Initial Report

WinWin

Ride-hailing platform for local-motorcycle service provider and user

Present to

Assoc. Prof. Dr. Chotirat Ratanamahatana

By G14 Waterfall

6230123921 Thitaree Setwipattanachai

6230252121 Tarm Kalavantavanich

6231301421 Kanokpich Chaiyawan

6231304321 Kittipong Deevee

6231307221 Jirawat Kusalangkurwat

6231333521 Nopdanai Sayamnet

6231353021 Raviporn Akekunanon

6231372021 Atiwat Deepo

Term Project

2110322 Database Systems, Semester 1 of Academic year 2021

Department of Computer Engineering,

Faculty of Engineering, Chulalongkorn University

**Table of Contents**

[Introduction 1](#_Toc83936098)

[Organization background 1](#_Toc83936099)

[Objective of the project 1](#_Toc83936100)

[An overview of the system 2](#_Toc83936101)

[System Functionalities 3](#_Toc83936102)

[ER Diagram 6](#_Toc83936103)

[Document-based design schema 7](#_Toc83936104)

[Data Dictionary 8](#_Toc83936105)

Project name: WinWin

Ride-hailing platform for local-motorcycle service provider and user

# Introduction

Nowadays, motorcycle taxis are a major type of transportation in Bangkok (second only to MRT). However, riders have low income because ride-hailing platforms snatch their market share. Moreover, ride-hailing platforms can make users more satisfied than motorcycle taxis e.g., users can call ride-hailing platforms everywhere.

Therefore, WinWin wants to digitalize the motorcycle taxi system and to utilize route familiarity and locality of local motorcycle taxis to be an advantage that other ride-hailing platforms do not have.

## Organization background

WinWin, an online platform that connects customers with the local motorcycle taxis, has two business partners. Firstly, the Department of Land Transport, Ministry of Transport, which provides WinWin with the information about motorcycle taxis in the Bangkok area. Secondly, Winnonie, a startup founded by Bangchak Corporation group that rents out electric motorcycles.

WinWin believes that employing local motorcycle taxis as service providers is the best option since the riders are familiar with the route and can arrive at the customer's location faster than other riders from other ride-hailing platforms.

WinWin also thinks that building this application would satisfy stakeholders such as the Department of Land Transportation, Ministry of Transport, Winnonie, Motorcycle taxis, and consumers. Because Winnonie would be the market leader in the electric-vehicle rental market, the government and the Ministry of Transportation would receive a lot of positive credit for reorganizing local motorbike service. Riders would have more ways to make money, and consumers would benefit from a low-cost service provided by locals.

# Objective of the project

The aim of this project is to develop a Mobile-based Application that will give customers the advantage of the locality of motorcycle taxis and cheaper prices. Along with increasing income to motorcycle taxis that have their market share taken from various ride-hailing nowadays, it also makes motorcycle taxis more reliable and fairer price. This project will match users with motorcycle taxis, clearly state the fare and let users select their own preference e.g., fast/slow ride.

# An overview of the system

This application is designed to bring benefits to both users and motorcycle taxis. Both customers and riders will use the same application, however, their interface will be different based on their user type.

For rider to get started, they will need to register through filling in their full name, citizen ID and reference number from Department of Land Transport, pay entrance fee including taking a picture of themselves to verify the identity of motorcycle taxis. This is to assure users that all motorcycle taxis in the application will be legal motorcycle taxis and to build confidence to customers that the rider is the approved one.

On the user side to register, it is necessary to verify identity through personal information.

The match making system starts when the user selects the pick-up location and destination they want to go to, selects payment method, and selects rider preferences. The motorcycle taxis in the surrounding location will be notified that there is new user’s ride booking. When a motorcycle taxi accepts a ride from any user, users will see motorcycle plate number and rider name, and then wait for a motorcycle taxi to pick up.

During the service, on motorcycle taxi side, there shall be an update to let the system know that the motorcycle taxi has arrived, on the way to the destination, or arrived at the destination.

When the service is completed. The customer will be charged for the ride fare via the selected payment method and gives a review the rider who serves them with rating and description.

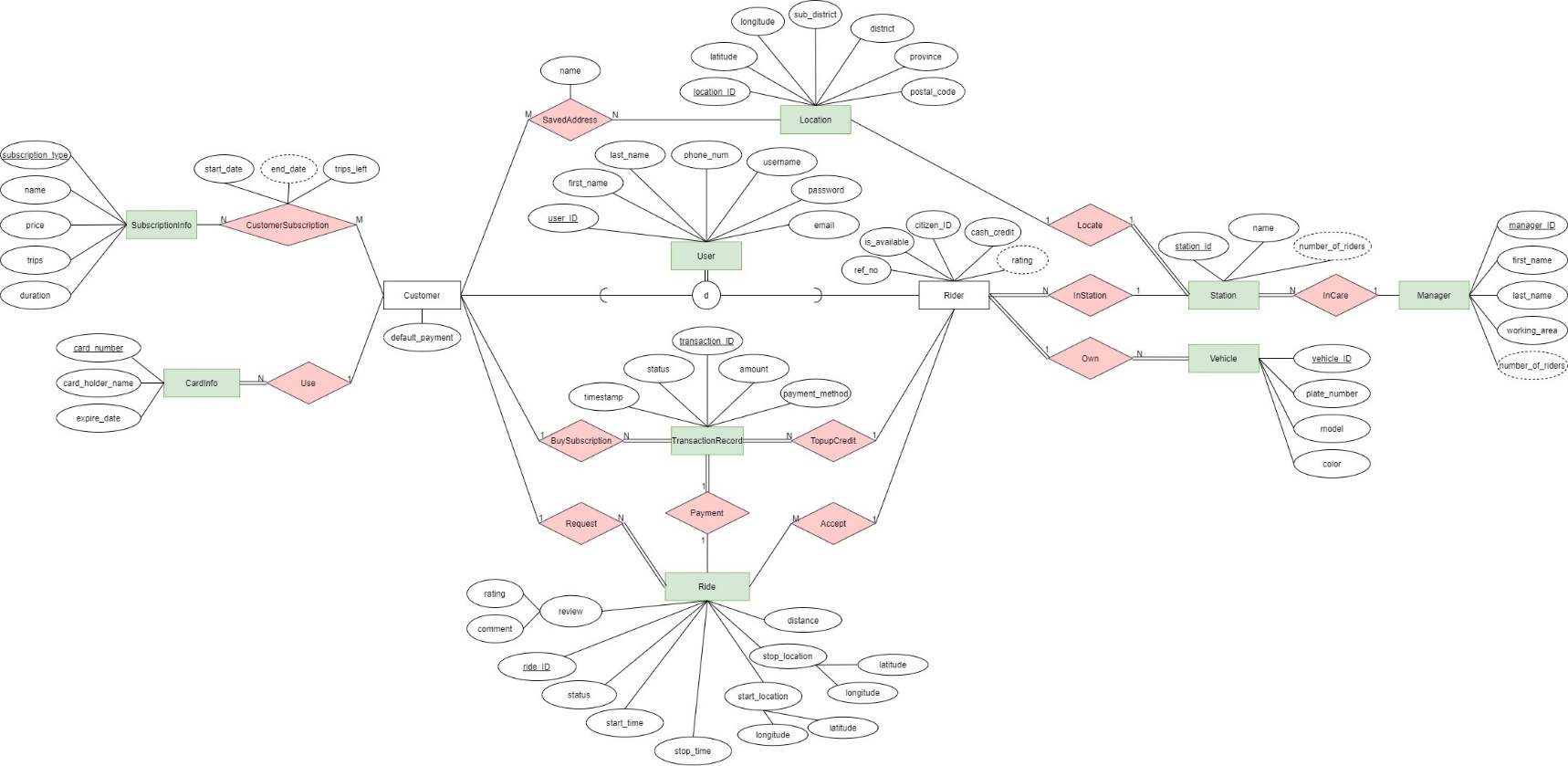
Diagram

Description automatically generated

# System Functionalities

1. Register and manage account
   1. The system shall allow both user and rider to create a new account with a unique userID on the system.
   2. The system shall allow both user and rider to create a specific password corresponding to the user’s userID.
   3. The system shall allow both user and rider to select account type from two categories including user and rider.
   4. The system shall allow the user to fill-in their profile including citizen ID number, first-name, last-name, and phone number.
   5. The system shall allow the rider to fill-in their profile including Reference number, first-name, last-name, citizen ID number, and phone number.
   6. For rider to start the service, the system shall allow the rider to pay entrance fee via bank transfer, credit card or debit card. This entrance fee will return to rider as rider’s cash credit
   7. The system shall allow both user and rider to update their profile.
   8. The system shall allow the user to save their favorite location.
2. Login/logout system
   1. The system shall allow the user and rider to login/logout the system.
   2. When a user or a rider tries to login, the system shall validate their identity by userID and password.
3. Match riders to user
   1. The system shall allow riders to set their availability to either available or unavailable.
   2. The system shall make notifications to riders about ride requests made by users in their acceptable vicinity.
   3. The system shall allow riders to accept ride requests that are available.
   4. The system shall allow riders to decline available ride requests that are notified to them.
   5. The system shall allow riders to cancel their acceptance of a ride request.
   6. The system shall allow one rider only, at a time, to accept a ride request.
   7. The system shall allow the user to look up available riders by location.
   8. The system shall record every ride every rider has accepted.
4. Book a ride
   1. The system shall allow the user to look up available riders by location.
   2. The system shall allow the user to set their destination for the ride.
   3. The system should allow the user to choose their preference for the ride.
   4. The system shall allow the user to choose between booking a ride right away or booking a ride in advance.
   5. In case of a right-away ride, the system shall allow the user to cancel the ride before the ride is accepted, without any penalty.
   6. In case of an in-advance booked ride, the system shall allow the user to cancel the ride before the scheduled time, without any penalty.
   7. The system shall record every ride users have requested.
   8. The system shall allow the user to see the price rate of the requested ride.
   9. The system should allow users to see the predicted amount of time for the ride.
5. Initiate a ride
   1. The system shall make notification to the user about the acceptance of their ride request.
   2. The system shall show the user the profile of the rider who accepted their ride request.
   3. The system shall allow the user to be able to see the current location of the rider who accepted their ride request.
   4. The system shall make notification to the user of the arrival of the rider who accepted their ride request.
   5. The system shall allow users to cancel their rides that are currently in progress but with a penalty.
6. Make payment
   1. Before the user books a ride, the service shall allow the user to select their desired payment method.
   2. In case the user decides to pay the service by transferring to a bank account, the system shall allow the user to transfer service fee when the user reaches the destination.
   3. In case the user decides to make a payment automatically from their credit or debit card, the system shall automatically make a payment from that credit or debit card after the rider marks the service as done.
   4. In case the user decides to pay by cash, the system shall deduct the rider’s cash credit equivalent to the service fee for that ride after the rider marks the service as done. (User pays the rider when they reach the destination.)
   5. In case the rider’s cash credit is under 50 baht, the system shall allow rider to top-up credit by bank transfer, credit card, and debit card.
7. Make review
   1. In case the service is success, the system shall allow users that use the service to review their rider via anonymous comment and rate them from 0 to 5 after the ride.
   2. The system should allow the user to view comments and ratings they have given to past rides.
   3. The system should allow the rider to view comments and ratings given to them.

# ER Diagram

[](https://app.diagrams.net/#G1dIbtUo-i9g-JparickuJlgLznPkrMn7I)

click the image for redirect to bigger picture

# Document-based design schema

LOCATION

{

"type": "object",

"bsonType": "object",

"title": "Location",

"description": "Location of customer and station",

"required": [

"\_id", "latitude", "longitude", "sub\_district", "district", "province", "postalcode"

],

"properties": {

"\_id": { "bsonType": "objectid" },

"latitude": { "bsonType": "double", "minimum": -90, "maxmum": 90 },

"longitude": { "bsonType": "double", "minimum": -180, "maxmum": 180 },

"sub\_district": { "bsonType": "string", "minLength": 1, "maxLength": 50 },

"district": { "bsonType": "string", "minLength": 1, "maxLength": 50 },

"province": { "bsonType": "string", "minLength": 1, "maxLength": 50 }

},

"patternProperties": {

"postalcode": {

"bsonType": "string",

"minLength": 1,

"maxLength": 5,

"pattern": "(1-9)(0-9)\*"

}

},

"minProperties": 8,

"dependencies": {

"locatedstation": {

"station": { "bsonType": "object" },

"numberofriders": { "bsonType": "int", "minimum": 0, "maxmum": 100 }

},

"customerSavedAddress": {

"bsonType": "array",

"items": {

"locationname": { "bsonType": "string", "minLength": 1, "maxLength": 50 },

"user": { "bsonType": "objectid" },

"userfirstname": { "bsonType": "string", "minLength": 1, "maxLength": 50 },

"userlastname": { "bsonType": "string", "minLength": 1, "maxLength": 50 },

"userphone": {

"bsonType": "string",

"minLength": 10,

"maxLength": 10,

"pattern": "(0)(1-9)(0-9)\*"

}

},

"minItems": 0,

"uniqueItems": true

}

}

}

# Data Dictionary

**Table of Entities**

|  |  |
| --- | --- |
| **Name** | **Description** |
| Customer | Users who want to find a ride |
| CardInfo | Information of credit card |
| SubscriptionInfo | Information of subscription types |
| Location | Location of customer and station |
| Rider | Users who provide a ride |
| Station | Station of rider |
| Vehicle | Vehicle of rider |
| TransactionRecord | Record of transaction in system |
| Ride | Ride from rider that provide to customer |
| Manager | Person who takes care riders in area |

**Table of Relationships**

|  |  |
| --- | --- |
| **Name** | **Description** |
| CustomerSubscription | Customer subscribes to the system |
| Use | Customer uses credit cards |
| SavedAddress | Customer saves their favourite locations |
| Request | Customers request rides |
| Payment | How the ride is paid |
| Accept | Rider accepts rides |
| Locate | Station is located at location |
| InStation | Riders are in station |
| Own | Rider owns vehicles |
| InCare | Manager takes care of riders |
| BuySubscription | Customer pays their subscription fees |
| TopupCredit | Rider buys Top-Up credits |

Note: Underline attribute stands for primary key and asterisk symbol (\*) stands for foreign key.

Entity Type Name: **Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| user\_ID | CHAR(8) | Customer ID |  | No |
| username | VARCHAR(50) | Username of customer |  | No |
| password | TEXT | Hashed password of customer |  | No |
| email | VARCHAR(100) | Customer Email | Email | No |
| first\_name | VARCHAR(50) | First name of customer |  | No |
| last\_name | VARCHAR(50) | Last name of customer |  | No |
| phone\_num | CHAR(10) | Phone Number | Phone number | No |
| default\_payment | VARCHAR(20) | Default payment of customer | Cash, Bank\_transfer, Credit\_card | No |

Entity Type Name: **CardInfo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| card\_number | CHAR(16) | Card Number | Credit card number | No |
| customer\_ID\* | CHAR(8) | Customer ID |  | No |
| card\_holder\_name | CHAR(16) | Shorten name of card holder |  | No |
| expire\_date | CHAR(4) | Expire date of credit card | Format: “MMYY” | No |

Entity Type Name: **SubscriptionInfo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| subscription\_type | CHAR(8) | Subscription type |  | No |
| name | VARCHAR(30) | Name of subscription type |  | No |
| price | U\_FLOAT | Price of subscription type |  | No |
| trips | U\_INT | Amount of trips |  | No |
| duration | U\_INT | How long until expires (days) |  | No |

Entity Type Name: **Location**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| location\_ID | CHAR(8) | Location ID |  | No |
| latitude | FLOAT | Latitude of location | Values from  -90 to 90 | No |
| longitude | FLOAT | Longitude of location | Values from  -180 to 180 | No |
| sub\_district | VARCHAR(50) | Sub district of location |  | No |
| district | VARCHAR(50) | District of location |  | No |
| province | VARCHAR(50) | Province of location |  | No |
| postal\_code | CHAR(5) | Postal code of location |  | No |

Entity Type Name: **Rider**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| user\_ID | CHAR(8) | Rider ID |  | No |
| station\_ID\* | CHAR(8) | Station ID |  | No |
| ref\_no | CHAR(16) | Reference Number |  | No |
| citizen\_ID | CHAR(13) | Citizen ID | Citizen ID | No |
| is\_available | BOOLEAN | Status available of rider |  | No |
| first\_name | VARCHAR(50) | First name of rider |  | No |
| last\_name | VARCHAR(50) | Last name of rider |  | No |
| phone\_num | CHAR(10) | Phone number of rider | Phone number | No |
| username | VARCHAR(50) | Username of rider |  | No |
| password | TEXT | Hashed password of rider |  | No |
| email | VARCHAR(100) | Rider Email | Email | No |
| rating | U\_FLOAT | Rating of rider | Values from 1.0 to 5.0 | Yes |
| cash\_credit | U\_FLOAT | Cash Credit is what will be deducted when the rider is paid by cash; must be topped-up. |  | No |

Entity Type Name: **Station**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| station\_ID | CHAR(8) | Station ID |  | No |
| location\_ID\* | CHAR(8) | Location ID |  | No |
| manager\_ID\* | CHAR(8) | Manager ID |  | No |
| name | VARCHAR(50) | Station name |  | No |
| number\_of\_riders | U\_INT | Number of riders assigned at the station |  | No |

Entity Type Name: **Vehicle**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| vehicle\_ID | CHAR(8) | Vehicle ID |  | No |
| rider\_ID\* | CHAR(8) | Rider ID |  | No |
| plate\_number | VARCHAR(16) | License plate number of vehicle |  | No |
| model | VARCHAR(50) | Model of vehicle |  | No |
| color | VARCHAR(16) | Color of vehicle |  | No |

Entity Type Name: **TransactionRecord**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| transaction\_ID | CHAR(8) | Transaction ID |  | No |
| user\_ID\* | CHAR(8) | User ID |  | Yes |
| type | CHAR(1) | Type for Transaction Record  t stands for Top-Up  s stands for subscription  r stands for ride | t, s, r | No |
| timestamp | TIME | Timestamp of Transaction |  | No |
| status | VARCHAR(16) | Status of the transaction | Pending, Success, Declined | No |
| amount | FLOAT | Amount of payment | Positive float | No |
| payment\_method | VARCHAR(20) | Payment Method | Cash, Bank\_transfer, Credit\_card | No |

Entity Type Name: **Ride**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| ride\_ID | CHAR(8) | Ride ID |  | No |
| customer\_ID\* | CHAR(8) | Customer ID |  | No |
| rider\_ID\* | CHAR(8) | Rider ID |  | Yes |
| status | VARCHAR(16) | Status of ride |  | No |
| start\_latitude | FLOAT | Latitude which the service has started | Values from -90 to 90 | No |
| start\_longitude | FLOAT | Longitude which the service has started | Values from -180 to 180 | No |
| stop\_latitude | FLOAT | Latitude which the service has finished | Values from -90 to 90 | No |
| stop\_longitude | FLOAT | Longitude which the service has finished | Values from -180 to 180 | No |
| distance | U\_FLOAT | Real distance from start location to stop location |  | No |
| start\_time | TIME | Time the service has started. |  | No |
| stop\_time | TIME | Time the service has finished. |  | Yes |
| review\_rating | BYTE | Rating for this ride | Values from  1 to 5 | Yes |
| review\_comment | TEXT | Comment from customer for this ride |  | Yes |

Entity Type Name: **Manager**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| manager\_ID | CHAR(8) | Manager ID |  | No |
| first\_name | VARCHAR(50) | First name of Manager |  | No |
| last\_name | VARCHAR(50) | Last name of Manager |  | No |
| working\_area | VARCHAR(64) | Working Area of manager |  | No |
| number\_of\_riders | U\_INT | Number of riders under this manager |  | No |

Relationship Type Name: **SavedAddress**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| name | VARCHAR(50) | Name of saved location |  | No |

Relationship Type Name: **CustomerSubscription**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Descriptive Name** | **Valid Values** | **Allow Nulls** |
| start\_date | DATE | Start date of subscription |  | No |
| end\_date | DATE | End date of subscription |  | No |
| trips\_left | U\_INT | Amount of trips left |  | No |