**CHAPTER 1**

**INTRODUCTION**

**INTRODUCTION**

**Aye Auto** is an android application that provides autoriksha’s near by the user. Driver also gets their customers through this application.

It is helpful to all the users to use this application. This application also provides the service to calculate hire and providing GPS facility to both the driver and customer. Customer can use this GPS for tracking the driver and the distance that used by driver. Driver can use the GPS for reach to the customer location and destination location.

This application reduces the man power. We can book an auto with some simple tasks. No need of a phone call or search in the roads.

### 1.1 SCOPE

The scope of this system is to provide user friendly interface resulting in knowing each and every usability features of the system. The Aye Auto can’t define just in the boundaries of auto service .Because the dynamics behind this technology opens up a wide array of opportunities, from auto riding to the entire logistic.

# **SYSTEM ANALYSIS**

System analysis is the process of collecting and interpreting facts, understanding problems and using the information to suggest improvements on the system. This will help to understand the existing system and determine how computers make its operation more effective. The aim of this analysis is to collect the detailed information on the system and the feasibility study of the proposed system. This analysis focuses on the flow of the system module by module and the efficiency of each. To design the proposed system we need the exact processing logic as well as the extended features of the existing system such as reliability, consistency, storage capacity etc. This report will discuss the advantages and drawbacks/disadvantages of the existing system and the modifications and enhancements can be done. This analysis will concentrate on the information gathering for the efficient, user friendly and reliable system, which will carry forward the features of the existing system.

The analysis is the process of identifying the particular problem or requirements of the software in details. Now the customer spends more time for direct going and searching or selecting models. This system will reduce the time and man power behind the system. The main aim of the system is to ordering and all transformation makes easily.

### 2.1 REQUIREMENT ANALYSIS

Requirements analysis results in the specification of software’s operational characteristics, indicates software’s interface with other system elements, and establishes constraints that software must meet. Requirements analysis allows you to elaborate on basic requirements established during the inception, elicitation, and negotiation tasks that are part of Requirements engineering.

**REQUIREMENT GATHERING**

The requirement gathering can be done by following ways.

* Interview
* Questionnaire
* Site visit
* Website visit

**2.1.1Interview questions**

**To the Driver.**

We asked set of questions to the driver. The questions are:

* How will you get the customer?
* What will you if the destination not familiar?
* What are the problems you faced in this field?
* How long have you been driving?
* How important is your passenger’s safety to you?
* Are comfortable working alone for long hours?
* How well do you know the city?
* If a customer was screaming at you because you made a wrong turn, how would you handle the situation?
* What do you think about uber and other ride sharing programs?
* What would you do if the passenger told you to turn off the meter and they would pay you cash for the ride?
* What area u normally drive?

**To the Customer.**

* How will you get the auto?
* Have you felt any kind of insecurity while traveling?
* How do you pay for the service?
* How do you get the auto in night, do you felt any difficulty?
* Will they take more amount of money?
* Do you heard about uber like app?
* Uber is a location based app that makes hiring an on-demand private driver easy, do you want this type of app?

**2.1.2 Questionnaire**

**To The Driver**

1. Do you currently drive Auto rickshaw?

Yes No

1. Did you ever faced any other expenses as an auto driver?

Yes No

1. Are you able to reach all destinations if the customer called you?

Yes No

1. Do you faced any problem to find location of the customer?

Yes No

1. Any problem you faced in this field?

Yes No

1. Are you comfortable to work for a long time drive?

Yes No

1. Did you reached on customer requested time?

Yes No

1. Are you available at the night?

Yes No

1. Do you charge the customer according to the meter reading?

Yes No

1. Did you consider the security of the customer?

Yes No

1. Do you have internet connection in your mobile phone?

Yes No

1. Do you know about Uber app?

Yes No

1. Uber is a location based app that makes hiring an on-demand private driver easy, do you want this type of app?

Yes No

1. Do you think it’s useful?

Yes No

1. Do you think that this type of app is useful for you to reach the specified location?

Yes No

**To The Customer**

1. Have you ever travelled in an Auto rickshaw?

Yes No

1. Do you face any kind of problem while travelling an auto rickshaw?

Yes No

1. Is there is any application used for travelling?

Yes No

1. Do you prefer to use auto rickshaws?

Yes No

1. Did you wait for the auto rickshaws?

Yes No

1. Did you get auto rickshaws at night?

Yes No

1. Do you felt any kind of insecurity while travelling at night?

Yes No

1. Did the driver charge you according to the meter reading?

Yes No

1. Will they take more amount of money?

Yes No

1. Are you using mobile phone?

Yes No

1. Do you have internet connection in your mobile phone?

Yes No

1. Do you know about Uber app?

Yes No

1. Uber is a location based app that makes hiring an on-demand private driver easy, do you want this type of app?

Yes No

1. Do you think it’s useful?

Yes No

1. Do you think that this type of app is useful for any situation while travelling by auto?

Yes No

**EXISTING SYSTEM**

Recent transportation and travelling system are loaded with high need of human power and other problems. Need of the auto increasing in the daily life. Our destinations are spitted to lot of other cities and villages. And it is very time consuming process when the customer is standing alone with an unknown place. Found a suitable auto is very risky and complicated. Many of them force to adjust their needs with the available auto or make the trip cancel. Founding a suitable cab is one of the important factor for comfort travel. And also there is no proof left which we are travelled in that auto. And the customer doesn’t know the name and address of the driver. Travelling with a totally strange man without any proofs are high risky. Normally when a customer needs a cab then he or she call their known drivers, and if they are busy then there is no other options. And he must need to search on road. This becomes more difficult when he stand in unknown places. Then there is no chance for known drivers. And also don’t know the root and map, Also drivers are waiting and seeking customers for long time. They are ready for trip, but no one call him. That will effect badly on his income. He only gets the trip from his known peoples and the customer who seeks the cab in road. And also customer is not fully satisfied with his cab. Because which one is not suitable for them and they are adjusting it, with the absence of proper autos. Driver don’t know the customer even the customer name, address. Travel with a stringer is risky and he not left any evidence he uses his auto.

Existing system is not user friendly because there is much support with the current technology and also more time consuming one. Every one want to wait lot of time for getting an auto. When the customer is stand in strange place then he can’t share the actual place he stands to the driver. And it make risky to found him. Also current system have a chance of problem in the calculation of hire.

**PROPOSED SYSTEM**

This proposed application is the ability to overcome all the current facing problems. This application provides drivers to customers, also customers for auto drivers. This is a user friendly application and it is capable to work with android mobile phones and tabs. Android is a top grossing mobile operating system. So it is helpful to all the users to use this application. This application also provides the service to calculate hire and providing GPS facility to both the driver and customer. Customer can use this GPS for tracking the driver and the distance that used by driver. Driver can use the GPS for reach to the customer location and destination location.

This application providing security to both drivers and customers. Customer must want to pay some fixed amount of money for the booking of a cab. This money is transformed to the admin account. And after the completion of trip this money is moved to drivers account. At that time customer want to pay the balance amount to the driver. This method helps to reduce false booking by the customer. And also there is registration fee for taxi registration. A small amount is needed to give admin. It help to increase the security and also from fake autos. Auto driver need to move as soon as possible after accepting the request. Otherwise his notification become disabled and notification is again transmitted to other drivers.

This application reduces the man power. We can book an auto with some simple tasks. No need of a phone call or search in the roads. Only need a proper working network connection with this application. Customer can get the driver’s name, vehicle registration number, phone number and such information about the driver. Also driver can get details about the customer like name, phone number, land mark this is the one of benefit given by this proposing application. This would improve the travelling experience. And also it reduces the risk that travel with a stranger. At a time customer can sent notification to more than one driver and the first driver who accept the request get this trip and others notification become disabled. So the customer can reduces the time to get the cab. And there is included the facility to choose suitable cabs according to their wishes. Improve the joy of the journey. This proposed system is helpful to both the customer and driver.

### 2.4 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that is spent on it. Feasibility study lets the developer foresee the future of the project and the usefulness. Feasibility study is a test of the system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use of resources. Thus when a new project has proposed, it normally goes through a feasibility study before the development. The system proposed has tested to check whether it is feasible by conducting the following:

* Technical feasibility
* Economic feasibility
* Behavioral feasibility

 Legal feasibility

**2.4.1 Technical Feasibility**

The project entitles “Aye Auto” is technically feasible because of the below mentioned feature. The project has developed in Android and in web application, which provides reliability, security and portability. The main advantage of android is it is used for mobile application development. The proposed system can run on any mobile phones and tablets, supporting android operating system and internet services. So it would be feasible in all technical terms of feasibility.

**2.4.2 Economic Feasibility**

For any system if the expected benefits equal or exceed the expected costs, the system can be judged economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system. We have initial cost for hosting the internet to the software.

**2.4.3 Behavioral Feasibility**

The system is designed in user friendly manner and we need not to provide any special training for the persons using this software. In the case of android application, this is a simple application to use as like other application.so no need to provide any special training for using this application software and hence it is behavioral feasible.

**2.4.4 Legal feasibility**

Legal feasibility study determines whether the proposed system conflicts with legal requirements. It determines whether the proposed system violates any copyright act or any rules in the organization. Since the proposed system does not violate any copyright act and it does not break any rules in the organization the proposed system is legally feasible.

### 2.5 SYSTEM REQUIREMENT SPECIFICATION

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent. The software specification document satisfies the following:-

* It specifies the external system behaviors.
* It specifies constraints on the implementation.
* It is easy to change.
* It serves as reference tool for system maintainers.
* It record forethought about the life cycle of the system.
* It characterizes acceptable response to undesired events**.**

**2.5.1 ACTOR IDENTIFICATION**

An actor is someone or something that interacts with the system. An actor is he /she who uses the system. An actor exchanges information with the system. Asking certain questions as detailed below can identify the actors of the system.

|  |  |  |
| --- | --- | --- |
| **1.** | Who will use the main functionality of the system? | Administrator,  Driver, Customer |
| **2.** | Who will lead support from the system and do their daily tasks? | Administrator,  Driver, Customer |
| **3.** | Who will maintain and administrate the system? | Administrator |
| **4.** | Which hardware devices does the system need to handle? | PC, Mobile phone(smart phone) |
| **5.** | With which other systems, does this system need to interact? | Database |
| **6.** | Who has interest in the result produced by the system? | Administrator,  Driver, Customer |

As per the above answers we can conclude the actors. They are:

* + - * Admin
      * Driver
      * Customer

**2.5.2 USECASE IDENTIFICATION**

A use cases represents the functionality of an actor. It is defined as a set of actions performed by a system, which yields an observable result. An ellipse containing its name inside the ellipse or below it represents it. It is placed inside the system boundary and connected to an actor with an association. This result shows how the use cases and the actor interact.

To find out the use cases, ask the following questions to each of the actors.

* Which functions does the actor require from the system? What does the actor need to do?
* Does the actor need to read, create, destroy, modify or store some kind of information in the system?
* Could the actor’s daily work be simplified or made more efficient by adding new functions to the system?

**2.5.2.1 USE CASES**

**Use case for the actor Administrator**

|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the Administrator require from the system? What does the Admin need to do? | Administrator requires the following functionalities from the system such as driver verification, view rider, send notification to driver and customer ,view booking , view complaint of driver and customer . |
| **2** | Could the Administrator’s daily work simplified by adding new functions to the system? | Yes, the system can reduce his/her daily work. |

Above questions, give the following use cases for the actor Administrator.

* Login
* Driver verification
* View rider
* Send notification to customer
* View booking
* Send notification to driver
* View customer complaint
* View Driver complaint
* logout

**Use case for the actor driver**

|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the driver require from the system?  What does the user need to do? | Driver requires the following functionalities from the system such as Registration, View profile, update profile, view request, Approve or reject request. View customer location, View route, view customer and admin notification, made complaint. |
| **2** | Could the supervisors’ daily work be simplified or made more efficient by adding new functions to the system? | Yes, the system can reduce his/her daily work. |

Above questions, give the following use cases for the actor Driver

* Login
* Registration
* View Profile
* Update profile
* View request
* Approve or reject request
* View customer location
* View customer notification
* Made complaint
* View admin notification
* logout

**Use case for the actor customer**

|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the customer require from the system?  What does the user need to do? | Driver requires the following functionalities from the system such as Registration, View driver and vehicle, view request status, track driver, Advance payment, calculate hire. View admin notification .made complaint. |
| **2** | Could the supervisors’ daily work be simplified or made more efficient by adding new functions to the system? | Yes, the system can reduce his/her daily work. |

Above questions, give the following use cases for the actor customer

* Login
* Registration
* View vehicle
* Send request
* View request status
* Track driver
* View route
* Advance payment
* View admin notification
* Made complaint
* logout

**2.5.2.2 USE CASE DIAGRAM**

A use cases represents the functionality of an actor. It is defined as a set of actions performed by a system, which yields an observable result. An ellipse containing its name inside the ellipse or below it represents it. It is placed inside the system boundary and connected to an actor with an association. This shows how the use cases and the actor interact. The use case diagram of my project is shown below. The actor of this project is administrator, driver and customer.

Admin

Customer

Driver

**2.5.3. ACTIVITY DIAGRAM**

The activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario. It uses rounded rectangles to imply a specific system function, arrows to represent flow through the system, decision diamonds to depict a branching decision, and solid horizontal lines to indicate that parallel activities are occurring.

The basic purposes of activity diagrams are similar to other diagrams. It captures the dynamic behavior of the system. Other diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

So the purposes can be described as:

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.

Describe the parallel, branched and concurrent flow of the system

Admin

View riders

Login

Authentication

Send notification

No

Yes

View booking

View complaint

Driver verification

Driver

No

View Route

View notification

Edit details

Login

Authenticate?

View

request

Yes

Approve request

Registration

View location

Made complaint

Customer

No

Track driver

Payment

View vehicle

Login

Authenticate?

Send request

Yes

View status

Registration

Add complaint

View notification

**2.5.4 SEQUENCE DIAGRAM**

Sequence diagrams are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence. It shows the objects participating in the interaction by their life lines and the messages they exchange, arranged in a time sequence.

A sequence diagram has two dimensions: a vertical dimension represents time,

Horizontal dimension represents different objects. The vertical line is called the object’s lifeline.

The lifeline represents the object’s existence during the interaction. This form was first popularized by Jacobson. An object is shown as a box at top of a dashed vertical line. A role is slot for an object within a collaboration that describes the type of object that may play the role and its relationships to other roles. However, a sequence diagram does not show the relationships among the roles or the association among the objects. An object role is shown as a vertical dashed line, the life line.

The sequence diagram is very simple and has immediate visual appeal- this is its greatest strength. A sequence diagram is an alternative way to understand the overall flow of the control of a program. Instead of looking at the code and trying to find out the overall sequence of behavior, we can use the sequence diagram to quickly understand that sequence.

Customer

Driver

Database

Admin

54

login

Registration

Registration

Driver verification

login

login

Send request

Edit Profile

View request

accept request

view status

Track driver

View rider

payment

Add rating

Add complaint

Add complaint

view complaint

logout

logout

logout

### 2.6 SYSTEM REQUIREMENTS

**2.6.1 HARDWARE AND SOFTWARE REQUIREMENTS**

Hardware and software requirements for the installation and smooth functioning of this product could be configured based on the requirements needed by the component of the operating environment that works as front-end system here we suggest minimum configuration for the both hardware and software components.

Working off with this software is requirements concrete on system environments. It includes two phases.

* Hardware Requirements
* Software Requirements

**CHAPTER-3**

**SYSTEM DESIGN**

**3.0 SYSTEM DESIGN**

Design is a meaningful engineering representation of something that is to be built. It is an iterative process through which requirements are translated in to a blueprint for constructing the software. The goal of the design phase is to plan a solution of the problem specified by the requirements document.

Major activities during the design phase are:

* Data Base Design
* Architectural Design
* Interface Design
* Modular Design

### 3.1. DATABASE DESIGN

A database is collections of inter related data stored with minimum redundancy to serve many users quickly and efficiently. In database design data independence, accuracy, privacy, and security are given higher priority. Database design is an integrated approach to file design. This activity deals with the design of the physical database. All entries and attributes have been identified while creating the database. The database design deals with the grouping of data into number of tables so as to reduce the duplication of data, minimize storage space, and retrieve the data efficiently.

Guidelines for designing a database:

* Design a relational schema so that it is easy to explain its meaning. Do not combine attributed from multiple entity and relationship types into a single relation.
* Design the database schema so that no insertion, deletion or modification anomalies are present in the relation.
* As far as possible, avoid placing attributes in a base relation whose values may frequently
* Design relation schemas so that they can be joined with equality conditions on attributes that are either primary keys or foreign keys in a way that no spurious tuples are generated.

**Advantage**

* Ease of use
* Data independence
* Accuracy and integrity
* Avoiding inordinate delays
* Recovery from failure  Privacy and security.

**3.1.1. E-R DIAGRAM**

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and relationship between entities, within an information system.

**There are three basic elements in ER models:**

* + - * **Entities** are the “things” about which we seek information  **Attributes** are the data we collect about entities.
      * **Relationships** provided the structure needed to draw information from multiple entities.

**E-R Diagram Symbols:**

|  |
| --- |
|  |

Entity

Attributes

Relation

**3.1.2 TABLE DESIGN**

In the database all the information are stored in the form of tables. A table is simply a way storing data in rows and columns. In the system data is stored in many tables. The table structures are shown below with sample data.

1. Log\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| lid | Int | Primary key | To provide a unique id |
| User name | Varchar(20) | Not null | Name of user |
| Password | Varchar(20) | Not null | Password |
| User type | Varchar(10) | Not null | Type of user |

1. Drive\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| drid | Int | Primary key | To provide a unique id |
| fname | Varchar(15) | Not null | Driver first name |
| lname | Varchar(15) | Not null | Driver last name |
| house\_name | Varchar(20) | Not null | House name |
| place | Varchar(20) | Not null | place |
| district | Varchar(20) | Not null | Driver district |
| pincode | Int | Not null | Pin code |
| Adhar\_no | Varchar(15) | Not null | Adhar card number |
| Vehicle\_no | Varchar(15) | Not null | Vehicle number |
| RC | Varchar(100) | Not null | Rc book photo |
| Vehile name | Varchar(20) | Not null | Vehicle name |
| Auto\_type | Varchar(20) | Not null | Auto type |
| Vehi\_photo | Varchar(100) | Not null | Photo of auto |
| lid | int | Foreign key | Login id |

1. Customer\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| Phone\_no | bigint | Primary key | To provide a unique id |
| fname | Varchar(20) | Not null | Name of user |
| lname | Varchar(20) | Not null | Last name of user |
| email | Varchar(20) | Not null | email of user |
| imi | Varchar(25) | Not null | Imi code of phone |

1. Route\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| rid | Int | Primary key | To provide a unique id |
| Route from | Varchar(20) | Not null | Route from |
| Route to | Varchar(20) | Not null | Rout to |

1. Drivalloc\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| alid | Int | Primary key | To provide a unique id |
| drid | int | Foreign key | Driver id |
| rid | int | Not null | Route id |

1. Comp\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| cid | Int | Primary key | To provide a unique id |
| complaint | Varchar(50) | Not null | complaints |
| Phone\_no | int | Foreign key | Phone number |
| drid | int | Not null | Driver id |
| replay | Varchar(50) | Not null | Replay |

1. Notification\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| nid | Int | Primary key | To provide a unique id |
| notification | Varchar(20) | Not null | Notification |

1. Locate\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| lid | Int | Primary key | To provide a unique id |
| longitude | Varchar(20) | Not null | longitude |
| latitude | Varchar(20) | Not null | latitude |
| alid | int | Foreign key | Allocation id |

1. Reqbook\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| reid | Int | Primary key | To provide a unique id |
| Sourse | Varchar(20) | Not null | sourse |
| destination | Varchar(20) | Not null | destination |
| phone | bigint | Not null | Phone\_no |
| status | Varchar(20) | Not null | Status |
| date | date | Not null | Date |
| time | time | Not null | Time |

1. Rate\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| rid | Int | Primary key | To provide a unique id |
| Phone\_no | bigint | Not null | Phone number |
| Rate | Varchar(20) | Not null | Rate of ride |
| Drid | Int | Foreign key | Driver id |

1. Message\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| Mid | Int | Primary key | To provide a unique id |
| Phone\_no | bigint | Foreign key | Phone number |
| Message | Varchar(50) | Not null | Message |
| Replay | Varchar(50) | Not null | Repaly |

1. Payment\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| Pid | Int | Primary key | To provide a unique id |
| Phone\_no | bigint | Foreign key | Phone number |
| date | date | Not null | Date |
| User type | Varchar(10) | Not null | Type of user |
| killometer | Varchar(10) | Not null | Kilometer |
| amount | Varchar(30) | Not null | amount |
| time | time | Not null | Time |

1. Account\_tb

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| aid | Int | Primary key | To provide a unique id |
| acc\_no | int | Not null | Account number |
| bank | Varchar(30) | Not null | Bank name |
| amount | Varchar(10) | Not null | Amount |

### 3.2 ARCHITECTURAL DESIGN

The architectural design develops a modular program structure and represents the control relationships between modules. It also defines interfaces that enable data to flow throughout the program.

**3.2.1. DATA FLOW DIAGRAM**

A data flow diagram is a graphical technique that depicts data flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD also called a fundamental system model or context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in next level i.e., Level 1 DFD. Each of the processes represented at level 1 are sub functions of overall system depicted in the context model.

**Data flow diagram symbol:**

|  |
| --- |
|  |

Source/Destination of Data

Data flow

Process

Storage

Level 0: Context Level

Request

Request

Data base

User

Response

Response

Level 1:

Login

Username &password

Admin

Login

Username &password

Database

Driver

data

Login

Customer

Level 2: Admin

Admin

Driver\_tb

Route\_tb

Drialloc\_tb

cust\_tb

notification\_tb

complaint\_tb

Level 2: Driver

Driver

driver\_tb

drivalloc\_tb

reqbook\_tb

locate\_tb

notification\_tb

message\_tb

complaint\_tb

payment\_tb

rate\_tb

Level 2: User

cust\_tb

Customer

reqbook\_tb

drivalloc\_tb

locate\_tb

message\_tb

complaint\_tb

rate\_tb