and maintenance of highway systems, is part of the National Cooperative Highway Research Program.
- High-Speed Rail IDEA Program, which focuses on innovative methods and technology in support of the Federal Railroad Administration's (FRA) next-generation high-speed rail technology development program.
- Safety IDEA Program, which focuses on innovative approaches for improving railroad, intercity bus, and truck safety. The Safety IDEA program is funded by the Federal Motor Carrier Safety Administration and the FRA.

Management of the four IDEA programs is coordinated to promote the development and testing of innovative concepts, methods, and technologies.

For information on the IDEA programs, look on the Internet at www.trb.org/idea or contact the IDEA programs office by telephone at (202) 334-3310 or by fax at (202) 334-3471.

IDEA Programs
Transportation Research Board
500 Fifth Street, NW
Washington, DC 20001

The publication of this report does not necessarily indicate approval or endorsement of the findings, technical opinions, conclusions, or recommendations, either inferred or specifically expressed therein, by the National Academy of Sciences or the Transportation Research Board or the sponsors of the IDEA programs from the United States Government.

EXECUTIVE SUMMARY

The results of the IDEA Transit-10 project indicated the potential of using radio frequency identification to integrate automatic passenger counting (APC) systems and automatic vehicle location (AVL) systems. This follow up Phase 2 demonstration examines the integration of APC/AVL and other monitoring functions in the form of a Transit Integrated Monitoring System (TIMS). This project specifically addresses the following three needs: 1) Advanced systems for transit vehicle location, identification, and management of transit systems, 2) Improved methods of data collection about or from transit users, and 3) Advanced systems for fare collection and control systems. The Phase 2 overall project objectives include developing a demonstration system that: 1) improves the hardware components of the system, 2) enhances the software components of the system, and 3) enables a better understand the operational performance of TIMS under transit conditions.

A key feature of this system is its ability to capture information about the passenger trips through the use of radio frequency based tagging devices. An important component of this tracking is the identification of the bus stops associated with a passenger's trip. The hardware system has been redesigned to utilize global positioning technology as the primary bus stop identification mechanism. In addition, the system concept has been changed to allow fare collection on a portable data terminal mounted within the vehicle. The results include a completed software conceptual model of the information content for the system, a set of use cases for the system, a design specification for the database, a design specification for the user interface, and the hardware requirements and specifications. In addition, the software and hardware were implemented and installed on a bus within the Charlottesville Transit System. The system was tested during a field trial of one week. The results of the testing indicated ….. The final product associated with this project is the completed design and specifications of TIMS, the integrated hardware and software, and a demonstrated system.

TABLE OF CONTENTS

1. IDEA PRODUCT

During the TRB sponsored TRANSIT/IDEA TRANSIT-10 project, a prototype was developed which involved passengers carrying radio frequency identification (RF/ID) tags which act as bus passes. Passengers carrying the cards can then be uniquely identified. In addition, RF/ID tags were embedded at bus stops along a route to track the movement of buses. Using multiple antennae mounted on the bus with a decoder and a RF modem, operational data can be collected in near real-time. The data captured included passenger and bus stop identifiers and event times (arrival and departure times of buses and boarding/alighting times of transit users). From these data, information such as origin destination pairs, passenger transit times, and schedule adherence can be derived.

The results of testing the prototype clearly indicated the potential of using radio frequency identification to integrate automatic passenger counting (APC) systems and automatic vehicle location (AVL) systems. This follow up Phase 2 study examines the integration of APC/AVL and other monitoring functions in the form of a Transit Integrated Monitoring System (TIMS). The Phase 2 overall project objectives include:

1. Improve the hardware components of the system
   a) Enhance reliability of data capture of APC
   b) Migrate to fare card sized tags
   c) Integrate global positioning system for AVL
2. Enhance the software components of the system
   a) Update database design
   b) Enhance user interface
   c) Enable fare collection and reporting functions
3. Better understand the operational performance of TIMS under transit conditions

We begin with an overview of the stages of the project and a presentation of the major tasks and deliverables. We then present the TIMS system concept and its software/hardware architecture. Next, the functionality of the system and its information requirements are given in the form of use cases and use case scenarios. The conceptual information model for the system is then presented in the form of conceptual class diagrams for the major software components. Finally, we present implementation details in the form of the database design and the user interface functionality. We conclude with a discussion of the results of field test trials and recommendations for future work.

2. CONCEPT AND INNOVATION

We address the need for innovation in the area of Transit Management Practices with a Transit Integrated Monitoring System (TIMS) in order to improve the methods and procedures for collecting data concerning transit users and vehicles. This project specifically addresses the following three needs (*1*):

1. "Advanced systems for transit vehicle location, identification, and management of transit systems."
2. "Improved methods of data collection about or from transit users."
3. "Advanced systems for fare collection and control systems."

Traditional automatic passenger counting (APC) and automatic vehicle location (AVL) functionality tends to be implemented in separate systems within transit information management systems. Standard APC is premised on the counting of fares by the fareboxes located on the vehicle. More recent advances in fare collection devices include the use of magnetic strip cards and smart cards. Smart cards are plastic credit card sized cards with an embedded computer microprocessor. The computer chip can hold information in electronic form with sophisticated security mechanisms. Smart cards based on radio frequency identification technology are called contactless cards and allow the passive collection of fares and rider usage information. Transit passengers carry the smart card and use it as a bus pass. Radio frequency receivers emitting an energy field activate the transponder within the card, receive the data stored on the card, and then store the data for later transmittal to the database. As a passenger boards the bus, the smart card is activated and an initiation of a trip is recorded. When the passenger disembarks the bus, the smart card is again activated and the completion of a trip is recorded.

Fare information is stored on the smart card such that the smart card is utilized as debit card or the fare for the trip can be tabulated and the passenger's transit usage account could be billed according to some fare structure. As the vehicle travels its route according to a schedule, the vehicle arrives at the bus stops to allow the boarding and alighting of passengers. A global positioning system records the location of the vehicle as it progresses through its route. When the vehicle stops at a bus stop, the bus stop can be uniquely identified and combined with the passenger transactions to identify the origin and eventually the destination of the passenger's trip. A database system is required to support the tracking and analysis of the smart card based transit system. The database should allow for improved route planning, scheduling, and fare collection analysis. The primary users are expected to be transit managers and transit operations engineers.

Figure 1 illustrates the system concept. Passengers carrying the smartcards will be read automatically as they board or alight the bus. The bus will contain a RF decoder for reading and writing to the smartcards and a unit to record the odometer of the bus. In addition, the bus will contain a global positioning unit to track the vehicle location along the route and a mechanism for capturing door open and close events. A portable data terminal will be used to integrate the RF decoder, the GPS unit, and the odometer reader and for storage of passenger and GPS transactions.

3. INVESTIGATION

The original investigation was planned for three stages; however, due to budget constraints, the project scope was narrowed and the project length shortened. The three stages were consolidated to the following two stages:

Stage 1

Upgrade the software and hardware design to accomplish an integrated transit monitoring and management system:

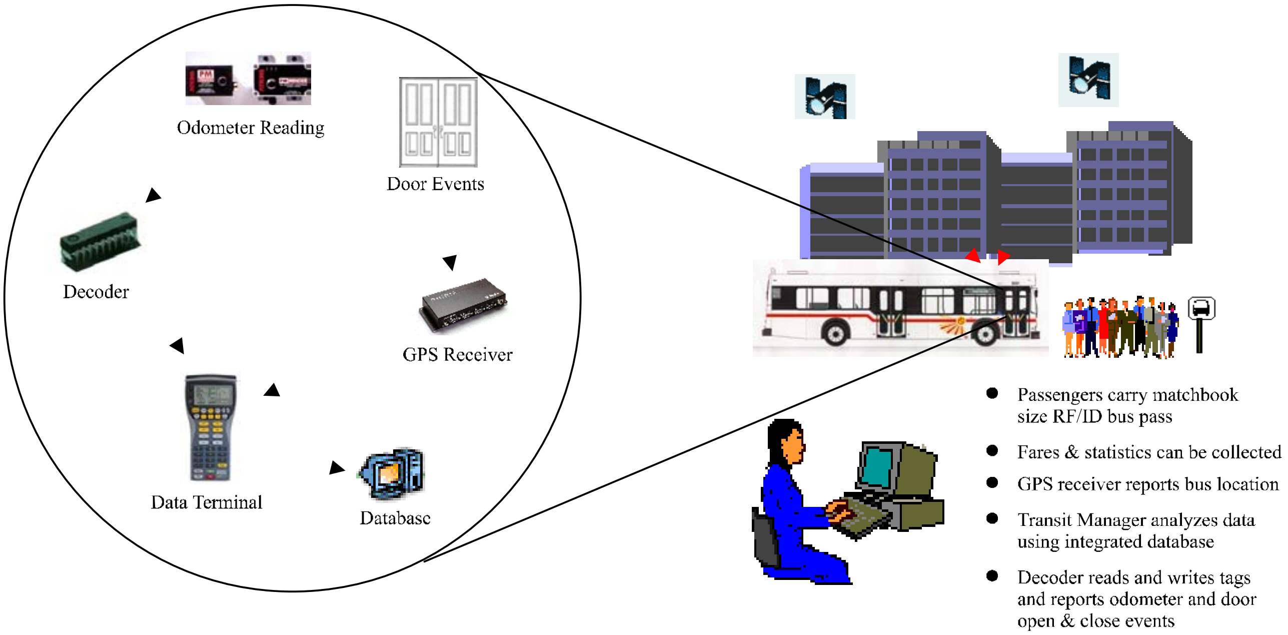
1. Upgrade the RF/ID tag design to reduce the size close to a smart card size,
2. Upgrade the hardware and software system components and integration with APC/AVL and other monitoring systems for improving the reliability and ease of data capture, and
3. Upgrade the user interface for integrated transit management and monitoring systems.
4. Convene a meeting of a regional panel of experts to review the software and hardware design improvements for the project.

Stage 2

Complete the software development and hardware integration and testing. Perform one week of simulated demonstration of the upgraded system from Stage 1 to ascertain the benefits for transit management and monitoring with a transit service provider. Convene the panel of experts to review demonstration results. Develop strategies and plans for outreach and deployment of the IDEA product by transit providers.

The investigation represents a collaborative effort between the Transportation Research Board, the University of Virginia's Department of Systems Engineering, Charlottesville Transit System, and Advanced Systems Group International (ASGI). ASGI's mission is to become a leader in radio frequency technology by delivering integrated software and hardware solutions. Tangible byproducts of the demonstration project include an operationally tested automated data collection system to serve as a model for Transit Agency adoption. In addition, the system requirements will assist transit operators in evaluating and/or implementing automatic data collection based on RF/ID technology

The rest of Section 3 presents the system design including hardware description, software analysis, design, and implementation.



- Passengers carry matchbook size RF/ID bus pass
- Fares & statistics can be collected
- GPS receiver reports bus location
- Transit Manager analyzes data using integrated database
- Decoder reads and writes tags and reports odometer and door open & close events

FIGURE 1 System Concept Diagram

3.1 Hardware Description

The hardware elements for the system include:

1. (1) RF Controller mounted within the bus,
2. (2) RF antennas per bus (front and rear door mounting)
3. 1 Neurotag for each passenger
4. (1) odometer reader or distance measurement device
5. (1) GPS receiver and external antenna
6. a portable data terminal mounted in the bus with solid state flash memory for data storage

This hardware configuration allows bus and passenger RF/ID transactions to be captured. In addition, the GPS unit should be able to capture door open or close events. Both the computer hardware and the GPS unit will be stored on a bus. The units must run on 12 Volts of direct current, withstand normal bus vibrations, and they should be modular to facilitate installation and future upgrades. Modularization will also aid in the storage and protection of the units from vandalism by passengers. Encapsulation of the units is preferable to prevent further damage. The chosen hardware units should allow integration of differential GPS at a future time. An external antenna for the GPS unit is required for mounting on the roof of the bus to optimize reception.

The primary reason for measuring the distance traveled is to assist the accuracy that is available via low cost GPS technology. Current, GPS receivers have a wide variety of performance characteristics. The GPS system has its roots in DOD technology and is currently limited to 100 meters +/- 2 sigma in accuracy because of DOD enforced selective availability (SA), which degrades the accuracy by introducing errors into the signals. Some GPS equipment vendors will state their accuracy with or without SA. Without SA it is possible to get accuracy of about 15 meters. The DOD has promised to phase out SA but this will not occur until some time in the next century.

With 100-meter accuracy, we have a problem, since bus stops could be within 100 meters of each other. This can occur because a stop can be positioned within 100 meters as the crow flies (either via the next stop, across the street, or around a corner, etc.). This makes it difficult to exactly pinpoint the bus stop. A measure of distance traveled facilitates the distinction between stops. Given the distance between stops, the last known bus stop, and the travel distance a set of candidate bus stops can be found. The GPS positions of those bus stops can be compared to the current GPS coordinate to pinpoint the bus stop. A dead-reckoning device or other odometer reading device can be utilized to capture the travel distance. A dead-reckoning system will also improve the GPS accuracy to approximately 60 meters and allow position updates during satellite blockages. Another approach is to utilize a real-time differential correction approach but this adds the additional cost of a GPS base station and the equipment costs to communicate with the base-station in real-time.

3.1.1 RF Controller

The purpose of the RF controller is to interpret signals from the antenna and to transmit information to the tags through signals to the antenna. It converts the incoming analog radio signals from the tags to digital information. The RF Controller sends the data to the host immediately, stores it for later transmission, or readies it for transmission to other tags. In addition, the controller is capable of writing to tags within the antenna field in order to change all or part of the data on an individual tag. The decoder can be connected to a host system via a standard serial interface. The decoder consists of a central processing unit, a radio frequency unit, a communications interface board, and a power supply unit. The decoder is capable of identifying up to 8 tags simultaneously within the interrogation field.

TABLE 1 Controller Specifications

Dimensions:	8” x 8” x 4”
Weight:	3.5 lbs.
Enclosure:	NEMA 1 box 132-RDS-C or W
Power Requirements:	110/220V AC or 5-12V DC
Interfaces:	RS232 is standard FDDI, Ethernet, RS485, LonWorks and Wiegand available
Read Speed	10 kbps

3.1.2 RF Antenna

The antenna produces an interrogation field that will cause tags within the field to “wake up”; i.e. it turns the transponder on within the tag. Attached to the antenna is a tuning unit (ATU) which enables the antenna to be tuned for maximum range and to allow for adjustments. The standard antenna configurations include 1) bar or side read antenna, 2) picture frame antenna and 3) road loop antenna. The bar or side read antenna is approximately 36” x 5” x 2” and is generally used to cover doorways or as vehicle side read antennas.

To shield tags that are already in the vehicle, a customized side-read antenna utilizing a Plexiglas cover and mounted on the wall at the lowest level of the stair wall will be used for both entrance and exit doors. Final dimensions will be determined, utilizing existing antennas as a base, through on-site testing. Discuss final mounting here.

3.1.3 RF NeuroTags

ASGI is in the final stages of an application-specific integrated circuit (ASIC) project with Honeywell. The purpose of the project is to convert the multiple-component RF circuit to a single chip. Test pieces will be available in late August, and ASGI is pursuing funding to package the chip such that one of several smart card manufacturers can embed it with a battery and antenna coil into a prototype smart card by year end. In the interim, the demonstration project will proceed with the NT132, which have a footprint of a credit card and taper to a thickness of 0.187 inches in the center. Update design plans here.

The NT132 is a high performance, long range, programmable, multiple-read active RF device. The NT132 has dimensions of (2.125” x 3.375” x 0.187”) and a weight of (8 oz). It is powered by an internal lithium battery with a 5-year life expectancy depending on rate of usage. The condition of the battery is monitored during transmission. The tag has an environmental operating temperature of -29C to 85C. The tag is capable of storing 1 Kbyte of information and transmitting at approximately 1 Kbytes per second with parity and checksum error checking. The transmission range is 132 kHz for receive and 315 MHz for transmit with an adjustable read range of 4 inches to 30 feet. Tags can send information packets to each other and the human body does not block reading or writing.

3.1.4 PM-Minder Odometer Reader

The PM-Minder is a compact preventive maintenance monitoring system developed by Atkins that can monitor up to five components (e.g. hours run, miles, fuel, and fluid levels). The unit weighs 1 pound and has dimensions of (4.5 x 3.8 x 2.2 inches). The PM-Minder can only report the odometer reading every 3.5 minutes. This may not be necessary given dead reckoning system.

3.1.5 GPS Equipment Selection and Description

Required features:

- Interface with computer via serial port (RS-232)
- Output latitude/longitude and time in ASCII format
- Affordable (maximum price of $1000)
- Portable
- 12 Volt Power Supply
- Automatic power-up with ignition
- Durable
- Accuracy within 20 meters
- Programmable sampling interval (one reading every three seconds)
- Modular
- Outlet for external antenna

Extra features:

- Upgrade to real-time corrections
- Interface with odometer
- AVL expansion possibilities
- Heading sensors
- Vehicle mount

In order to select the appropriate GPS unit, we evaluated 7 different units in terms of price, number of serial ports, output formats, power supply requirements, ignition sensitive, sampling rate, external antenna capabilities, and whether or not the unit was modular. The complete evaluation results were given in the TRANSIT-19 Stage 1 Report. Based on the evaluation the Trimble Placer 455 DR product was chosen to meet the GPS requirements for the project. With this selection of the Trimble Placer 455 there is ample room for future upgrades, such as a complete AVL system. The Trimble Placer 455 has the capability to respond to system events that occur on the bus such as reporting a position if the door on the bus opens. The Trimble Placer 455DR utilizes dead reckoning technology to provide positions during satellite blockages to within 60-meter accuracy.

3.1.6 Portable Data Terminal Selection and Description

Required features:

- Portable
- Easy file transfer to Desktop computer
- Operate on 12 Volt power
- Easy to mount on bus
- Rugged to handle bus vibrations
- Able to collect data without user intervention
- Affordable
- Easy to develop application to interface with decoder/GPS unit
- Easy installation on bus
- At least 4 MB storage capacity
- At least 2 serial ports (RS-232)

Extra features:

- Automatic power up on ignition
- PCMCIA or equivalent

Based on the above requirements, we evaluated 4 portable data terminal devices in terms of cost, number of serial ports, storage capacity, ease of file transfer, mounting options, ignition sensitive, and user interface. The complete evaluation results were given in the TRANSIT-19 Stage 1 Report. Based on this evaluation, the Psion Workabout was chosen as the portable data terminal. The Psion Workabout allows simpler integration of the other hardware components at an affordable price.

3.2 Operational Description

The system should log the GPS location information of the bus continuously after vehicle ignition. Then, during operation, when the door opens the distance from the last stop should be computed. As passengers embark or disembark the vehicle, the information on the smartcards should be read. When the doors close, the smartcard read/write cycle should end. The following tabulates the system events and the appropriate responses.

	Event	System Response
1	Program startup	a) RF controller, GPS receiver, PM-Minder, Psion Workabout power up b) Psion Workabout startup boot sequence initiated c) Driver queried for information
2	GPS Position	a) Trimble reports position b) Psion logs message to file
3	Door opens	a) Trimble unit sends door open message b) Psion logs message to file c) Psion asks RF controller for odometer reading d) Psion starts listening for odometer read e) Psion enables RF controller tag reading f) Psion listens for passenger embark and passenger disembarks
4	Door closes	g) Trimble unit sends door close message h) Psion logs message to file i) Psion disables RF controller card reading j) Psion listens for door open event
5	Odometer read	a) RF controller sends odometer data from PM-Minder b) Psion logs data to file c) Psion stops listening for odometer reading
6	Passenger embarks	a) RF controller reads card b) RF controller reports card data to Psion c) Psion logs card message to file
7	Passenger disembarks	a) RF controller reads card b) RF controller reports card data to Psion c) Psion logs card message to file

Notes:

- The data terminal should allow simple input by the driver to initialize recording.
- The GPS receiver reports position messages to data terminal every 2 (two) seconds.
- All the messages sent to the data terminal are stored in one file to simplify the parsing of the data into the database. The data terminal adds a time stamp, as reported by the data terminal clock, to every message it receives. The Psion will pre-pend each message it receives with an integer indicating its source.
- Data terminal will continuously record GPS messages and watch for door events reported from the Trimble unit
- The door open and close events are reported to the data terminal through the GPS receiver.
- When a door open event occurs, the distance from the last stop should be computed.
- The RF controller only reports tag reads between a door open and close event, to reduce probability of duplicate reads.

The Psion creates a Dbase III file that logs the events that occur during operation. This file is transferred to the database application and imported into the database tables.

3.3 Software Analysis Documentation

In this section, we discuss the results of the software analysis phase of the project. The purpose of analysis is to fully understand the problem, application domain, and system requirements in order to yield a precise model of the real world situation. The focus of analysis is on the problem domain not the design of implementation artifacts. Analysis should ignore, to the extent possible, the application and concentrate on the problem domain. We first present a description of the use cases identified for the application, and then present the class diagrams for the application.

3.3.1 Use Case Models

A **use case** is a statement of high-level functional system requirements in narrative text. Use cases serve as the basis for the software requirements and for setting the stage for the early identification of objects within the system. They set the stage for analysis and then are useful in developing dynamic and functional models. In addition, use cases serve as a basis for system test development.

Use cases should emphasize top-level functions and interfaces based on the viewpoint of an actor. An **actor** is a well-defined role for a user or group of users. Actors can be people or they may be other systems with which the system interacts. Use cases begin at a high level and can be refined by subsystem all the way down to the object level. A **use case model** is the statement of use, use case diagram, use case description, and actor description.

Actors: (The main users of the transit monitoring system)

Actor Name: Transit Patron
Description: A transit patron is a human SmartCard holder that uses the transit system to travel to a destination. A transit patron uses the SmartCard as a method of payment.

Actor Name: Driver
Description: A human user who operates the vehicle where the collection system is installed. This user will have minimal interaction with the system, but must have knowledge of how to start and turn off the equipment. Also, this actor will be responsible for transporting the data to and from the data compiling location.

Actor Name: Database Manager
Description: A human user who takes the data collected from the transit vehicles and places it into the database. This user must have knowledge of the validation process and the database.
Notes: The database manager can also be a transit manager

Actor Name: Transit Manager
Description: A human user who manages the day-to-day operations of the transit company.

Actor Name: Administrator
Description: A human user who helps make decisions regarding the transit company's future and strategies.

The following use case diagrams identify the main use cases for the three main needs described in the system concept section:

- "Advanced systems for transit vehicle location, identification, and management of transit systems."
- "Improved methods of data collection about or from transit users."
- "Advanced systems for fare collection and control systems."

For each of these three areas, high-level use case are shown and then further refined during design.

Objective 1: Vehicle location, identification, and management of transit systems

In this model, the use cases dealing with vehicle information and management are shown with their appropriate actors. The driver of the transit vehicle will be responsible for collecting data, which is done automatically by the installed hardware and software. After the data has been collected it is imported into a database for validation, viewing, and analysis. A person who understands the database structure and data validation, a database manager, will import the collected data into the database. Once the data is in the database transit managers and administrators can view the data and use it.

Use Case: Collect Data
Actors: Driver
Description: The actual collection of data is done automatically by the hardware and software installed on the transit vehicles. The driver drives the vehicle, which collects data as it moves along the route. The driver is responsible for making sure that the data collection devices are running and should mention any problems to a technician.

Use Case: Import Data
Actors: Database Manager
Description: Once the data has been collected it must be compiled into a database to be viewed and analyzed. As the data is being compiled it must be validated, calibrated, and summarized.

Use Case: View and Use Data
Actors: Transit Manager, Administrator
Description: Decision-makers and managers to help with day-to-day operations view the data in the database. Some uses might include making worker schedules and viewing route efficiency data.

Objective 2: Transit user data collection

In this use case scenario, the use cases dealing with transit user information are shown with their appropriate actors (figure 2). The transit patron must first initialize his account by giving personal information when purchasing a SmartCard. When a patron uses the transit system by riding one of the vehicles data is collected about the patron's trip. The driver is responsible for driving the transit patron and making sure that the data collection devices are running. After the data has been collected it is imported into a database for validation, viewing, and analysis. A person who understands the database structure and data validation, a database manager, will import the collected data into the database. Once the data is in the database transit managers and administrators can view the data and use it.

Use Case: Initialize Information
Actors: Transit Patron
Description: In order to collect data regarding transit patrons, initial information must be entered. When the patron purchases a SmartCard user information is given and initializes a file for later use.

Use Case: Collect Data
Actors: Transit Patron, Driver
Description: Passenger data is collected when a transit patron enters or leaves a transit vehicle. The hardware and software installed on the transit vehicle tallies the information automatically. The driver drives the vehicle, which collects data as it moves

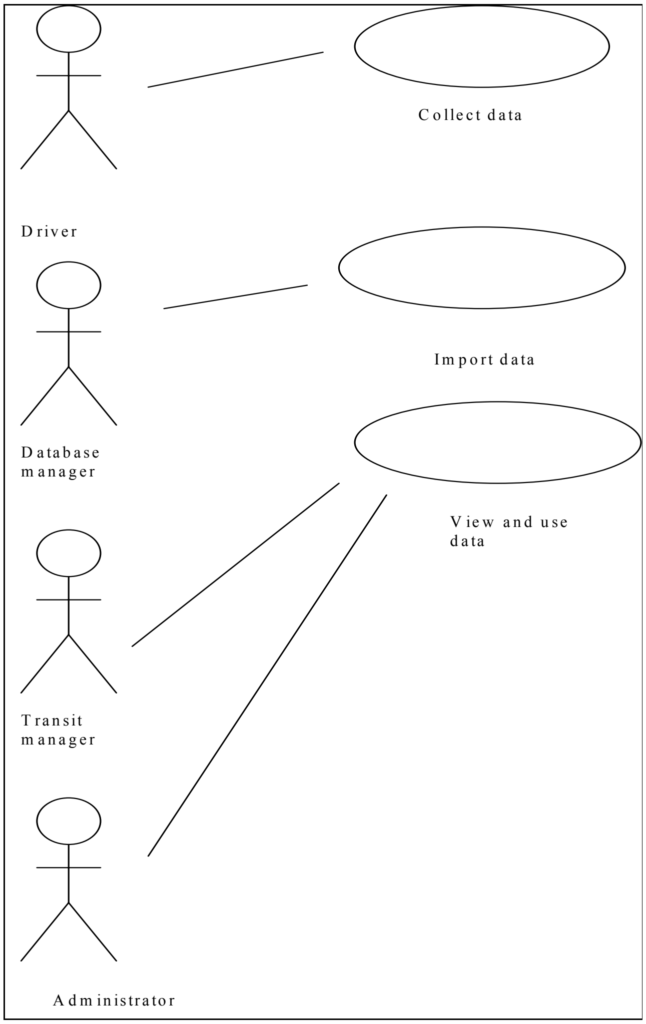


FIGURE 2 Objective 1 Use Case Scenario

along the route. The driver is responsible for making sure that the data collection devices are running and should mention any problems to a technician.

Objective 3: Fare collection and control

In this use case scenario, the use cases dealing with fare collection and control are shown with their appropriate actors. The transit patron must first initialize his account by giving personal information when purchasing a SmartCard. Also, the transit patron can view the fare data by viewing printed statements.

Use Case: View and Use Data
Actors: Transit Manager, Administrator, Transit Patron
Description: Decision-makers and managers to help with day-to-day operations view the data in the database. Some uses might include making worker schedules and viewing route efficiency data. A transit patron can also view and use the fare data to see the status of an account.

Each high-level use case is further refined in use case scenarios. Use case scenarios present more detail involving the interaction of the actors with the system. The use case scenarios provide guidance to the development and implementation of functionality from the perspective of the user.

3.3.2 Conceptual Model Class Diagrams

In this section, we present the system's conceptual class diagrams. These diagrams model the transit monitoring system. The different classes describe the components that make up a transit monitoring system. Some of these classes have attributes and/or operations listed inside of the class box. Other attribute/operation information for the sake of brevity has been omitted from the diagram. Collectively, the diagrams document the complete conceptual model. To better understand and explain the conceptual model, it is easier to split up the complete model into manageable distinct packages: data capture, importation, fare analysis, transit analysis, statistical analysis. The data capture package describes the program that is resident on the portable data terminal and functions as the on-vehicle data collector. The importation package handles the importation of the data collection files into the database from the file produced by the data capture package. The fare analysis package handles the data associated with the smart cards and their use as fare collection devices. The transit analysis package stores information about the transit system and information concerning its operations. The statistical package is a utility package used to assist in statistical analysis of the database.

3.3.2.1 Data Capture Package

The data capture package initializes the data collection, manages the communications between hardware components, and logs events that occur during transit. This program follows an event-driven structure where an event triggers an action. Figure 3 displays the various components of data capturing.

The VehicleMonitor acts as the central controller of the other pieces of hardware. It connects to each of the data transmitting devices when needed and reports Events that occur. An Event has a type, time stamp, and a message associated with it. For example, when a GPS coordinate is sent to the monitor, it is considered an event, and the coordinate information is logged as a message along with the time stamp and event type.

These Events are logged into an EventLog, which is a collection of the transit vehicle events. The EventLog must be initialized at the beginning of the data collection by using openLog(). Whenever a Event occurs, it is logged with logEvent().

The Placer455 object is a representation of the actual hardware unit. It is interfaced through a SerialPort and data is sent to and received from the Placer455. After starting the reporting of data points using startScheduledReporting(), messages are received and checked for Events. Some messages are from the Placer455's DigitalInputLine, which describe bus door actions (opening and closing).

As the transit vehicle drives along the route, its odometer reading is transferred to the VehicleMonitor through the SerialPort and is recorded as an Event.

When a door open event occurs it triggers the SmartCardReader to begin reading tags. The SmartCard identification number is read and is compared to the entries in the SmartCardCollection to see if the tag read represents someone who is boarding or alighting the transit vehicle. Based on that information the SmartCardCollection is updated and the corresponding event is logged to the EventLog.

The driver of the transit vehicle interacts with the system through the SystemScreen. On the screen the status of the monitoring system can be seen and the driver can choose whether to start or stop the data capturing. Upon starting the system the driver must enter route, vehicle, and driver information.

The data capture package creates a DBF file with the EventLog. This file can be taken out of the Psion Workabout and be imported into the main database.

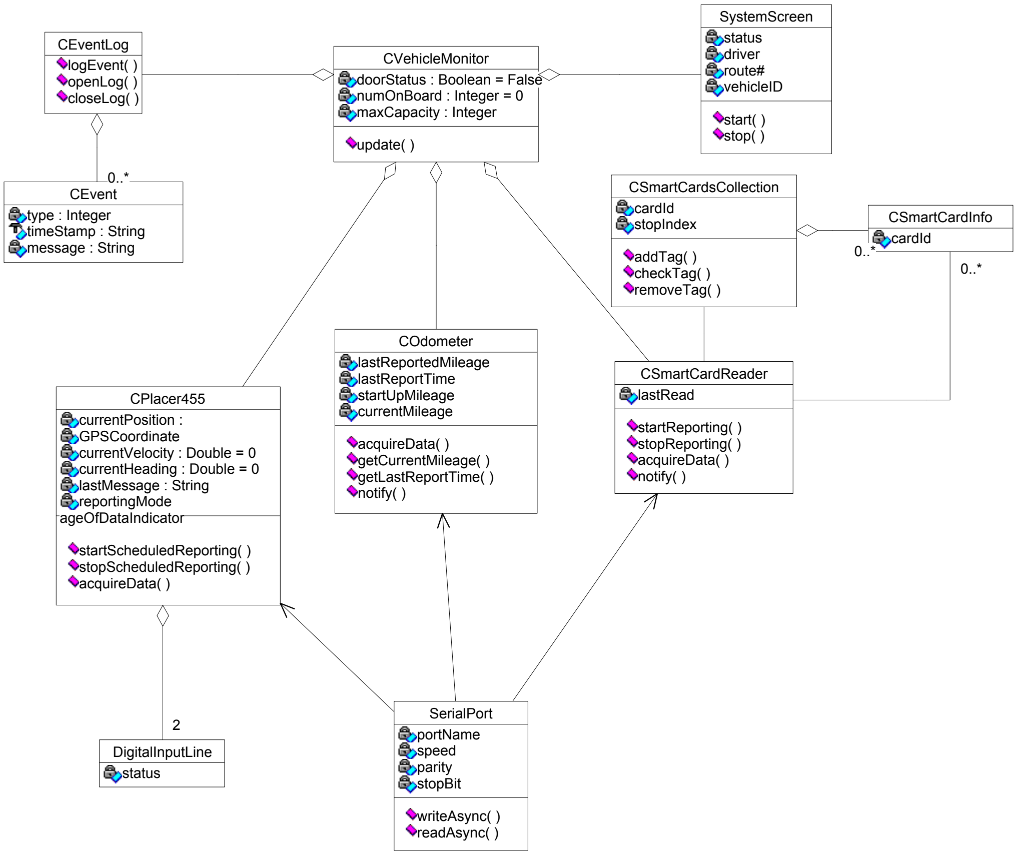


FIGURE 3 Data Capture Package Concept Diagram

3.3.2.2 Importation Package

The importation package takes the data captured during the operation of the vehicle along the route, processes the data, and imports the data into the main database. The importation package's structure mirrors the structure of the data capture package and the transit package. Figure 4 illustrates the structure of the importation package.

The EventLog represents the Dbase III file created by the data capture package. The EventLog class is responsible for linking to the Dbase III file and recording that the data has been committed to the database. An EventLog contains TIMSEvents that are processed and sent to the Vehicle for handling. The Vehicle class represents a vehicle that has been in operation. The Vehicle responds to startup, update position, update odometer, open door, close door, pick up passenger, and drop off passenger events. A vehicle contains a GPSPositionRecorder that is responsible for recording the GPS position messages. A GPSPositionRecorder contains two collections of GPSPositions. GPSPositions is a standard collection class, but also allows for the collection of statistical averages (TallyStatistic and TimeStatistic) on the GPSPosition data that it contains. The two GPSPositions (duringDwell and duringTraversal) represent the GPSPosition's that were recorded during a dwell at a bus stop or during a traversal from one bus stop to another bus stop. A GPSPosition contains the information sent by the GPS device and includes the latitude, longitude, elevation, velocity, heading and time stamp of the GPS message.

The vehicle also contains three collections of SmartCards. The SmartCards class is a standard collection class for storing SmartCard objects. The vehicle keeps track of SmardCard objects that are boarding the vehicle, that are already on-board the vehicle, and those that are exiting the vehicle. The vehicle contains an Odometer for recording the distance traveled. The vehicle also contains a TraversalRecorder. The Traversal Recorder is responsible for recording the route, point, and link traversals that occur as a vehicle moves along a route in the transit network. The TraversalRecorder has a current RouteTraversal and contains a set of completed RouteTraversal objects. A RouteTraversal mirrors a Route in structure. A RouteTraversal represents an instance of the vehicle traveling along the entire route. Just as a Route contains an ordered collection of Point objects, a RouteTraversal contains an ordered collection of PointTraversal objects and an ordered collection of LinkTraversal objects. A PointTraversal is an instance of a vehicle dwelling at a Point in the Route network. A LinkTraversal is an

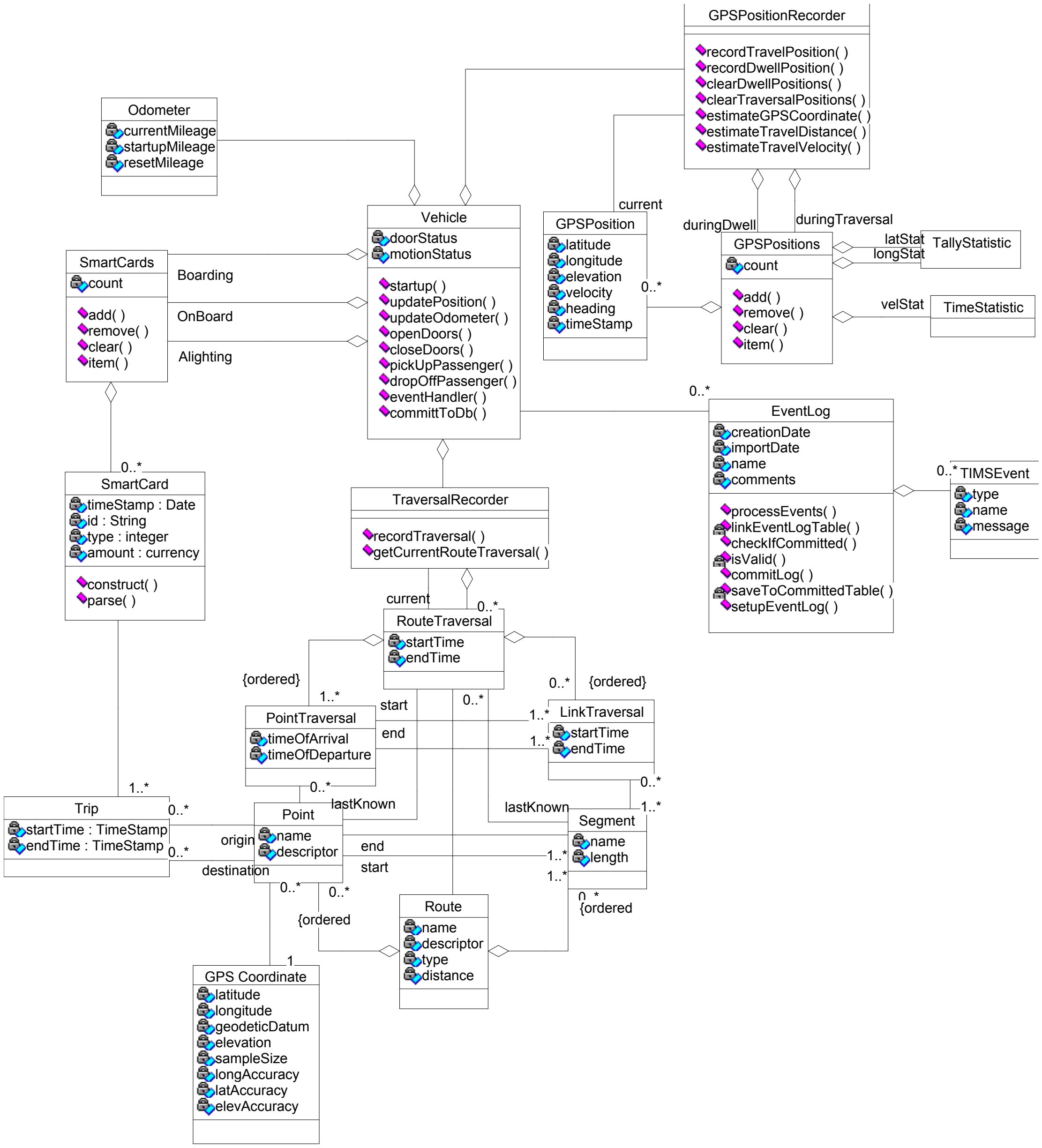


FIGURE 4: Importation Package Concept Diagram

instance of a vehicle moving from a starting PointTraversal to and ending PointTraversal. A LinkTraversal may be associated with many Segment objects.

A Segment is a connection between two Point objects on a Route. A Segment may be associated with many LinkTraversal objects. The two point objects for a segment represent its starting and ending points in the route. A Route contains many ordered points and segments that are related to each other. Each Point has a GPSCoordinate that locates it within the transit network. A Point acts as the origin for many Trip objects or as the destination for many Trip objects. A Trip represents the passenger's use of the transit system via a SmartCard. Each Trip utilizes only one SmartCard, but a SmartCard may be associated with many Trips.

The importation package must interface with the other classes used in the database. The main packages that it interacts with are the fare analysis and transit analysis packages. Within Figure 4 the structure of the RouteTraversal, PointTraversal, LinkTraversal, Route classes etc. are repeated. The classes represented within the importation package allow for the creation of objects that represent the traversals, etc. that occurred during the data capture process. These classes then are responsible for committing this data to the database in accordance with the structure outlined by the fare analysis and transit analysis packages.

3.3.2.3 Fare Analysis Package

The fare analysis package (figure 5) is responsible for managing the fare and accounting information when the smartcards are used as fare collection devices. A trip is generated via the importation package and is associated with a smartcard. One SmartCard can be used for zero to many trips, and each trip is associated with one SmartCard. A fare is associated with each trip and card. A fare is a type of transaction that can be associated with an account. Payment transactions can also be associated with an account.

A SmartCard is purchased at a card outlet and some information is tallied at the time of the transaction. Some information about the purchase can be recorded. In addition, when a person opens an account or purchases a debit card, demographics information may be recorded to provide statistics for market research. Each person may or may not have demographic information stored about them. Demographic information may be recorded without an associated person.

A SmartCard can be either a debit card or a credit card. A debit card has a balance set by the user and acts like a self-contained checking account. When the passenger uses the SmartCard on the transit vehicle the debit balance decreases by the amount of the fare for the trip. The credit card SmartCard acts like a normal credit card. The transit patron is credited for trips taken and billed at regular intervals.

An account records the transactions and allows statements to be issued. An account can have more than one credit type SmartCard associated with it. A person acting as an account holder may have an account. An account holder can authorize other people to use the cards in his/her account. Each account has a mailing address. A statement summarizes account activity over a period of time.

Each person may or may not have a mailing address. Each mailing address is associated with one or more persons. For example, two members of a household might have a SmartCard account.

3.3.2.4 Transit Analysis Package

The transit analysis package (figure 6) is responsible for storing the structure of the transit system (e.g. routes, schedules, vehicles, etc.) In addition, data recorded during the operation of the system is also recorded (e.g. trips, traversals, event logs, etc.)

A trip is made up of a start time, end time, and is related to an origin and a destination point. The origin and destination are different points along a bus route. A route consists of ordered sets of Points and Segments. Segments are logical links between two points along a particular route. A segment can be associated with the roads that physically connect two bus stops. A point can serve as a start or end point for a segment. A point can be an AccessPoint, where a passenger can get on or off the

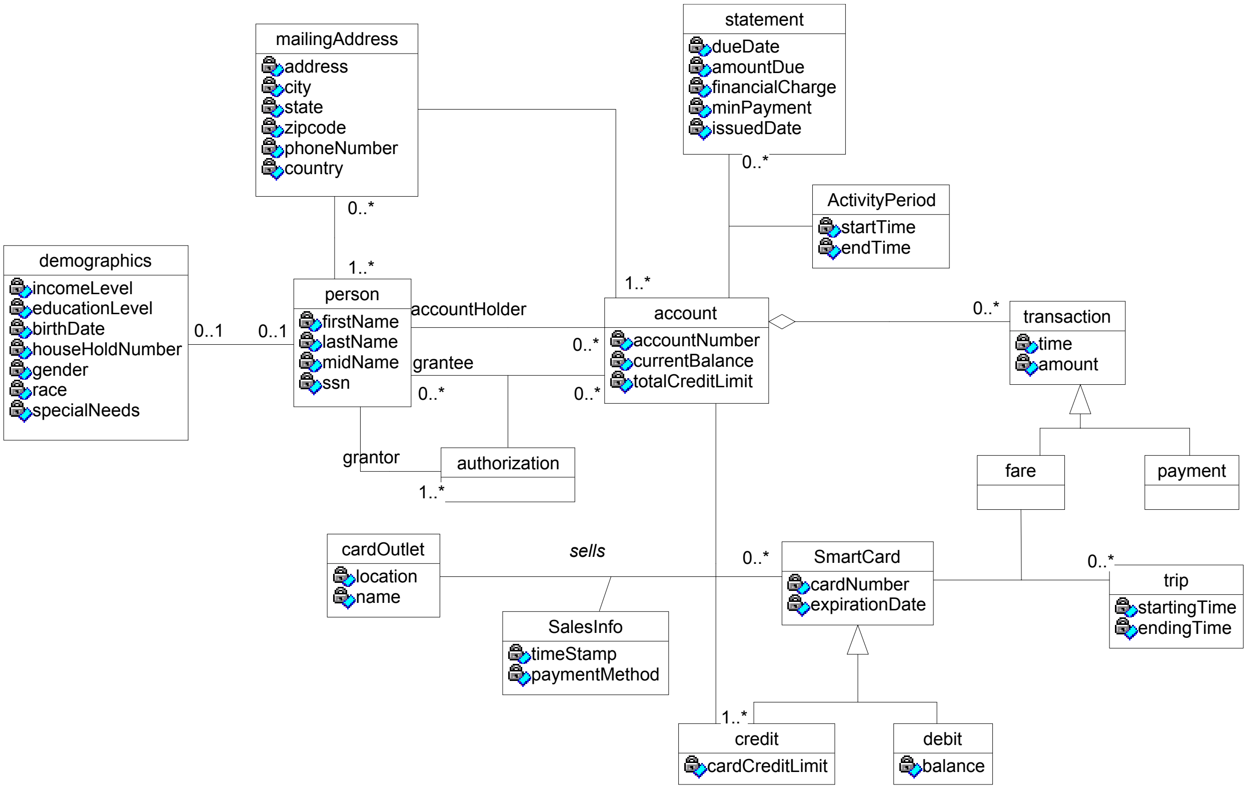


FIGURE 5: Fare Analysis Package Concept Diagram

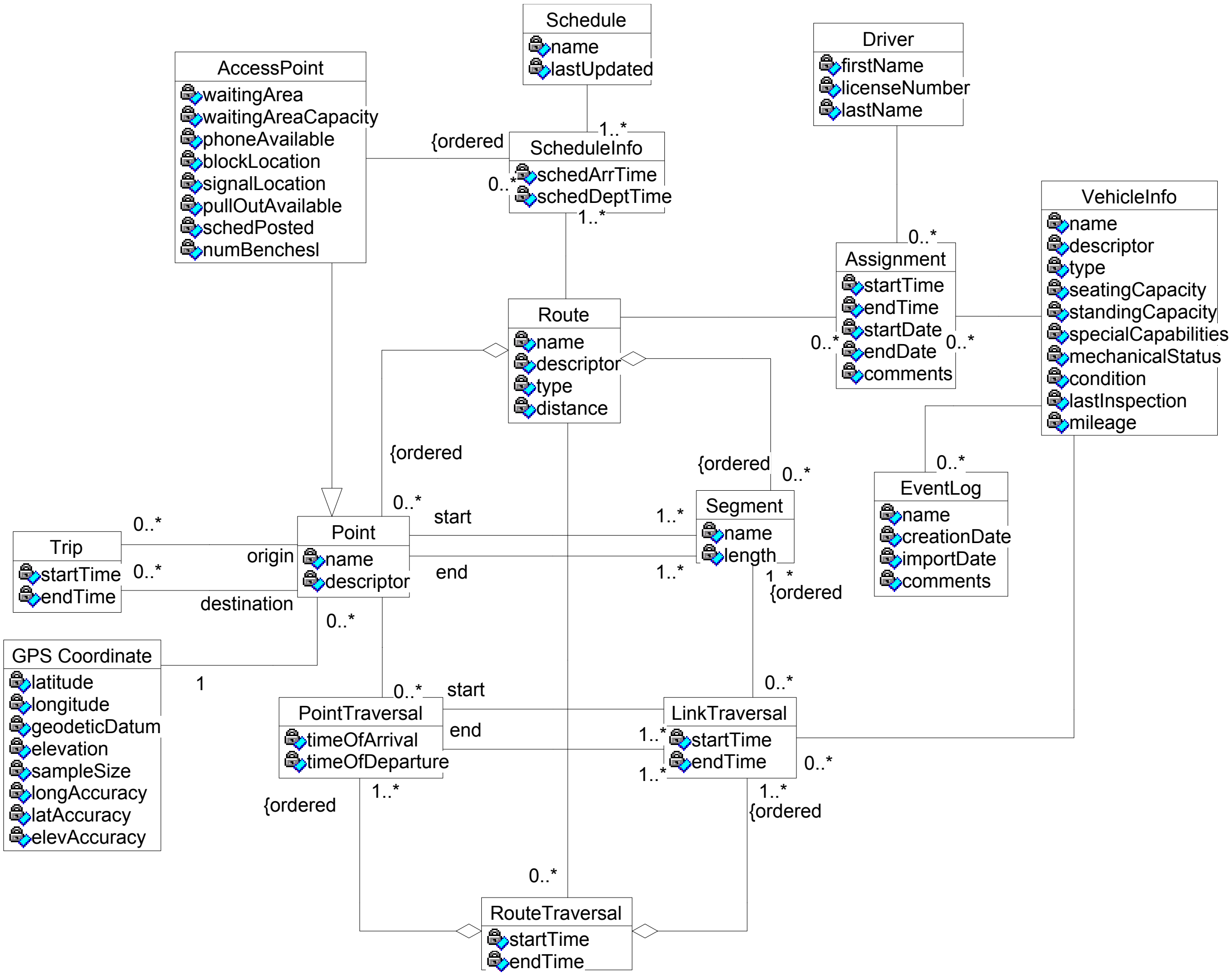


FIGURE 6: Transit Analysis Package Concept Diagram

bus. An AccessPoint can be a bus stop or a drop-off point requested by the passenger. A point also has a GPS coordinate, which describes its location using longitude and latitude coordinates.

Groups of ordered AccessPoints appear with schedule information. Schedule information (ScheduleInfo) represents the linkage between a route, access points, and a schedule. ScheduleInfo has a scheduled arrival time and a scheduled departure time. These are the times when a bus is expected to arrive and depart from that location. A schedule is linked to a particular route through the schedule information. This allows a route to have one or more schedules because schedules can change with the season, day of week, and time of day, etc.

As a transit vehicle travels along a route it moves along a set of adjacent segments causing a LinkTraversal. One vehicle can make zero to many LinkTraversals, and a LinkTraversal is made by one vehicle. A group of LinkTraversals makes up a RouteTraversal, which is one complete cycle of a route. A LinkTraversal is related to many segments. A PointTraversal represents the dwell at a Point. A route has a vehicle and a driver assigned to it. The assignment has a start time, end time and dates. A vehicle is driven by a driver and can be driven by many different drivers during operation. Also, a driver can drive multiple buses. The driver is assigned to the bus by a duty designated by a manager. An event log is captured for each vehicle.

3.3.2.5 Statistical Analysis Package

Statistics about the transit service are kept in the SummaryStatistic object, which tallies statistics for the traversals and Trips. For each summary statistic, there is a time range that must be entered to make the statistic. Each trip may have many summary statistics generated and each summary statistic may be associated with many trips. Each traversal may have many summary statistics generated and each summary statistic may be associated with many traversals. Figure 7 is a diagram of the statistical analysis class.

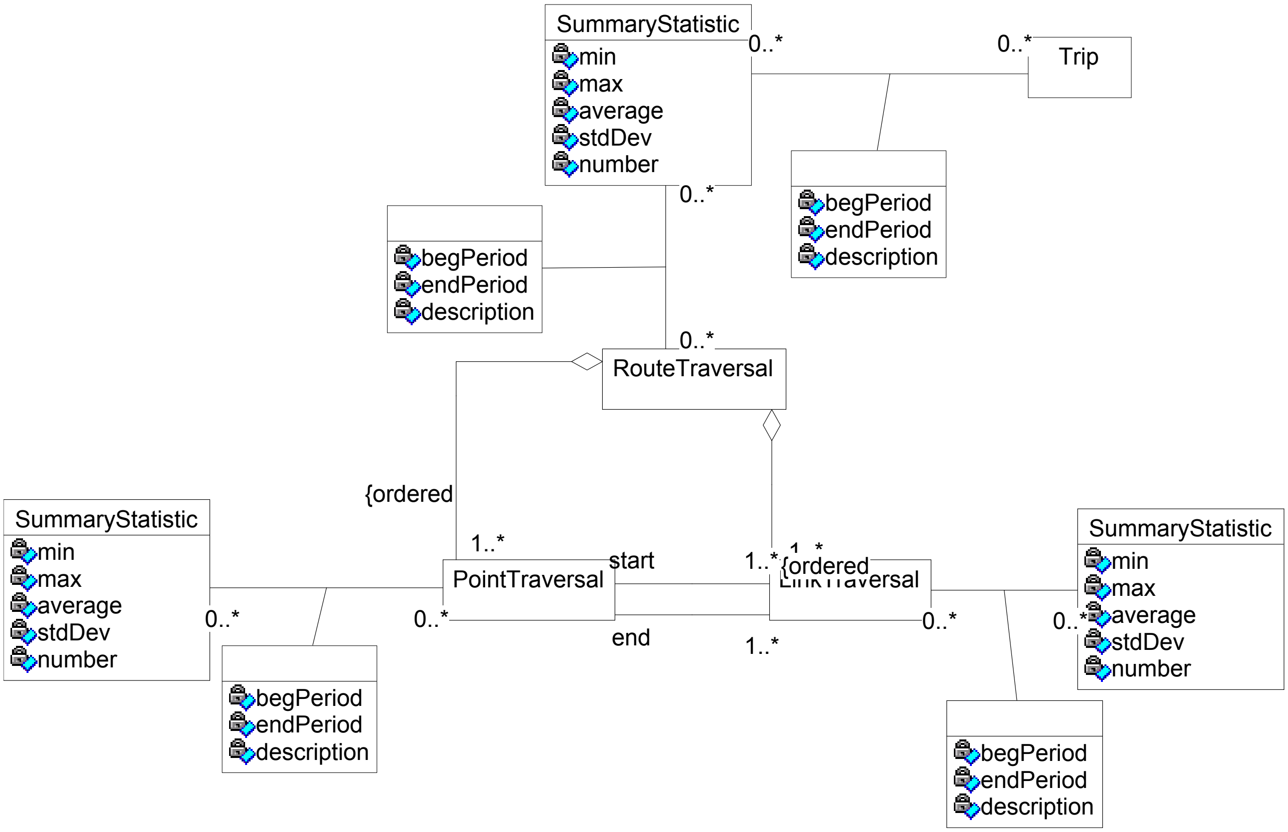


FIGURE 7: Statistical Analysis Class Diagram

3.4 Software Design Documentation

This section presents detailed design documentation associated with the database design. We present the specifics of implementing the database using the relational database paradigm. This includes the implementation of object identity, the implementation of data domains and constraints, and finally the mapping of the relevant class diagrams to relational tables. We then present the design of the user-interface including its structure and important design features.

3.4.1 Database Implementation

The TIMS conceptual model includes persistent data that must be stored over time. The primary classes involved are those presented in the transit analysis package, fare analysis package, and the statistical analysis package. In database design, the key issues involved in translating from the object model to the relation model involve the implementation of object identity, the specification of domains, and the mapping of the object model to relations. For this process, we followed the guidelines given in reference *(2)*.

3.4.1.1 Identity Implementation

Primary keys are used to address the rows within a table. Each table should have a primary key unless the table is a temporary table used for intermediate calculations. Foreign keys are used to implement associations and generalizations. Conceptually, objects have a unique identity, but practically a computer implementation is required to guarantee identity. In this application existence-based identity will be used to implement identity:

Existence-Based Identity

1. Adds an object identifier to each class and make it the primary key
2. Uses the internal support for identifiers from the RBMS, for example use the AutoNumber domain in MS Access
3. Recommended for applications with more than 30 classes
4. Eliminates the difficulties associated with compound primary keys
5. Simplifies implementation of associations via foreign keys
6. Other candidate keys must still be enforced via the use of unique indexes.

3.4.1.2 Domain Specifications

A domain refers to the set of possible values for an attribute. The specification of domains allows the values of an attribute to be represented by data types within the implementation language. For more information on domains, see reference *(2)*.

3.4.1.3 Class To Table Mapping

In the relational model, objects must be represented as tables (i.e. as rows and columns). The basic strategy is to begin by mapping each class to a separate table. In addition, additional attribute columns for existence based identity, buried associations, and generalization discriminators are added as necessary to implement the relationships in the model. This process is detailed in Chapter 13 of reference *(2)*.

For the most part, the mapping of the transit analysis package follows the guidelines given in Chapter 13 of reference *(2)*. The ScheduleInfo class is a transformation from a ternary association involving AccessPoints, Routes, and Schedules. The Assignment class is also a transformation from a ternary association involving Route, VehicleInfo and Driver. The ordered constraints, for example Routes and Segments) were handled by implementing a sequence number attribute. The generalization of Point and AccessPoint involved implementing a table for each class and adding a type enumerator to the Point table. The model depends heavily upon the concept of a traversal. A traversal is in essence an activity that occurs between two points in space and time. The route, point, and link traversal classes could be generalized to an abstract traversal class, but this was not done for simplicity. This generalization would make the statistical analysis package simpler by avoiding the duplication of attributes within the multiple tables.

For the fare analysis package the transaction generalization (transaction, fare and payment) was not implemented as three separate tables. Instead, each subclass was mapped to a table and the common attributes duplicated within the tables. This makes the implementing the Trip to SmartCard association involving a fare simpler in the relational model.

3.4.1.4 Referential Integrity Specifications

In the relational model, a foreign key represents a reference to a row in another table containing the matching candidate key value. The problem of ensuring that the database does not include any invalid foreign key values is called the referential integrity problem.

1. A row may not be added to a table with a foreign key unless the referenced value exists in the referenced table.
2. If the value in a table that is referenced by a foreign key is changed (or the entire row is deleted), the rows with the foreign key must not be “orphaned”.

Three Integrity Options

1. Disallow: The change is disallowed by forbidding the deletion/update operation of the referent row if there exists referencing rows.
2. Cascade: For updates, the change is cascaded to all dependent tables. For deletions, the rows in all dependent tables are deleted.
3. Nullify. For updates/deletions, the dependent foreign key values are set to null (if allowed).

Since we have chosen existence-based identity, we do not need to consider the update of foreign keys because the identifiers are artificial numbers that do not need updating.

3.4.1.5 Overview of Database Queries

Because a database is an organized repository of data, queries concerning the data are available to investigate the relationships between the data. For TIMS the major objectives included answering questions concerning vehicle operations, transit users, and fare collection. For each of these objectives, we defined an example list of possible queries or operations on the data.

Objective 1: Vehicle location, identification, and management of transit systems

1. Vehicle Management
   a) Find the mileage for a vehicle
   b) Find when the Last Inspection was for a vehicle
   c) Find all vehicles made in specified year
   d) Find all vehicles used this week
   e) Find all vehicles that are in the shop
   f) Find the operating cost per revenue vehicle-mile
   g) View revenue miles per vehicle
   h) View total passengers per vehicle
   i) Show the routes taken by a certain vehicle

2. Driver Management
   a) View this week’s work schedule
   b) View a driver’s past duties
   c) View the hours worked by a specific driver
   d) View the revenue vehicle-hours per employee

3. Route Efficiency
   a) View the route times for a route
   b) View the number of riders during a day
   c) View total passengers per route
   d) View revenue passengers per service area population
   e) View percentage of population served
   f) View system reliability

Objective 2: Transit user data collection

1. Patron Trip Info
   a) Find all patrons who have a disability
   b) Find all patrons who live in a particular city
   c) Find the average number of trips made per person
   d) View routes taken by a specific person
2. Patron Statistics
   a) Find all patrons who use the transit service 8 times a week
   b) Find the average number trips for the patrons from a particular area of town (zip code)
3. Demographic Info
   a) Find the ethnicity breakdown for the riders of a route
   b) Find the income levels for a particular bus stop

Objective 3: Fare collection and control

1. Mail statements
   a) Create statements for all customers for trip for the past month
   b) Create statements for all customers with outstanding debts
2. Update Balances
   a) Find all patrons who have submitted fare payment this month
   b) Find all patrons who have charges over $10.00 this week
3. View Financial Statistics
   a) Find the average fare charges for patrons last week
   b) Find the route with the most fare charges

3.4.2 User-Interface Implementation

The material in this section is base in part on reference *(3)*. The TIMS database design focuses on ease-of-use in order for the application to be an effective tool for transit managers. The resulting interface allows users to interact with the application with minimal training and effort. This section explains the design criteria that were used throughout the development and gives a detailed description of the user interface.

3.4.2.1 Graphical User Interface Design Protocols

The TIMS graphical user interface was designed using criteria compiled from the latest user interface research. These protocols were compiled into the following design criteria:

1. Geometrical and Movement Arguments
   a) Use large targets so users can easily use mice pointers.
   b) Set up the screen to facilitate short mouse movements.
   c) Minimize switching between the mouse and the keyboard.
2. Memory Arguments
   a) Design menus that do not require the user to recall information.
   b) Present help information in conjunction with the task.
   c) Structure the interface so users do not have to rely on prior experience with the system to use it.
   d) Include shortcuts that give users the option to bypass application events.
3. Problem Solving Arguments
   a) Label choices so users can associate an option with a solution to a task.
   b) Clearly indicate that an action has actually been accomplished by the system.
   c) Allow actions to be easily reversible.
4. Attention Arguments
   a) Present information with a big change in the display.
   b) Present information close to where the user is already looking.
   c) Use auditory signals as well as visual signals.
   d) Design the layout so that there is an initial focus on every screen that draws in the user and then directs them to secondary areas of information.
5. Cluster Controls
   a) Organize the screen into visually separate blocks, each containing similar controls (elements which behave differently should be in different areas).
   b) Title each block.
   c) Group similar commands on the same menu (all the buttons concerning a similar function should be grouped together).
6. Visibility
   a) Make frequently used controls obvious and easy to access.
   b) Hide less used controls.
   c) Make features on the screen consistent with what the user would find in the real world.
7. Consistency
   a) Design a single screen and repeat that design throughout the interface.
8. Color and Fonts
   a) Use color sparingly - only to emphasize information.
   b) Once a color scheme has been established, it should be used throughout the entire application in the same way.
   c) Red and green should only be used in the center of the screen, never in the periphery.

- d) Blue, black, white, and yellow should be used in the periphery.
- e) Use less than three fonts and three sizes.
- f) If multiple fonts are used they should be associated with different classes of information.
- g) Use a maximum of 40 - 60 characters per line of text.
- h) Text should be flush left and numbers flush right.
- i) Upper and lower case characters should be used as much as possible.

9. Screen Organization
   - a) Only the pertinent features should be on the screen - with similar controls grouped together.
   - b) Users will be looking at this screen over and over again, so the amount of information on it should not be overbearing.
   - c) Each screen should have a clear and consistent structure.

3.4.2.2 Introductory Windows Interface Design

When the TIMS application is opened, the user is greeted with a series of welcome screens. These introductory windows are similar to one another in order to maintain a consistent look and feel across the application. These windows are light gray with black type (Arial font), with the exception of the logo, which is in color. Each window is composed of five components in addition to the logo, as shown in Figure 8.

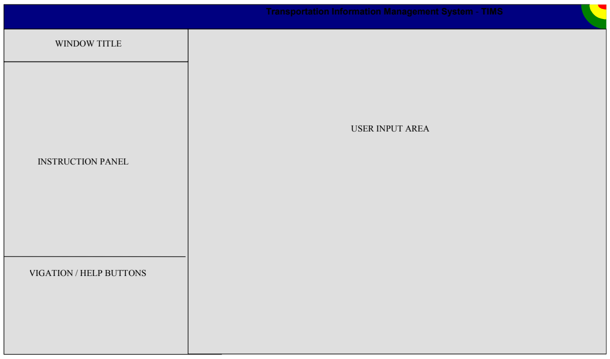


FIGURE 8 Introductory Window Design

System Title Bar

A dark blue panel occupies the top of each window. The text reads "Transportation Information Management System."

Window Title Bar

This area contains the title of the current window. Examples include: Main Menu, Table Menu, Report Menu, etc.

Instruction Pane

This panel contains hints, instructions, and help regarding the current window. If users request additional information concerning a term or database element, they may hit the "Help" button, which will open up a window with additional information.

User Input Area

This area houses the interactive controls, which allow for communication between the user and the system. For details on what features are found in this area, see *the Introductory Screen Detailed Functionality* section of the report.

Navigation Buttons

Three buttons are located in this block: Back, Help, and Quit.

TIMS Logo

This logo appears on all the windows, as well as on the reports and other output from the system. The orange wedge rests on top of a yellow wedge, and the yellow wedge on top of a green wedge. These wedges represent the radio signals that relay information between the radio frequency identification tags and the antenna.

This window structure will be used for the introductory windows presented to the user. The more specific areas of the database, including tables, reports, etc., will have different user interface designs depending on the nature of the information being presented. These designs are described in further sections of the report.

3.4.2.3 Introductory Windows Flow Chart

The TIMS application introductory screen order is represented by Figure 9. Each of these windows is designed based upon the Introductory Window Design specifications. Each of these screens prompts the user for information that is stored in the database. Only database administrators will have the option to go into the "System Maintenance" window. The unique features of each window are described next.

3.4.2.4 Introductory Windows Detailed Functionality

Each of the introductory windows shown in the previous flow chart exhibits unique TIMS functionality. The descriptions below illustrate how the user must navigate through these initial screens in order to have access to the transit data.

Window #1: User Identification

This window provides users with general information regarding how the database works, what they should expect to see in the different areas of the database, how to access help, etc. The user is requested to choose a log-in

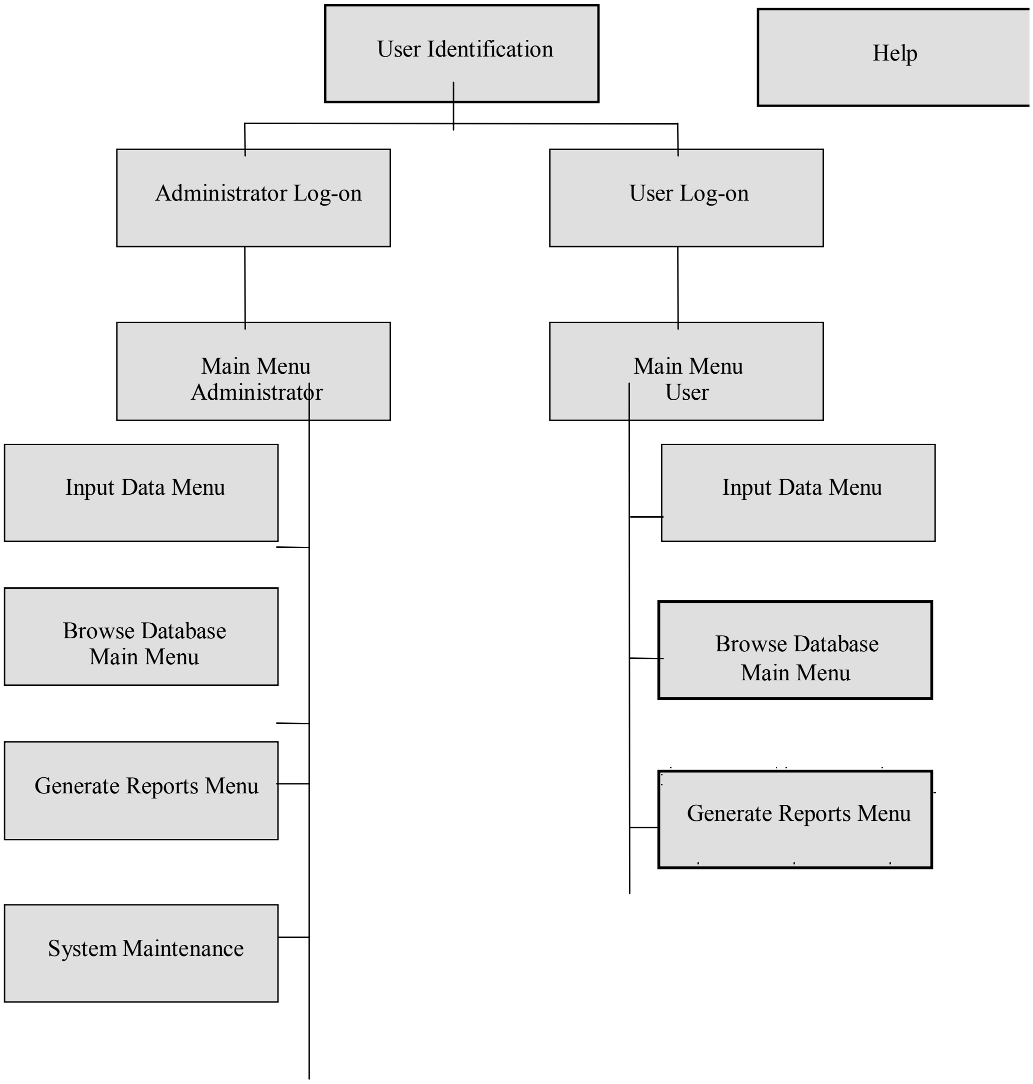


FIGURE 9 - Introductory Windows Flow Chart

type. The user may choose system administrator, transit personnel, accountant, or operations supervisor.

Window #2: Log-In
This window prompts the user to enter in a log-in password/identification.

Window #3: Main Menu
TIMS users are presented with one of two main menus depending upon their user category. Transit managers, operational supervisors, and accountants will be able to browse the database, enter raw data, and generate reports. In addition to the options present to the primary users, the Systems Administrator may choose a feature that allows him/her to perform database maintenance (add/delete tables, change the layout of the forms, etc.).

Browse Database Main Menu
This menu is composed of buttons that represent the various tables making up the database. By clicking on these buttons, users are able to access the bulk of the information contained in the database.

Input Data Menu
This window presents users with instructions on how to download data from a disk and have that data entered into the database.

Generate Reports Menu
Various reports may be generated from the transit data. This menu will contain buttons describing the various options available to the user. The *User Input Area* for this window will be similar to the *Browse Database* window. The various reports available are listed in the Generate Reports Windows Flow Chart.

System Maintenance
System Administrators may need to go through the database and change various elements of the tables and relations. Administrators may also get into these various elements using this control.

Help
Each screen contains a button that opens up a help window. The help windows contain information regarding the various functions of the database and provide users with information regarding application problems that may develop.

3.4.2.5 Browse Database Windows Interface Design

Once users have gone through the opening screens, they can choose to browse the database. Users may run queries, filter data, and manipulate various data from the table windows. Each of the tables that the user can browse will be presented as a button in the initial "Browse Database" window. The browse windows will be light gray with black type (Arial font), with the exception of the logo, which will be colored. Each window will be composed of 4 components, as shown in Figure 10, in addition to the title bar.

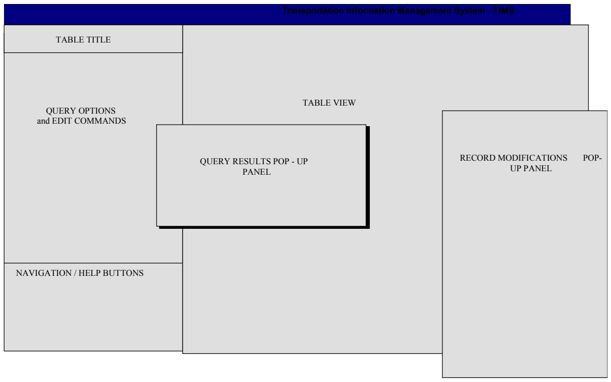

Transportation Information Management System - TIMS
TABLE TITLE
QUERY OPTIONS
and EDIT COMMANDS
TABLE VIEW
QUERY RESULTS POP - UP
PANEL
RECORD MODIFICATIONS POP-
UP PANEL
NAVIGATION / HELP BUTTONS

FIGURE 10 Browse Database Window Design

Table Title Bar
This area contains the title of the current Table being viewed.

Query Options and Edit Commands Pane
This panel contains the query and edit buttons available to the user. The results from the query appear in a Query Results pop up window.

Table View
The tables are displayed in this area. If the tables are too large to display in this space, users may scroll in all four directions in order to view all the elements in the table.

Navigation Buttons
Three buttons are located in this block: Back, Help, and Quit.

Query Results Window
A separate window appears with the results of a query.

Modify Data
In the event that the user would like to modify a record, a separate window will appear with options such as: delete record, add record, and modify record.

3.4.2.6 Browse Database Windows Flow Chart

The Browse database main menu allows users to choose from three information areas with each of these areas further sub-divided. The three areas correspond to the three main associated with the application. Each of these windows was designed using the Browse Database Design Structure. Figure 11 represents the Browse Database window structure. Each individual table window has functions supporting the queries and filters associated with that table.

3.4.2.7 Browse Database Windows Detailed Functionality

The following section describes preliminary query options and other functionality associated with the browse database windows. For a listing of what columns will make up each table, refer to the *Database Schema* section of the report. Each table supports insert, change, and delete functionality for users with appropriate access. Both the edit buttons and the query buttons described below are located in the *Navigation Button* area of the window.

Bus Stop
After selecting a specific bus stop id, the transaction query will show each passenger transaction that occurred at that bus stop.

Trips
This menu allows the analysis of trips. For example, the list of trips that meet the following fare classifications will be displayed.

- Children 5 and under
- Half Fare/Senior Citizen
- Para-Transit passenger
- Free Rider
- Mall/Agency Pass
- Approved Short Fare
- Full Fare

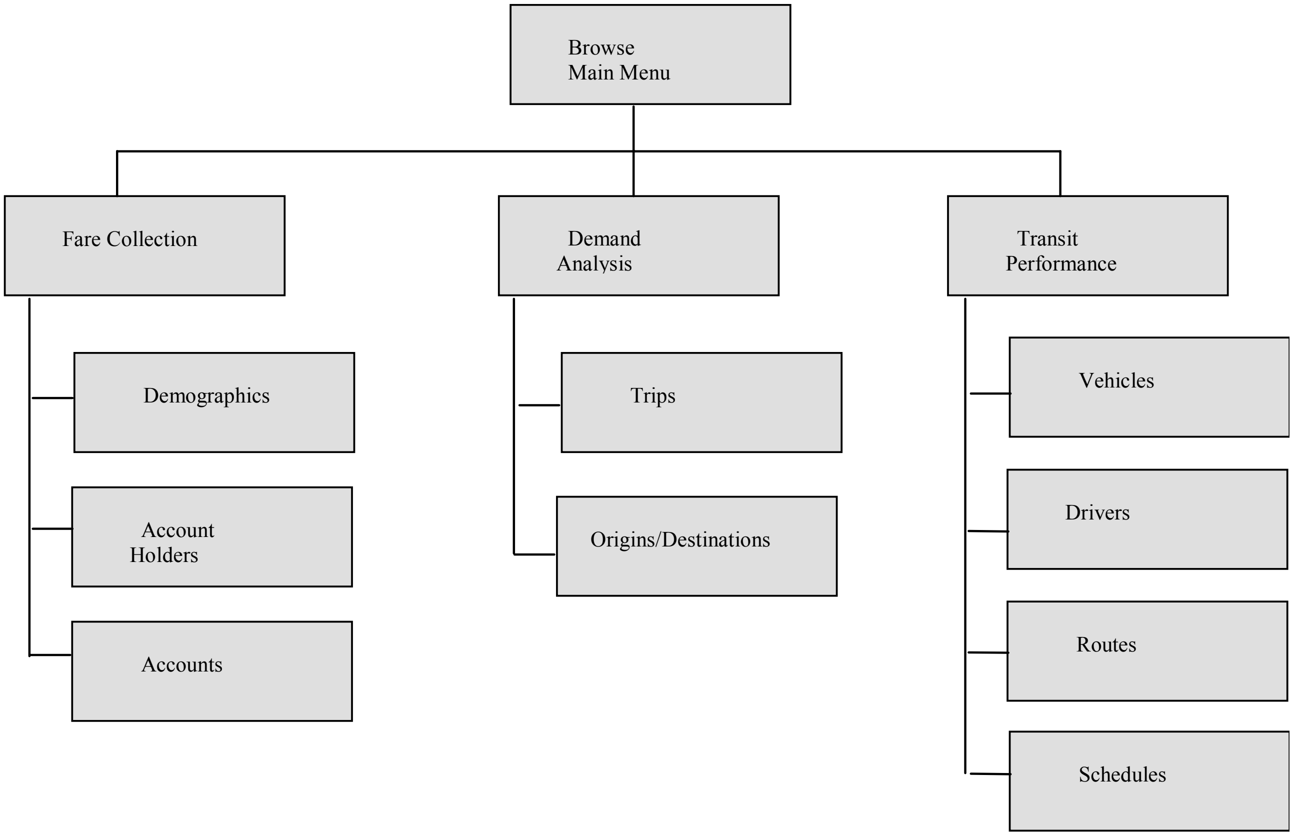


FIGURE 11 Browse Database Flow Chart

Origin/Destination
This menu supports the analysis of origin/destination and bus stop access. Users may enter a date range, and the query will return the origin/destination information for each pair of bus stops for the specified time period. This table should support the following additional queries:

1. list trip origins for bus stop during date range
2. list trip destinations for bus stops during date range
3. list trips for particular transit user

Vehicles
Users may perform three queries based on the data in the vehicle and its associated tables:

1. determine vehicle mileage per day by vehicle id
2. list each trip associated with the vehicle by vehicle id
3. determine which vehicle was associated with a segment traversal on a given day

Drivers
The driver form contains two query options:

1. list driver hours by date and driver id
2. list all duty information for a particular driver

Routes
There are three queries on this form based upon the selection of a route id:

1. list the number of passengers riding a particular route for a given date
2. list the vehicle mileage per route per day
3. list segment characteristics of the selected route
4. list trips associated with a particular segment

Schedules
Once schedule form is chosen the following queries are available:

1. list schedule information
2. list schedule adherence data

3.4.2.8 Generate Reports Windows Interface Design

Reports and graphs are the polished, tangible products from the system. The TIMS application automatically pulls the requested data from the underlying tables to product the reports. The graphical user interface for these windows is based upon the Introductory Window Design. Users may generate reports in each of the three areas, fare collection, demand analysis, and transit performance.

3.4.2.9 Generate Report Windows Detailed Functionality

The report windows contain the same navigation buttons (Back, Help, and Exit) that are available in the browse windows. The instructional panel on the left-hand side of each report window describes the information that will be presented on the generated report. The user must select the data's time period (month and year) data for the report. The various reports are described below under the window name. Each report lists the data for four time periods: the month selected, the same month last year, this year, and last year (unless otherwise indicated.) Example reports are given below. A graph of the data may be generated for each report as well. Figure 12 shows a flow chart of the report window functions.

Vehicle Mileage

1. *Bus Mileage* shows mileage based upon vehicle id.
2. *Bus Cost per Mile* shows cost based upon vehicle id.

Route Mileage

1. *Monthly Data* shows the monthly mileage per route based upon route id
2. *Daily Data* shows the daily mileage per route based upon route id.

Schedule Adherence

The schedule adherence report window prompts the user to specify a specific day in addition to a month and year. The application also requires the user to select a route, and a specific time period. The time period options include:

- 9:00 AM - 11:00 AM: Morning Block
- 11:00 PM - 2:00 PM: Mid-day Block
- 2:00 PM - 5:00 PM: Late afternoon Block
- 8:00 AM - 5:00 PM: Full Day

The report breaks down the number of vehicle arrivals at each bus stop based upon the following time categories:

- Totals Arrivals
- Over 5 Minutes Early
- 2 - 5 Minutes Early
- 0 - 2 Minutes Early
- On Time
- 2 - 5 Minutes Late
- 5 - 9 Minutes Late
- Over 10 Minutes Late

Origin/Destination Profile

The user must select an originating bus stop for this report, in addition to a month and year. The report lists each destination, along with the total number of passengers for that origin-destination pair.

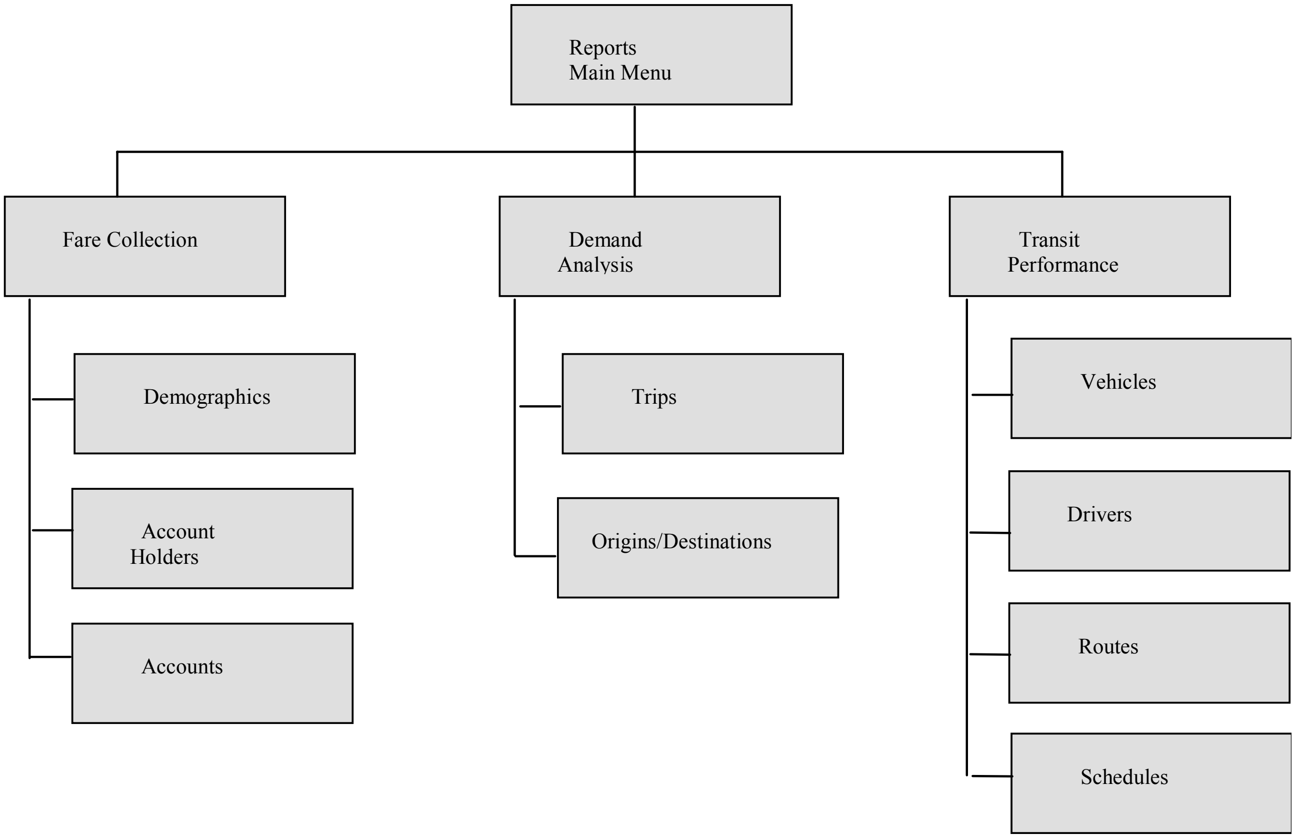


FIGURE 12 Report Windows Flow Chart

4. INVESTIGATOR PROFILE

Manuel D. Rossetti, Assistant Professor
Department of Systems Engineering
University of Virginia
Thornton Hall
Charlottesville, VA 22903

Phone: (804) 924-4438
Fax: (804) 982-2972
E-mail: mr2m@virginia.edu

Education: Dr. Rossetti received his MS and Ph.D. degrees in Industrial and Systems Engineering from The Ohio State University in 1988 and 1992 respectively, and his BS in Industrial Engineering from the University of Cincinnati in 1985.

Experience: Dr. Rossetti joined the faculty of the Department of Systems Engineering in 1993. He is responsible for research and teaching in the areas of automatic data collection systems, manufacturing systems, transportation systems, discrete event simulation, stochastic modeling and vehicle routing and scheduling.

Selected publications:

1. Rossetti M. D. and Clark G. M.(1995) "Estimating the Expected Sojourn Times for an Exiting Markov Renewal Process," *Communications in Statistics: Theory and Methods*, Vol. 24, No. 2, pp. 553-579.

2. Benton W. C. and Rossetti M. D.(1992) "The Vehicle Scheduling Problem with Intermittent Customer Demands," *Computers and Operations Research*, Vol. 19, No. 6, pp. 521-531.

3. Rossetti, M. D. and Clark G. M. (1998) "Estimating Capacity Loadings From Arrival and Departure Events", *Journal of Manufacturing Systems*, Vol. 17, No. 1, 1998, pp. 65-76.

4. Rossetti, M. D. and Turitto, T. (1998) "Comparing Static and Dynamic Threshold Based Control Strategies", to appear in *Transportation Research Part A*

5. Rossetti, M. D. and Turitto, T. (1997) "Design of an Integrated Transit Monitoring System based on RF/ID" to appear in *International Journal of Technology Management.*

5. REFERENCES

1. Transportation Research Board Project Announcement TRANSIT-IDEA, Transit Cooperative Research Program (TCRP) Innovations Deserving Exploratory Analysis (IDEA). May 1993.

2. Blaha, M., and W. Premerlani. 1998. *Object-Oriented Modeling and Design for Database Applications*, Prentice-Hall.

3. Sharma, S. 1997. *Database Re-Engineering and User Interface Design for a Transportation Information Management System,* B.S. Thesis, University of Virginia, School of Engineering and Applied Science, Charlottesville, VA 22903

4. "Global Positioning System Overview." http://wwwhost.cc.utexas.edu/depts/grg/gcraft/notes/gps/gps.html

5. Haggart, Craig. "How GPS Works: An Introduction." http://ati83.ati.ac.at/Public/GPS/files/gpsworks.html

6. Hurn, Jeff. Differential GPS Explained. Trimble Navigation Limited, 1993.

7. Hurn, Jeff. GPS A Guide to the Next Utility. Trimble Navigation Limited, 1989.

8. Kennedy, Michael. The Global Positioning System and GIS. Ann Arbor Press, Inc., 1996.