

BUPT Taxi-Hailing Management System

Software Requirements Specification

3.0

2019.11.20

Hang Pan

Requirements Analysis Engineer

Revision History

Date	Description	Author	Comments
2019.11.12	1.0	Hang Pan	Create an overall document framework; Complete the first and second parts.
2019.11.16	2.0	Hang Pan	Add multiple graphical descriptions of functional requirements; Update the text description of the functional requirements.
2019.11.20	3.0	Hang Pan	Fulfill non-functional requirements; Complete requirement change management; Format, index, check and proofread

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Your Name>	Lead Software Eng.	

Table of Contents

REVISION HISTORY	II
DOCUMENT APPROVAL.....	II
1. INTRODUCTION	1
1.1 PURPOSE.....	1
1.2 SCOPE	2
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	2
1.4 REFERENCES.....	2
1.5 OVERVIEW.....	3
2. GENERAL DESCRIPTION.....	3
2.1 PRODUCT PERSPECTIVE	3
2.2 PRODUCT FUNCTIONS	3
2.3 USER CHARACTERISTICS	4
2.4 GENERAL CONSTRAINTS	5
2.5 ASSUMPTIONS AND DEPENDENCIES	6
2.5.1 Assumptions.....	6
2.5.2 Dependencies	6
3. FUNCTIONAL REQUIREMENTS.....	6
3.1 CONTEXT ANALYSIS	7
3.1.1 Context Diagram	7
3.1.2 Data flow level 1.....	8
3.1.3 Data flow level 2.....	9
3.2 FUNCTIONAL REQUIREMENTS	11
3.2.1 Functional Structure.....	11
3.2.2 User Interaction Module.....	12
3.2.3 Driver Interaction Module	13
3.2.4 Navigation Module	13
3.2.5 Online Payment Module.....	14
3.2.6 Inspector Interaction Module.....	14
3.2.7 Administrator Interaction Module.....	15
3.2.8 Intelligent Analysis Module.....	15
3.2.9 Database Module.....	16
3.3 USE CASES	16
3.3.1 User Role	16
3.3.2 Use Case #1 Driver	17
3.3.2 Use Case #2 Taxi-Hailing User.....	18
3.3.2 Use Case #3 Business Inspector	19
3.3.2 Use Case #4 System Administrator.....	20
3.4 CLASSES / OBJECTS.....	20
3.4.1 Driver Class	20
3.4.2 Taxi-Hailing User Class	21
3.4.3 Business Inspector Class.....	22
3.4.4 System Administrator Class.....	22
3.4.5 Navigation System Class.....	23
3.4.6 Online Payment System Class	23
3.4.7 System Database Class	23
3.4.8 Intelligent Analysis System Class	24
4. NON-FUNCTIONAL REQUIREMENTS	24

4.1 QUALITY REQUIREMENTS.....	24
4.1.1 <i>Performance</i>	24
4.1.2 <i>Reliability</i>	25
4.1.3 <i>Availability</i>	25
4.1.4 <i>Security</i>	26
4.1.5 <i>Maintainability</i>	26
4.1.6 <i>Portability</i>	26
4.2 ENGINEERING REQUIREMENTS.....	26
4.2.1 <i>Inverse Requirements</i>	26
4.2.2 DESIGN CONSTRAINTS.....	27
4.2.3 LOGICAL DATABASE REQUIREMENTS.....	27
4.3 QUALITY REQUIREMENTS.....	28
5. CHANGE MANAGEMENT PROCESS.....	28
A. APPENDICES	30
A.1 APPENDIX 1	30

1. Introduction

This document is a requirement report for BUPT Taxi-Hailing Management System. The documentation is more comprehensive. The system interacts with eight roles: Drivers, Taxi-Hailing Users, Business Inspector, System Administrator, Navigation System, Online Payment System, System Database and Intelligent Analysis System. At the same time, the traditional graphical modeling method is used. The system structure, data flow diagram, functional structure diagram and system entity-relational diagram are used to analyze the structure and function of the system and the relationship between users. At the same time, the functional and non-functional requirements of the system were initially proposed. I hope that this document will facilitate the development of this system. Here are my plans and explanations.

1.1 Purpose

- 1) Deepen the understanding of the system, elaborate on the function, performance and service design of BUPT Taxi-Hailing Management System, providing a complete specification.
- 2) Set the corresponding requirements for Drivers, Taxi-Hailing Users, Business Inspector, System Administrator, etc., and analyze the relationship between the roles, and strive to be consistent.
- 3) Describe the mission of the implementation of BUPT Taxi-Hailing Management System to help customers judge whether the specified system meets the requirements and how to modify them to meet their requirements.
- 4) It can be used as a reference document for product design and product detailed function design. On the one hand, it provides the basis for the detailed design and subsequent programming development of the open side; on the other hand, it provides the tester with the basis for acceptance.
- 5) Provide formal expressions to allow exchange of views between parties, providing a basis for agreement between end users, clients and developers.
- 6) Make the parties reach the consensus as much as possible on the final specification and reach a consensus.
- 7) Improve development efficiency. Detailed and detailed specifications can reduce rework activities that are redesigned, recoded and retested afterwards. It is easy to find some omissions, misunderstandings and inconsistencies in the early stages of development, so as to correct them in time.
- 8) Provide a basis for system costing and scheduling.
- 9) As a reliable basis for continuous improvement and improvement of the system.

This specification applies to:

- Users: Understand the functionality and performance of the intended project and discuss and negotiate the entire need with the developer.
- Developers: Use this document to understand the capabilities of the intended project and to design and develop the system accordingly.

- Testers: Perform functional and non-functional testing of software products in accordance with this document.
- Other related personnel: such as user document writers, project managers, etc.

When reading this document, you should first understand the functional overview of the product, and then you can properly understand each function according to your needs.

1.2 Scope

The software system requirements analyzed in this document are as follows:

- 1) The name of the software system is: BUPT Taxi-Hailing Management System, BUPT-THMS.
- 2) The system is suitable for the following situations:
 - The performance of BUPT Taxi-Hailing Management System is minimum for 500,000 taxi-hailing users and 10,000 drivers.
 - The system should keep the basic information of the users intact and consistent.
 - The system guarantees security, does not reveal user information, and does not allow users without specific permissions to change information.
 - DBMS recommended by the system: MySQL.
 - This document should be consistent with other documentation such as system requirements documents, software project requirements documents, and so on.
- 3) This system is not suitable for the following situations:
 - This system belongs to the software system under the Web platform, so it cannot be used in the environment without a computer network.
- 4) The system is for Drivers, Taxi-Hailing Users, Business Inspector and System Administrator to use.

1.3 Definitions, Acronyms, and Abbreviations

Form 1-1 Term and Definition

<i>Term</i>	<i>Definition</i>
BUPT-THMS	BUPT Taxi-Hailing Management System
PIM	Personal Information Management
PI	Personal Information
IM	Information Management

1.4 References

- [1] 王安生. 软件工程化[M]. 北京: 清华大学出版社. 2014.
- [2] IEEE Software Engineering Standards Committee. IEEE SA 830-1998, IEEE Recommended Practice for Software Requirements Specifications. October 20, 1998.

1.5 Overview

This document will analyze the requirements of BUPT Taxi-Hailing Management System through four parts:

- General Description: This chapter includes product perspective, product functions, user characteristics, general constraints and assumptions and dependencies.
- Functional Requirements: Several schematic modeling methods are used in this chapter. Vividly explain the functional requirements of BUPT-THMS.
- Non-Functional Requirements: Introduce the system requirements from another perspective.
- Change Management Process: Use swim lane flowchart to illustrate the correct flow of requirements changes.

2. General Description

This chapter focuses on the overall analysis of the project's expectations, functions, and other factors that have an impact on system development.

2.1 Product Perspective

The purpose of this system is to manage BUPT-THMS, as well as some other administrative parts, such as personal information management, bill management, and accident management.

- After receiving the external input, the system responds accordingly, ensuring that the taxi user can take the taxi accurately and quickly, and the driver can grab the order nearby.
- The taxi-hailing user information, driver information and order information are stored in the database server. The system ensures that taxi-hailing users and drivers can accurately and quickly view historical order information and personal information. When taxi users make legal additions, deletions and changes to personal information, the system will update the database source in time to ensure data consistency.
- In order to ensure the security of the system, the system sets different permissions for the three user roles of Drivers, Taxi-Hailing Users, Business Inspector and System Administrator. Users are divided into corresponding permission operations based on roles. The system administrator has the highest authority on the system.

2.2 Product Functions

BUPT-THMS can realize the high-speed, real-time and accurate transmission of taxi-hailing information through information technology. The achievable functions are as follows:

1) Drivers

- Register and log in to his or her account.
- Get nearby orders as much as possible.
- Check and edit personal information and vehicle information.
- Check order information and account information.

- Check taxi-hailing user information and contact him or her.
 - View current location.
 - Get driving directions and route.
- 2) Taxi-Hailing Users
 - Register and log in to his or her account.
 - Create and pay for orders.
 - Check and edit personal information.
 - Check order information and account information.
 - Check driver information and contact him or her.
 - View current location.
 - 3) Business Inspector
 - Register and log in to his or her account.
 - Monitoring the system.
 - Handle unexpected situations.
 - Publish information.
 - 4) System Administrator
 - Register and log in to his or her account.
 - Manage information, such as adding, deleting, changing, and checking information.
 - Maintain the system in real time.
 - Update the system regularly.
 - 5) Navigation System
 - Manage location information.
 - 6) Online Payment System
 - Generate orders online.
 - Manage online transactions.
 - 7) System Database
 - Process taxi-hailing user information.
 - Process driver and vehicle information.
 - Process order information.
 - 8) Intelligent Analysis System
 - Analysis information.
 - Process orders.
 - Analysis and plan driving routes.
 - Analysis unexpected situations and decide whether to inform the police.

2.3 User Characteristics

1) Users

During the morning peak and the late peak period, a large number of users access the system at the same time, which requires the system to have certain concurrency, so that the system can maintain normal use even when the concurrent access volume exceeds the range.

- Taxi-Hailing Users
 - a) Student Group: The way to receive confidence is more diverse and easier to receive new things. So, it is easier for students to try our software and it is our first users. Based on the student's financial situation, we can design a carpooling function for them. At the same time, students generally have experience in using taxi-hailing system and are familiar with similar information management systems. Therefore, the system does not need to provide students with detailed instructions for use.
 - b) Working Group: Because of the work, the demand for taxis is relatively large, and it is our main user. They have higher requirements for speed and efficiency of taxis, so we can design functions such as adding money to taxis and applying for driving.
 - c) Elderly Group: Not easy to receive new things. So, the page design should be easy to use. At the same time, we can design a function for one-click car and voice-call car for them. The system needs to provide them with detailed instructions for use.
- Drivers
 - a) Regular Taxi Driver: Have an operating license, security is guaranteed.
 - b) Private Car Driver: Irregular without a business license, cannot reach passengers in certain locations.
- 2) Developers

System developers will also have deviations during the development process. They need to continuously adjust according to the requirements analysis documents, such as modifying, refining, and adding corresponding functional modules to ensure that the developed system meets the requirements of the client and the actual needs of the end users.
- 3) Testers

Perform functional and non-functional testing of software products in accordance with this document.
- 4) Maintenance staff

Order information may also change as user information changes over time. Maintenance personnel are required to have basic information about the users in the system. Due to the huge amount of data generated each month, maintenance personnel need to perform regular backups of the database, and error recovery can also be performed when an error occurs.

2.4 General Constraints

- 1) Basic project constraints: The performance of BUPT Taxi-Hailing Management System is minimum for 500,000 taxi-hailing users and 10,000 drivers.
- 2) Technical limitations of the project:
 - The commonly used management system adopts the B/S development mode, and the system should adopt the basic technology of website development.
 - Systems with Web servers deployed on top of Linux systems generally have high operational stability.

- The configuration of the server system hardware can support efficient and stable operation of the server.
- In order to achieve future scalability of the system, the system needs to minimize the use of hardware. The entire system should also minimize calls between modules and try to loosely couple them.
- Data files and system configuration files should be stored securely and reliably.
- The system's data needs to be reliable enough to keep the system up and running.

2.5 Assumptions and Dependencies

2.5.1 Assumptions

- 1) The system is mainly responsible for managing BUPT-THMS, as well as other management components such as personal information management, bill management and incident management. Some other features, such as insurance business, are not included.
- 2) The software development team provides documentation for the corresponding development phase, and the user provides appropriate industry standards to combine software development with typical case assessment.
- 3) The operator and the user must operate the system in accordance with the operating procedures and must not perform malicious destructive operations.
- 4) The user must provide an effective database interface standard for the relevant running software, and notify the software developer in time during the change process to ensure that the budget and final parameters are correctly read and the cost budget is made.

2.5.2 Dependencies

The establishment of user information in this system depends on the public security system and the communication carrier system. If a user does not exist in the public security system and the communication carrier system, he or she cannot exist in the system.

3. Functional Requirements

The system is served to: Drivers, Taxi-Hailing Users, Business Inspector and System Administrator. External entities: Drivers, Taxi-Hailing Users, Business Inspector, System Administrator, Navigation System, Online Payment System, System Database and Intelligent Analysis System (Other external entities: The Police and the government). In this chapter, the user requirements mentioned in Chapter 2 will be presented in a more accurate, clearer, more complete, and unambiguous form, making this specification a basis for subsequent development and testing.

3.1 Context Analysis

3.1.1 Context Diagram

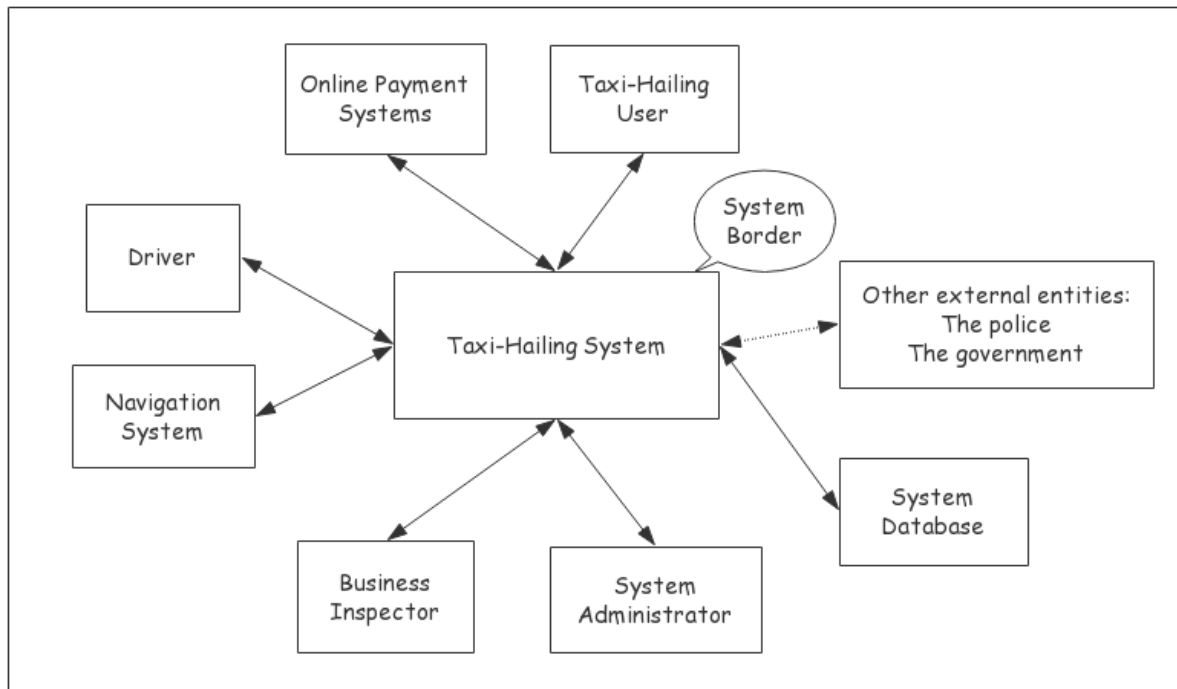


Figure 3-1 BUPT-THMS system context diagram

The role of the system context diagram is to distinguish the boundary between the system and the surrounding environment. Diagram 3-1 illustrates the interaction and interface relationship between BUPT-THMS and the external system. The external system mainly includes eight entities: Drivers, Taxi-Hailing Users, Business Inspector, System Administrator, Navigation System, Online Payment System, System Database and Intelligent Analysis System (Other external entities: The Police and the government).

3.1.2 Data flow level 1

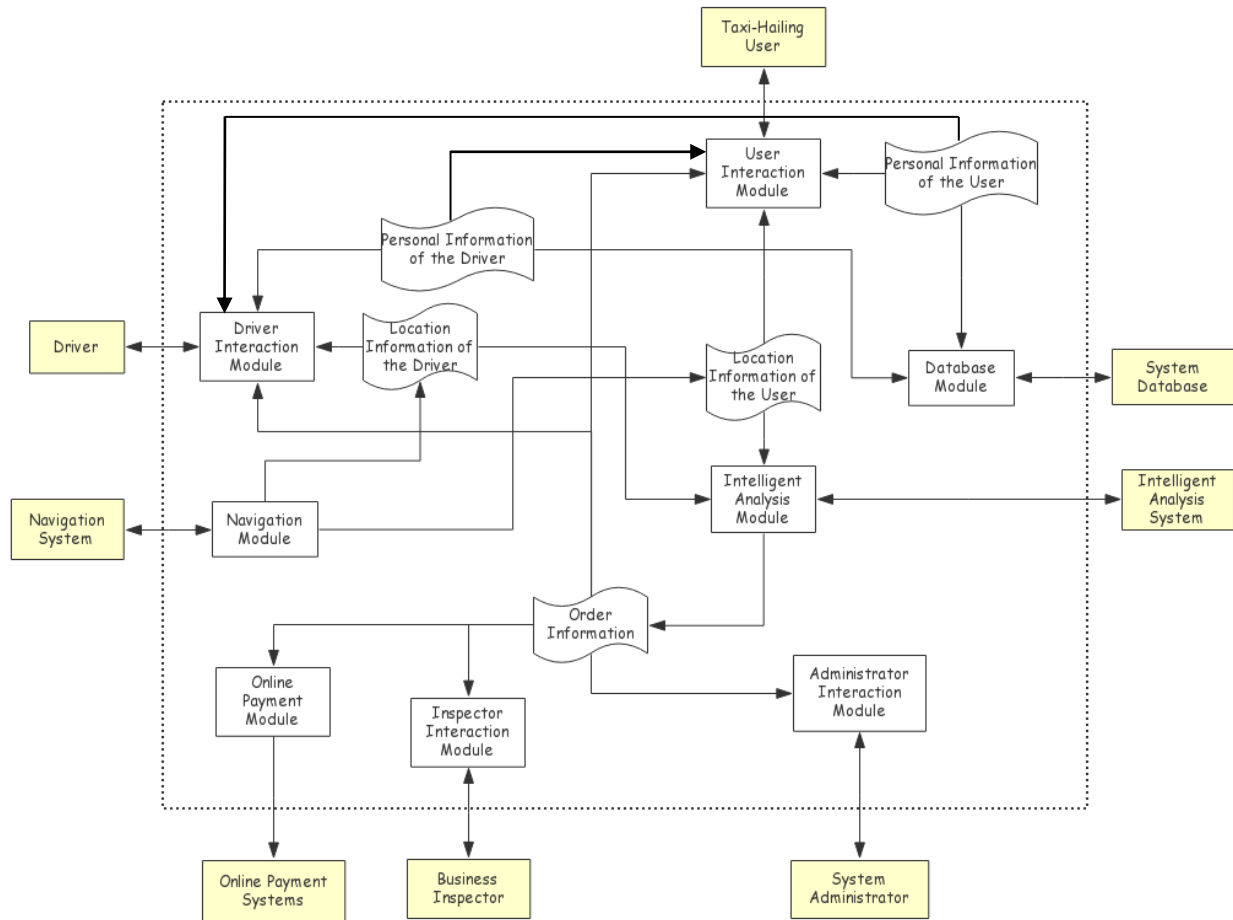


Figure 3-2 BUPT-THMS preliminary system conception

The internal functions of the BUPT-THMS are subdivided into the first level to form Figure 3-2. In Figure 3-2, its internal functions are refined into eight function modules: “User Interaction Module”, “Driver Interaction Module”, “Navigation Module”, “Online Payment Module”, “Inspector Interaction Module”, “Administrator Interaction Module”, “Intelligent Analysis Module” and “Database Module”. The main design ideas are as follows.

The data storage status of the internal system can be divided into five:

- 1) Personal Information of the User: Contains taxi-hailing user information for logging in to the system currently.
- 2) Personal Information of the Driver: Contains driver information for logging in to the system currently.
- 3) Location Information of the User: Contains location information of the taxi-hailing user for logging in to the system currently.

- 4) Location Information of the User: Contains location information and route information of the driver for logging in to the system currently. The computer can generate navigation routes based on an intelligent analysis system optimization.
- 5) Order Information: Contains current order information.

3.1.3 Data flow level 2

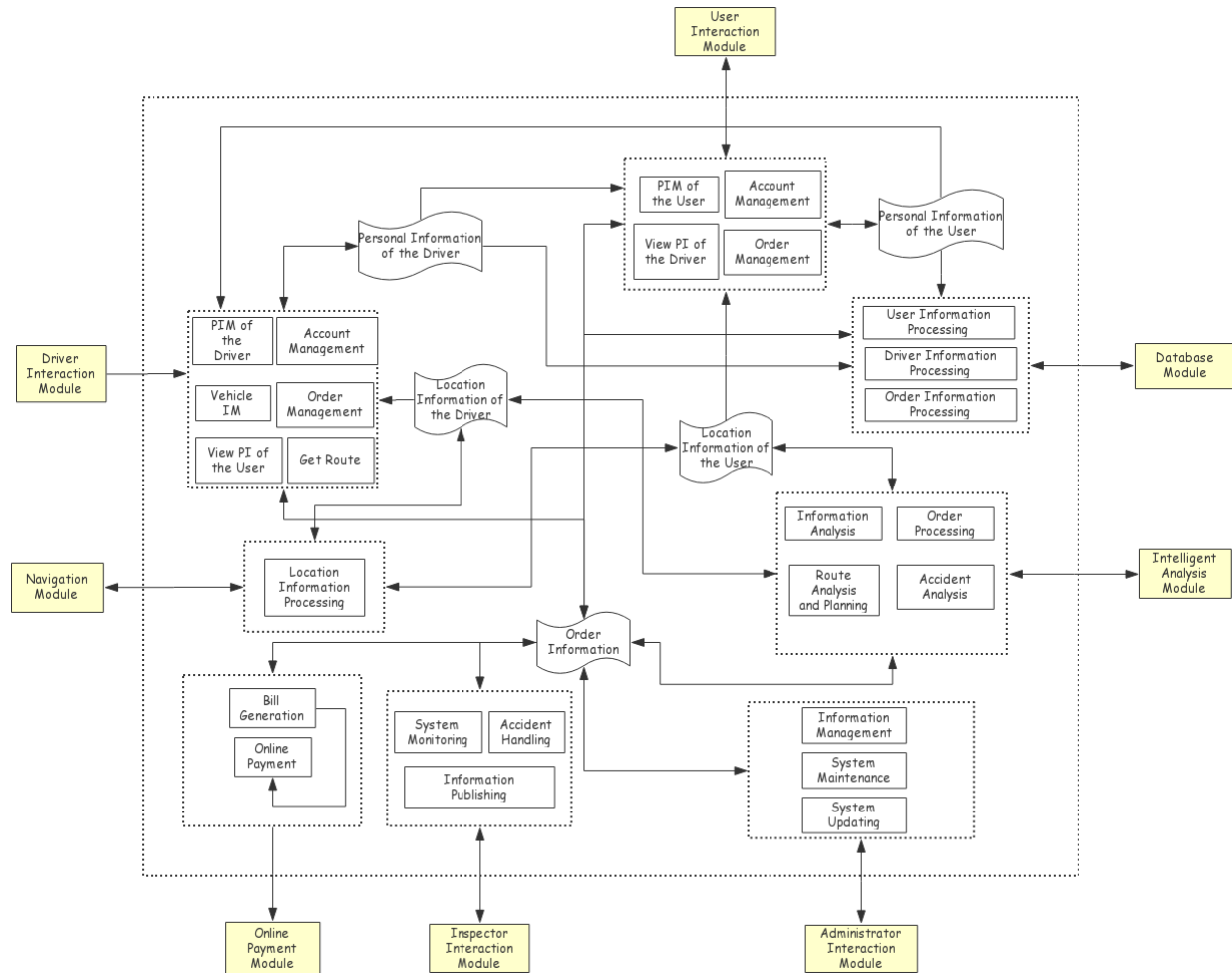


Figure 3-3 BUPT-THMS more detailed system conception

The internal functions of the BUPT-THMS are subdivided into the second level to form Figure 3-3. In Figure 3-3, The internal functions of “User Interaction Module”, “Driver Interaction Module”, “Navigation Module”, “Online Payment Module”, “Inspector Interaction Module”, “Administrator Interaction Module”, “Intelligent Analysis Module” and “Database Module”, are subdivided. The main design ideas are as follows.

- 1) In User Interaction Module, taxi-hailing user first logs in to the system, then he or she can: create and pay for orders; check and edit personal information; check order information and account information; check driver information and contact him or her; view current location.

- 2) In Driver Interaction Module, driver first logs in to the system, then he or she can: get nearby orders as much as possible; check and edit personal information and vehicle information; check order information and account information; check taxi-hailing user information and contact him or her; view current location; get driving directions and route.
- 3) In Navigation Module, the system can give the taxi-hailing user and driver positioning information according to the GPS.
- 4) In Online Payment Module, the system can: generate orders online; manage online transactions; update the account of taxi-hailing user and driver based on online transaction information.
- 5) In Inspector Interaction Module, the business inspector first logs in to the system, then he or she can: monitoring the system; handle unexpected situations; publish information.
- 6) In Administrator Interaction Module, the system administrator first logs in to the system, the he or she can: Manage information, such as adding, deleting, changing, and checking information; maintain the system in real time; update the system regularly.
- 7) In Intelligent Analysis Module, the system can: analysis information; process orders; analysis and plan driving routes; analysis unexpected situations and decide whether to inform the police.
- 8) In Database Module, the system can: process taxi-hailing user information; process driver and vehicle information; process order information.

3.2 Functional Requirements

3.2.1 Functional Structure

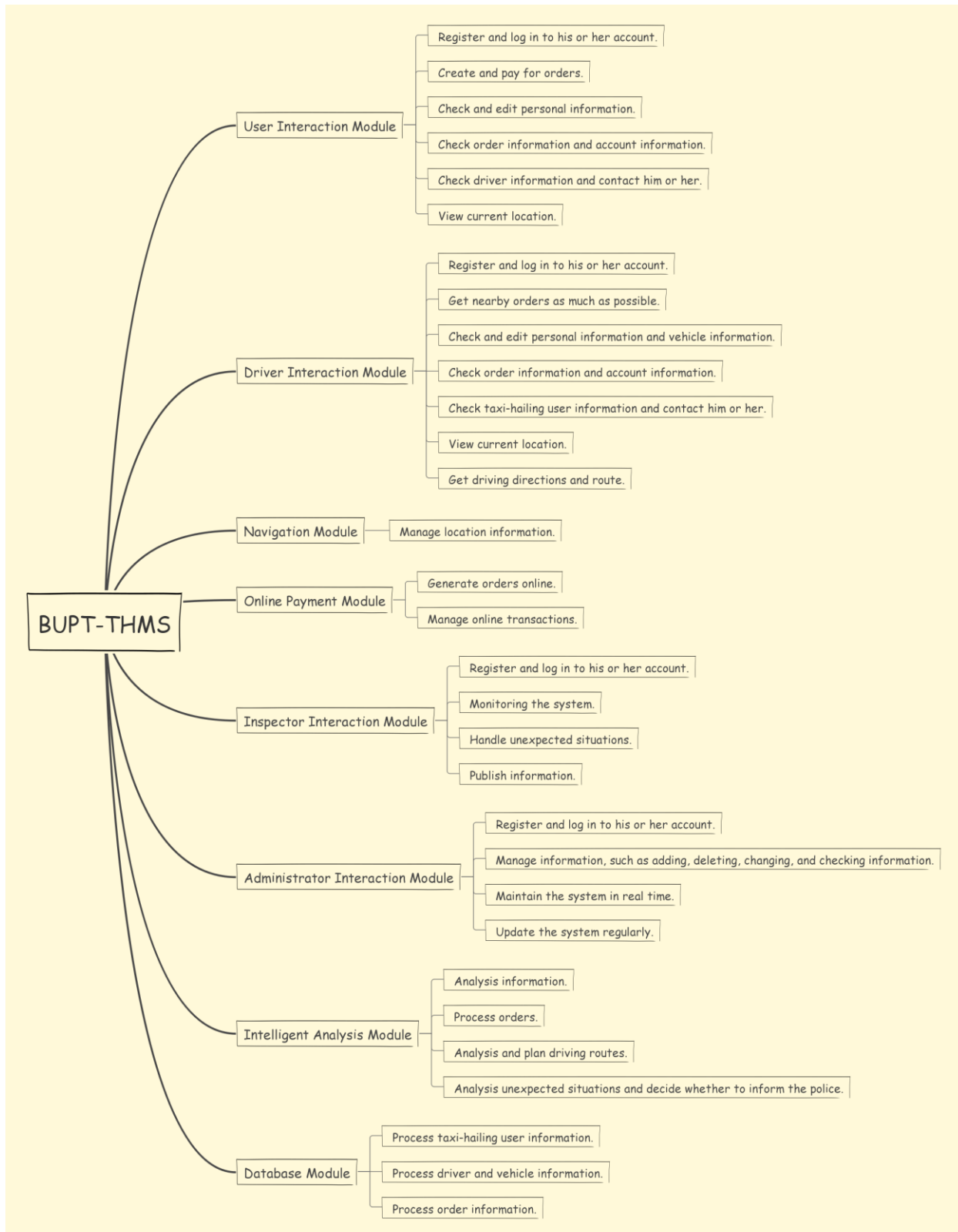


Figure 3-4 BUPT-THMS functional structure diagram

In the previous section, the data transfer process in the system was described. In this section, the functions of the system were summarized with the function structure diagram, and each function point was aggregated together.

Since each user role should have the same function, classifying the system according to the function module will result in intricate data flow. Therefore, the first layer of the system's functions is divided according to the user's role. Divided into eight modules: User Interaction Module, Driver Interaction Module, Navigation Module, Online Payment Module, Inspector Interaction Module, Administrator Interaction Module, Intelligent Analysis Module and Database Module.

3.2.2 User Interaction Module

3.2.2.1 Introduction

This module mainly implements the dynamic interaction between the taxi-hailing user and BUPT-THMS.

3.2.2.2 Inputs

Operations of taxi-hailing users.

3.2.2.3 Processing

After the taxi-hailing user logging in to his or her account, the system responds according to his or her operations.

- When the user chooses to create an order, the system will generate corresponding orders based on the user's current location and selected services.
- When the user chooses to check personal, order, account or driver information, the system will call the corresponding information in the database. And, the user can contact the driver based on the his or her information.
- When the user chooses to pay for his or her current order, the system will conduct online transactions based on order information. At the same time, the user's account balance will change accordingly.
- When the user chooses to edit his or her personal information, the system will make changes to the corresponding data in the database according to the user's addition, deletion, and modification operations.
- When the user chooses to view his or her current location, Navigation Module will be called to get his or her location information.

3.2.2.4 Outputs

- A new order.
- Personal, order, account and driver information.
- New account data.
- Modified personal information data in the database.
- Current location information.

3.2.2.5 Error Handling

- When the user chooses to edit his or her personal information, if the user enters illegal information data, the system will alert the user.
- When the user's Alipay or WeChat account balance is insufficient, the system alerts the user.

3.2.3 Driver Interaction Module

3.2.3.1 Introduction

This module mainly implements the dynamic interaction between the driver and BUPT-THMS.

3.2.3.2 Inputs

Operations of drivers.

3.2.3.3 Processing

After the driver logging in to his or her account, the system responds according to his or her operations.

- When the driver chooses to get nearby orders, the system will choose the most recent order for him or her as much as possible.
- When the user chooses to check personal, vehicle, order, account or taxi-hailing user information, the system will call the corresponding information in the database. And, the driver can contact the taxi-hailing user based on his or her information.
- After the system conducts online transactions based on order information, the driver's account balance will change accordingly.
- When the driver chooses to edit his or her personal information, the system will make changes to the corresponding data in the database according to the driver's addition, deletion, and modification operations.
- When the driver chooses to view his or her current location, Navigation Module will be called to get his or her location information.
- When the driver chooses to get driving directions and route, the system will provide the best solution.

3.2.3.4 Outputs

- Personal, vehicle, order, account and taxi-hailing user information.
- New account data.
- Modified personal information data in the database.
- Current location information.
- Best solution of driving directions and route.

3.2.3.5 Error Handling

- When the driver chooses to edit his or her personal information, if the driver enters illegal information data, the system will alert the user.
- When the driver deviates from the best route, the system will alert the driver and re-planning the best route.

3.2.4 Navigation Module

3.2.4.1 Introduction

This module mainly implements the dynamic interaction between the navigation module and BUPT-THMS.

3.2.4.2 Inputs

User's or driver's request for current location.

3.2.4.3 Processing

- When the user chooses to view his or her current location, Navigation Module will be called to get his or her location information.
- When the driver chooses to view his or her current location, Navigation Module will be called to get his or her location information.
- When the driver chooses to get driving directions and route, the system will provide the best solution.

3.2.4.4 Outputs

- Personal, vehicle, order, account and taxi-hailing user information.
- New account data.
- Modified personal information data in the database.
- Current location information.
- Best solution of driving directions and route.

3.2.4.5 Error Handling

When the driver deviates from the best route, the system will alert the driver and re-planning the best route.

3.2.5 Online Payment Module

3.2.5.1 Introduction

This module mainly implements the dynamic interaction between the online payment module and BUPT-THMS.

3.2.5.2 Inputs

Taxi-hailing users' operations to create orders.

3.2.5.3 Processing

After the system conducts online transactions based on order information, the taxi-hailing user's and driver's account balance will change accordingly.

3.2.5.4 Outputs

New account data.

3.2.5.5 Error Handling

When the user's Alipay or WeChat account balance is insufficient, the system alerts the user.

3.2.6 Inspector Interaction Module

3.2.6.1 Introduction

This module mainly implements the dynamic interaction between the inspector interaction module and BUPT-THMS.

3.2.6.2 Inputs

Inputs of business situations and operations of business inspectors.

3.2.6.3 Processing

After the business inspector logging in to his or her account, the system responds according to his or her operations. The system will show the current business status. The business inspector can monitor business status, handle unexpected situations and publish information by the system.

3.2.6.4 Outputs

- Handling results of accidents.
- Published information.

3.2.6.5 Error Handling

When the business inspector's operation is illegal, the system alerts the user.

3.2.7 Administrator Interaction Module

3.2.7.1 Introduction

This module mainly implements the dynamic interaction between the administrator interaction module and BUPT-THMS.

3.2.7.2 Inputs

Inputs of system situations and operations of system administrators.

3.2.7.3 Processing

After the system administrator logging in to his or her account, the system responds according to his or her operations. The system will show the current system status. The system administrator can manage information by the system, such as adding, deleting, changing, and checking information. When the system administrator chooses to maintain the system, the system can give real-time feedback. At the same time, the system administrator can update the system regularly.

3.2.7.4 Outputs

- Processed information.
- System after maintenance.
- Upgraded system.

3.2.7.5 Error Handling

When the system administrator's operation is illegal, the system alerts the user.

3.2.8 Intelligent Analysis Module

3.2.8.1 Introduction

This module mainly implements the dynamic interaction between the intelligent analysis module and BUPT-THMS.

3.2.8.2 Inputs

- Various inputs and operations.
- Information from other modules.

3.2.8.3 Processing

When the intelligent analysis system gets the information from other modules, various inputs and operations, the system will make corresponding intelligent analysis. The system can: analysis information; process orders; analysis and plan driving routes; analysis unexpected situations and decide whether to inform the police.

3.2.8.4 Outputs

- Processed information.
- Processed orders.
- Planned driving routes.
- Results of accident analysis.

3.2.8.5 Error Handling

When the inputs are illegal, the system alerts the user.

3.2.9 Database Module

3.2.9.1 Introduction

This module mainly implements the dynamic interaction between the intelligent analysis module and BUPT-THMS.

3.2.9.2 Inputs

Various operations on Database.

3.2.9.3 Processing

The database will give corresponding feedback on the operations of adding, deleting, modifying and checking.

3.2.9.4 Outputs

Updated database.

3.2.9.5 Error Handling

When the operations are illegal, the system alerts the user.

3.3 Use Cases

3.3.1 User Role

There are four user roles in BUPT-THMS: Driver, Taxi-Hailing User, Business Inspector and System Administrator. The following article will draw their use case diagrams for four kinds of users.

3.3.2 Use Case #1 Driver

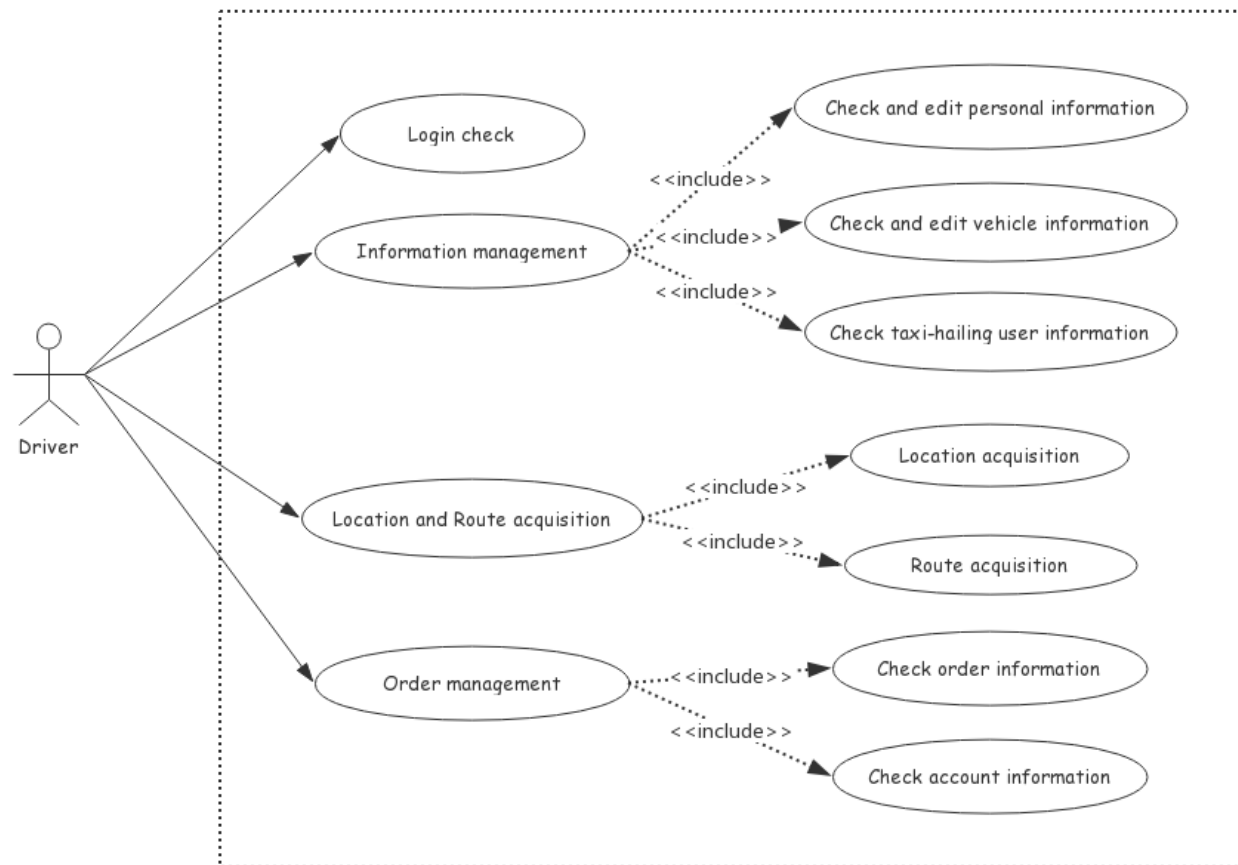


Figure 3-5 #1 Driver Use case diagram

Design discussion:

- 1) The function that Driver is responsible for is Driver Interaction Module.
- 2) Sub-functions of Driver Interaction Module include:
 - a) Login check.
 - b) Information management.
 - c) Location and Route acquisition.
 - d) Order management.

3.3.2 Use Case #2 Taxi-Hailing User

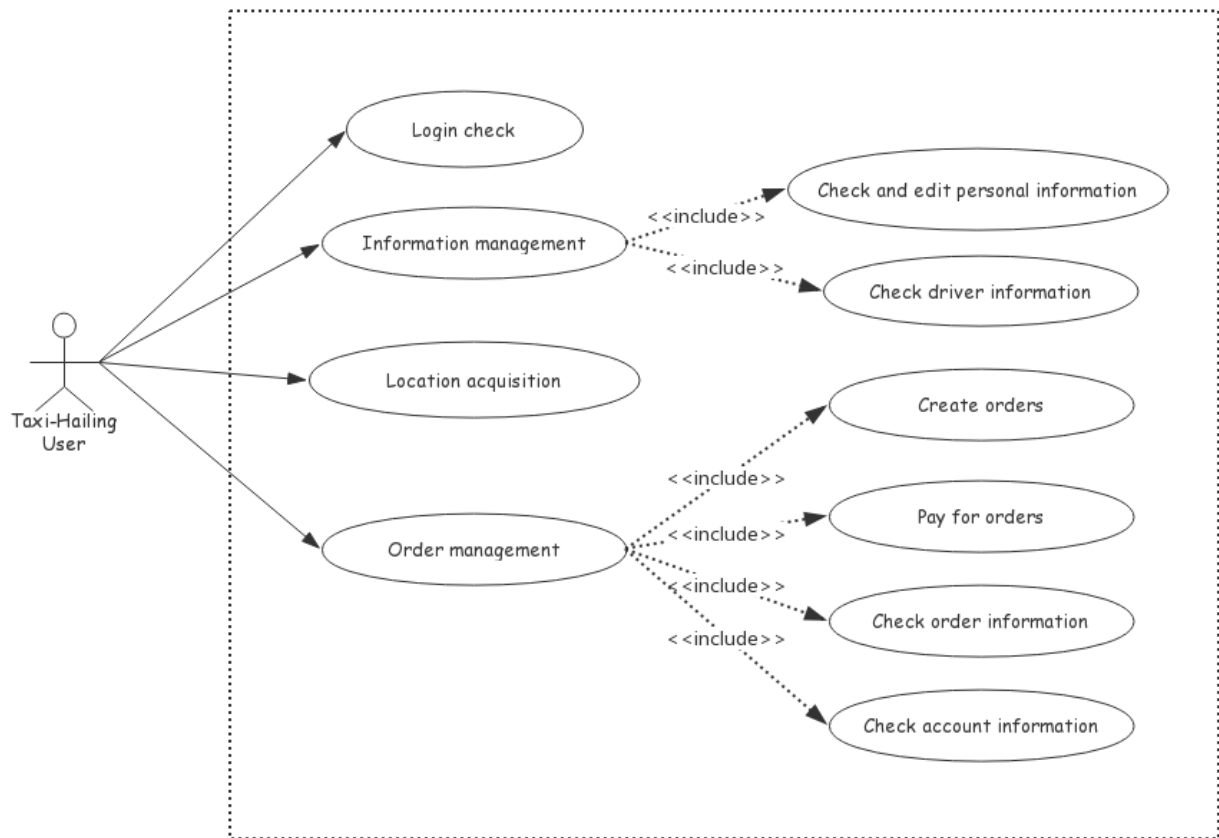


Figure 3-6 #2 Taxi-Hailing User Use case diagram

Design discussion:

- 1) The function that Taxi-hailing User is responsible for is User Interaction Module.
- 2) Sub-functions of User Interaction Module include:
 - a) Login check.
 - b) Information management.
 - c) Location acquisition.
 - d) Order management.

3.3.2 Use Case #3 Business Inspector

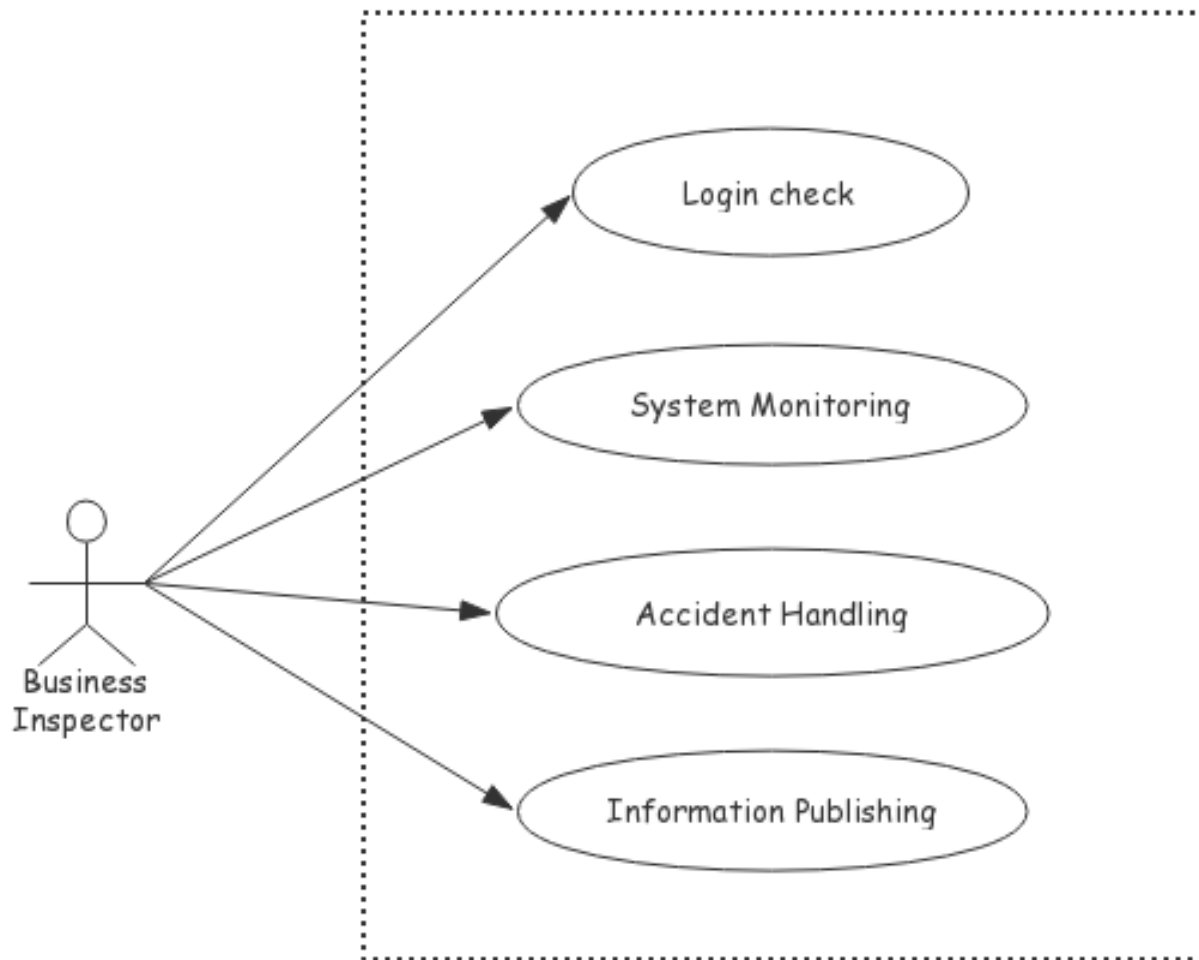


Figure 3-7 #3 Business Inspector Use case diagram

Design discussion:

- 1) The function that Business Inspector is responsible for is Inspector Interaction Module.
- 2) Sub-functions of Inspector Interaction include:
 - a) Login check.
 - b) System Monitoring.
 - c) Accident Handling.
 - d) Information Publishing.

3.3.2 Use Case #4 System Administrator

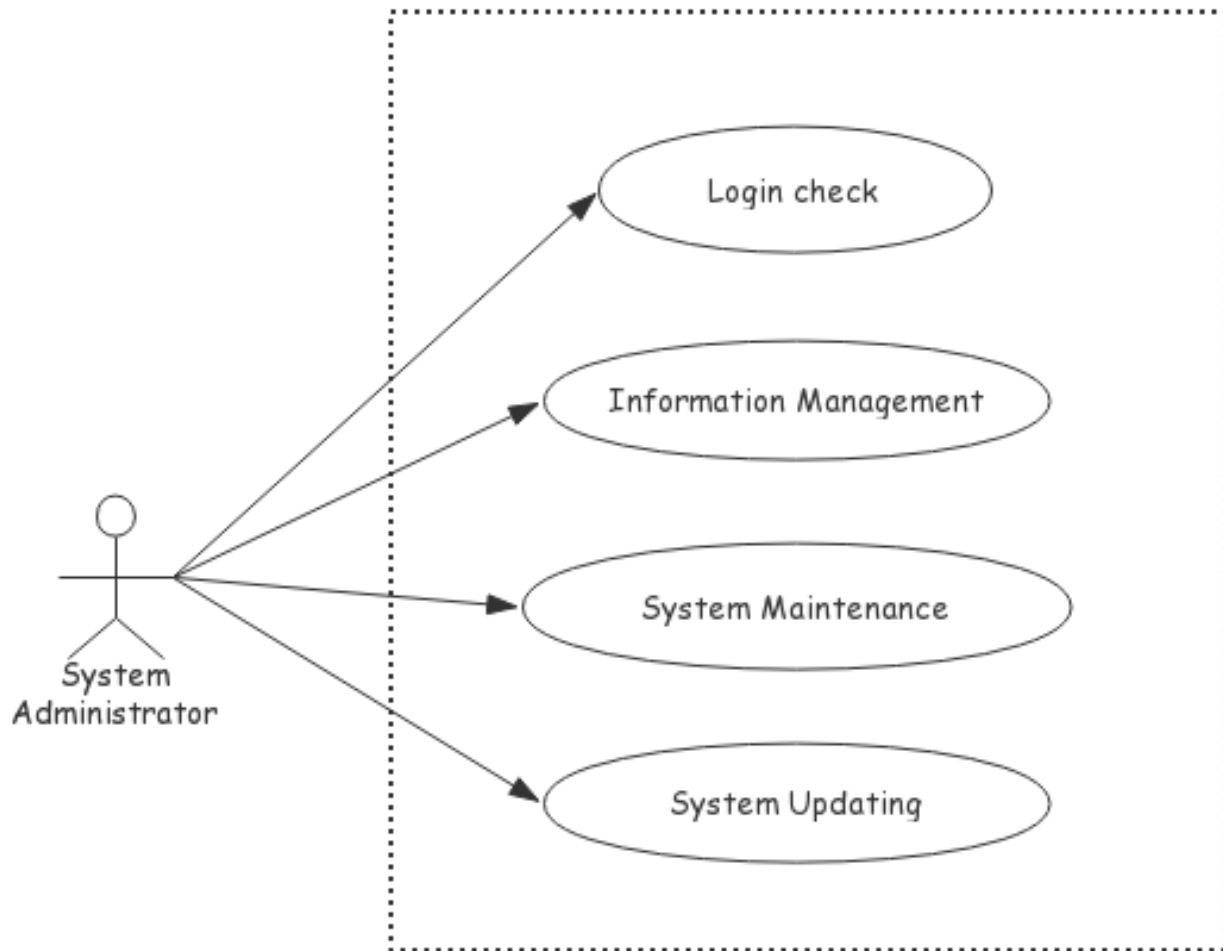


Figure 3-8 #4 System Administrator Use case diagram

Design discussion:

- 1) The function that System Administrator is responsible for is Administrator Interaction Module.
- 2) Sub-functions of Administrator Interaction Module include:
 - a) Login check.
 - b) Information management.
 - c) System Maintenance.
 - d) System Updating.

3.4 Classes / Objects

3.4.1 Driver Class

3.4.1.1 Attributes

Form 3-1 Driver Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
DriverId	Varchar	20
DriverPassword	Varchar	20
DriverName	Varchar	10
DriverGender	Char	2
DriverPhoneNumber	Char	11
LicensePlateNumber	Char	7
DriverAccount	Double	10

3.4.1.2 Functions

- 1) Login_check()
- 2) PIM_of_the_Driver()
- 3) Account_Management()
- 4) VehicleIM()
- 5) Order_Management()
- 6) View_PI_of_the_User()
- 7) Get_location()
- 8) Get_Route()

3.4.2 Taxi-Hailing User Class

3.4.2.1 Attributes

Form 3-2 Taxi-Hailing User Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
Taxi-HailingUserId	Varchar	20
Taxi-HailingUserPassword	Varchar	20
Taxi-HailingUserName	Varchar	10
Taxi-HailingUserGender	Char	2
Taxi-HailingUserPhoneNumber	Char	11
Taxi-HailingUserAccount	Double	10

3.4.2.2 Functions

- 1) Login_check()
- 2) PIM_of_the_User()
- 3) Account_Management()
- 4) Order_Management()
- 5) View_PI_of_the_Driver()
- 6) Get_location()

3.4.3 Business Inspector Class

3.4.3.1 Attributes

Form 3-3 Business Inspector Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
BusinessInspectorId	Varchar	20
BusinessInspectorPassword	Varchar	20
BusinessInspectorName	Varchar	10
BusinessInspectorGender	Char	2
BusinessInspectorPhoneNumber	Char	11

3.4.3.2 Functions

- 1) System_Monitoring()
- 2) Accident_Handling()
- 3) Information_Publishing()

3.4.4 System Administrator Class

3.4.4.1 Attributes

Form 3-4 System Administrator Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
SystemAdministratorId	Varchar	20
SystemAdministratorPassword	Varchar	20
SystemAdministratorName	Varchar	10
SystemAdministratorGender	Char	2
SystemAdministratorPhoneNumber	Char	11

3.4.4.2 Functions

- 1) Information_Management()

- 2) System_Maintenance()
- 3) System_Updating()

3.4.5 Navigation System Class

3.4.5.1 Attributes

Form 3-5 Navigation System Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
UserPosition	Varchar	20
DriverPosition	Varchar	20

3.4.5.2 Functions

Location_Information_Processing()

3.4.6 Online Payment System Class

3.4.6.1 Attributes

Form 3-6 Online Payment System Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
UserCost	Double	10
DriverIncome	Double	10

3.4.6.2 Functions

- 1) Bill_Generation()
- 2) Online_Payment()

3.4.7 System Database Class

3.4.7.1 Attributes

Form 3-7 System Database Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
UserInformation	InformationType(Varvhar)	50
DriverInformation	InformationType(Varvhar)	50
OrderInformation	InformationType(Varvhar)	50

3.4.7.2 Functions

- 1) User_Information_Processing()
- 2) Driver_Information_Processing()

3) Order_Information_Processing()

3.4.8 Intelligent Analysis System Class

3.4.8.1 Attributes

Form 3-8 Intelligent Analysis System Class Attributes

<i>Attribute Name</i>	<i>Type</i>	<i>Size</i>
UserInformation	InformationType(Varvhar)	50
DriverInformation	InformationType(Varvhar)	50
OrderInformation	InformationType(Varvhar)	50

3.4.8.2 Functions

- 1) Information_Analysis()
- 2) Order_Processing()
- 3) Route_Analysis_and_Planning ()
- 4) Accident_Analysis()

4. Non-Functional Requirements

The so-called non-functional requirements refer to the characteristics that software products must have in addition to functional requirements in order to meet the user's business needs. Non-functional requirements are often overlooked or not taken seriously during the requirements analysis phase. Non-functional requirements for software products include system quality requirements and engineering requirements. For this system, the non-functional requirements of the system are described mainly through the following three aspects.

4.1 Quality Requirements**4.1.1 Performance**

- 1) Under normal circumstances, the BUPT-THMS can guarantee 500,000 users online at the same time. At peak times, at least 1,000,000 users will be online at the same time.
- 2) When 500,000 users query at the same time, the response time does not exceed 1s. When 1,000,000 users query at the same time, the delay is no more than 3s. When 2,000,000 users simultaneously query, corresponding within 7s.
- 3) System response time Considering the number of students in the school, the response time of the system should be as reasonable as the following criteria:
 - The system should be able to view the basic information of the user. The response speed is optimally b0.2s, with an average of a0.5s and no more than m1.0s.
 - The system should be able to view the user's location information. The response speed is optimally b0.5s, with an average of a0.8s and no more than m1.0s.

- The system should be able to allow users to modify the information. The response speed is optimally b0.5s, with an average of a1.0s and no more than m1.2s.
 - The system should be able to make online route planning. The response speed is optimally b0.5s, with an average of a0.8s and no more than m1.0s.
 - The system should be able to conduct online transactions. The response speed is optimally b0.5s, with an average of a0.8s and no more than m1.2s.
- 4) There is a 99.9 percent probability that the system supports 1,000,000 users to use taxi-hailing function at the same time.
 - 5) The probability of system data error during transmission is less than 0.1%. The system has a 100% chance of finding and fixing errors.
 - 6) Backup system data once a week.
 - 7) The system runtime memory shall not be less than 512GB.
 - 8) Using a gigabit switch.

4.1.2 Reliability

- 1) Select the Linux operating system that does not need to be restarted for update.
- 2) Backup important data of the system in multiple places.
- 3) After the normal error occurs, the system can immediately detect and recover by itself.
- 4) After serious errors occur, the system can stop losses in time. Then the maintenance team can solve the errors quickly.
- 5) The system gives warning when the administrator submits important changes.
- 6) The legitimacy of user input should be verified, and users' critical modifications should be warned.
- 7) The system should undergo multiple tests before deployment.

4.1.3 Availability

- 1) The system's usability is reflected in its ability to support multiple operating systems and multiple browsers. At the same time, the system should have easy-to-operate features.
 - Provide user manuals for different users to facilitate user learning.
 - The system should provide an online help interface to facilitate user learning.
- 2) Because there are many people who operate the system, and the operating habits, education level, age, and ability to accept things are different, this requires the system to have good human-computer interaction capabilities. The various functions provided by the system are easy for the user to understand, the operation is simple, and the user is easy to grasp.
 - The interface design of the system should be concise and clear, so that users can learn to use the system themselves.
 - The system should have a certain aesthetic appearance, and can refer to the flat design of most current websites.

4.1.4 Security

- 1) Generally speaking, the actual management system must have corresponding security performance. Users at all levels of the system have their own permission settings. For example, taxi-hailing users cannot view or modify other people's information.
- 2) The system should ensure the security of data transmission by setting up a firewall. Use a reliable operating system to ensure the safe operation of the system and ensure that the system operates in a safe and reliable environment.
- 3) The system should ensure that user information is not leaked, and system configuration files and database storage files should be encrypted.
- 4) The system should ensure that it will not crash due to malicious attacks, and there are no obvious vulnerabilities in the system development process.
- 5) The system should be able to ensure that the selected developer does not have a commercial competitor or a similar malicious opponent.

4.1.5 Maintainability

The maintainability of software refers to the difficulty of improving software. The structure, interfaces, functions, and internal processes of the system are easily understood by maintenance personnel during the development and tracking phases. At the same time, the system has good testing and diagnostic system error features. When the system is applied under different public security management systems, it should have good adaptability. Users do not need to be modified through a large number of interfaces and internal processes.

4.1.6 Portability

- 1) Both android and apple clients should be provided.
- 2) The system can be adapted to all popular Linux system versions.
- 3) Support online update.
- 4) No conflict with other software in the system.

4.2 Engineering Requirements

In order to meet the customer needs and business needs of the development, after the system is installed, in the subsequent function maintenance and expansion, it needs to be well maintained and expandable, so that it can be upgraded and modified in the future.

4.2.1 Inverse Requirements

- 1) Details of each release iteration should be documented.
- 2) Retain historical source code.
- 3) Fully test and verify possible errors.
- 4) Define the boundary of system behavior, that is, define what should not be done by the system.

4.2.2 Design Constraints

- 1) The frameworks available for web development are: typical springMVC framework, J2EE, PHP framework.
- 2) The web server is deployed on a Linux system to ensure a stable and reliable service.
- 3) The hardware configuration of the Web server system needs to satisfy the server to be efficient and stable and retrograde. Need to meet:
 - CPU: the fifth generation or more dual core.
 - Memory: 8G or more.
 - Hard disk: 1T or more.
 - Can be equipped with a tape drive, etc.
- 4) Based on system security and confidentiality considerations, the system's configuration files, data storage files, etc. should be encrypted, using internationally accepted encryption algorithms to prevent accidental disclosure or malicious attacks.
- 5) Based on system reliability considerations, the system's data storage files should be redundantly backed up, such as disk redundant array storage RAID.
- 6) For future scalability of the system, the system hardware uses internationally-used hardware and should not use targeted hardware. The entire system should also minimize calls between modules and try to loosely couple them.

4.2.3 Logical Database Requirements

- 1) Basic Requirements:
 - a) The system recommends using MySQL. MySQL is a commercial database, open source, low cost, large usage and high reliability.
 - b) If the funds permit, you can use a large database such as Oracle, which has high reliability and high stability.
 - c) The field definition of the database should have some flexibility to ensure certain modifiability.
 - d) The design of the database table should ensure consistency and integrity, avoid data redundancy, and data inconsistency should be adjusted in time.
 - e) Under the above conditions, the database data storage hardtop saves space as much as possible.

- 2) Database Requirements:

Database requirements design is divided into two parts: conceptual structure design and logical structure design. Conceptual structure design refers to drawing an E-R model. The conceptual structure is further transformed into a data model supported by a DBMS, and then the data model is appropriately adjusted and optimized according to the logic design criteria, the semantic constraints of the data, the normalization theory, etc., to form a reasonable global logical structure, and designed User sub mode. This is the task that the database logic design has to accomplish.

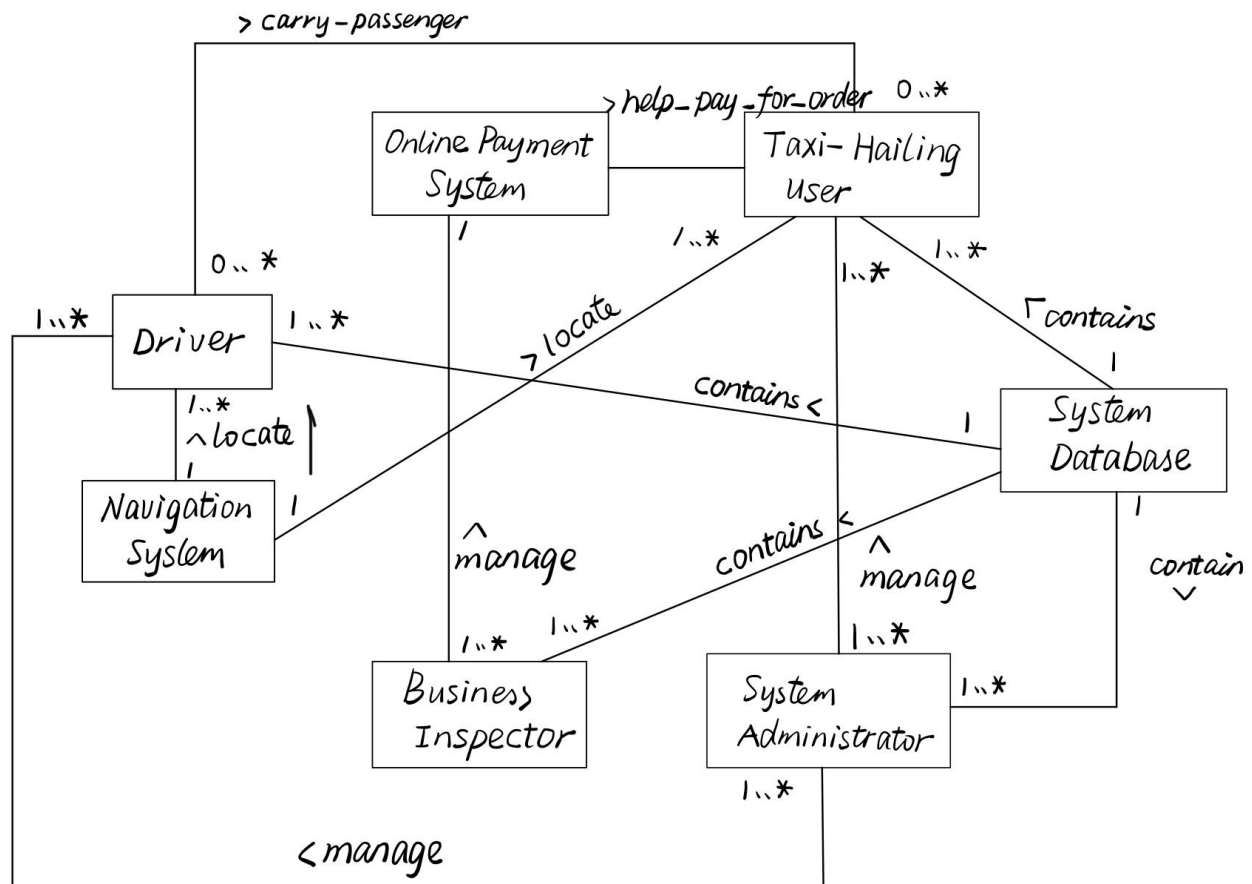


Figure 4-1 E-R diagram

4.3 Quality Requirements

The software service shall comply with the latest laws and regulations. Different regions and cities may have different regulations. Therefore, different functional services shall be provided automatically according to the city where the passengers are located. If the price increase is not allowed in some places, the function will not be used automatically in this city.

5. Change Management Process

When some benchmarks of the project change, the quality, cost and scope of the project will change accordingly. In order to ensure the realization of the project objectives, it is necessary to take necessary contingency measures for various changes of the project, which is demand change management. The process of managing requirements change can be represented by the following flow chart:

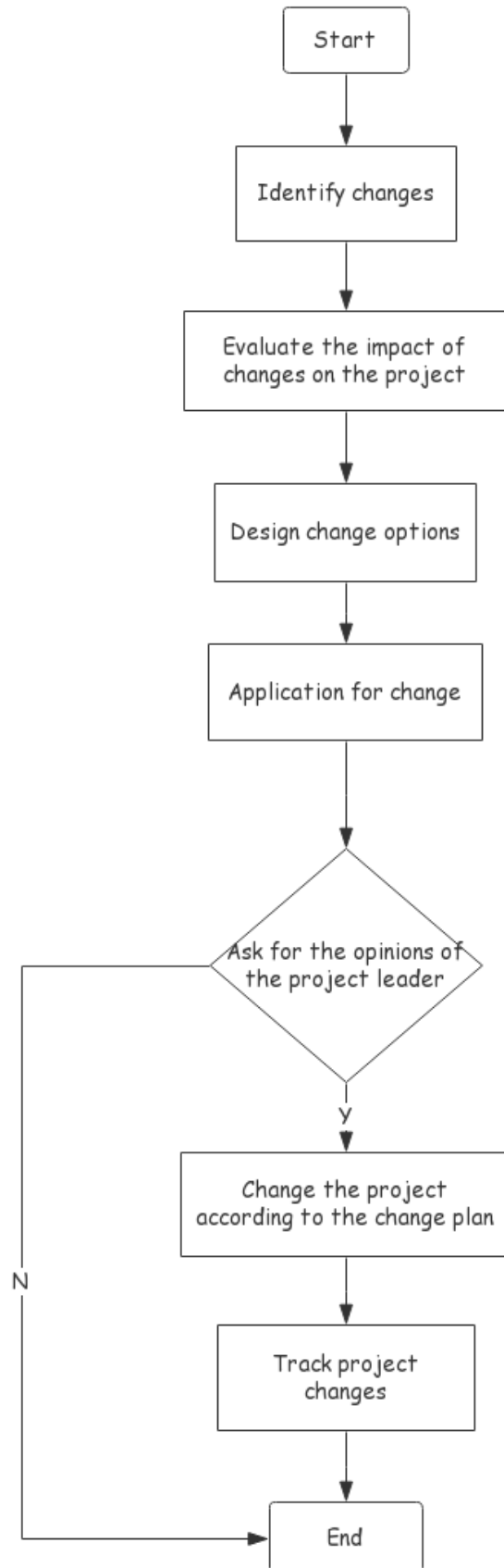


Figure 5-1 Change Management Process Diagram

The main design ideas are as follows:

- 1) Identify changes: Analyze whether the problem in the project belongs to the change requirement. The criterion for distinguishing whether it is the change requirement is whether a certain work is not in the project work benchmark.
- 2) Evaluate the impact of changes on the project: If it is a change request, conduct a feasibility analysis, what impact will the change have on the project cost, schedule, quality, etc., and determine whether the change is feasible.
- 3) Design change options: Develop a requirements change control document. List several possible change handling options, such as very urgent change requirements are approved immediately, and changes that have less impact on the project can be processed later.
- 4) Application for change: Formally submit a written request for change.
- 5) Ask for the opinions of the project leader: Decide whether or not to change and how to change based on the joint opinions of the client and the open party.
- 6) Change the project according to the change plan: All project leaders involved in the change should be involved in the project change.
- 7) Track project changes: After the change is approved, the system and related personnel need to track the implementation of the change and record it.

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

Form 1-1 Term and Definition

Figure 3-1 BUPT-THMS system context diagram

Figure 3-2 BUPT-THMS preliminary system conception

Figure 3-3 BUPT-THMS more detailed system conception

Figure 3-4 BUPT-THMS functional structure diagram

Figure 3-5 #1 Driver Use case diagram

Figure 3-6 #2 Taxi-Hailing User Use case diagram

Figure 3-7 #3 Business Inspector Use case diagram

Figure 3-8 #4 System Administrator Use case diagram

Form 3-1 Driver Class Attributes

Form 3-2 Taxi-Hailing User Class Attributes

Form 3-3 Business Inspector Class Attributes

Form 3-4 System Administrator Class Attributes

Form 3-5 Navigation System Class Attributes

Form 3-6 Online Payment System Class Attributes

Form 3-7 System Database Class Attributes

Form 3-8 Intelligent Analysis System Class Attributes

Figure 4-1 E-R diagram

Figure 5-1 Change Management Process Diagram