|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **RoseBeam\_Software Requirement Specification\_22May2014** | CM Red Taxi Service Assistant\_SRS\_0.1 | Reviewed | 22-05-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |
| **RoseBeam\_Software Requirement Specification\_16\_JUNE\_2014** | CM Red Taxi Service Assistant\_ SRS \_0.2 | Reviewed | 16-06-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |
| **RoseBeam\_Software Requirement Specification\_27 JUNE 2014\_** | CM Red Taxi Service Assistant\_ SRS \_0.3 | Reviewed | 27-06-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |
| **RoseBeam\_Software Requirement Specification\_05 JULY 2014** | CM Red Taxi Service Assistant\_ SRS \_0.4 | Reviewed | 05-07-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |
| **RoseBeam\_Software Requirement Specification\_09 JULY 2014** | CM Red Taxi Service Assistant\_ SRS \_0.5 | Reviewed | 09-07-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |
| **RoseBeam\_Software Requirement Specification\_24 JULY 2014** | CM Red Taxi Service Assistant\_ SRS \_0.6 | Reviewed | 24-07-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |
| **RoseBeam\_Software Requirement Specification\_31 JULY 2014** | CM Red Taxi Service Assistant\_ SRS \_1.0 | Reviewed | 31-07-2014 | PM, SA, DEV, QA, Tester, Advisor | Kanittee,  Pimchittra | Kanittee,  Pimchittra |

**The Chiang Mai Red Taxi**

**Service Assistant**

**Software Requirement Specification**

**By**

**Miss. Kanittee Hongron 542115003**

**Miss. Pimchittra Sukkasem 542115042**

**Department of Software Engineering**

**College of Arts, Media and Technology**

**Chiang Mai University**

**Project Advisor**

**Ms. Yun Rim Park**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Table of Contents

Chapter 1: Introduction5  
1.1 Purpose5  
1.2 Software Scope5  
1.3 Operating Environment 7  
1.4 Definition, Acronyms and Abbreviation 8  
 1.4.1 Key Definitions8  
 1.4.2 Key Acronyms and Abbreviations 8

Chapter 2: Overall Description and Requirements9  
2.1 Product Perspective9  
2.2 User Characteristics9  
2.3 Features10  
2.4 Functional & User Requirements11

Chapter 3: Specific Requirements16  
3.1 Use Case Diagram16  
 3.1.1 Use Case Diagram Overview 16  
 3.1.2 Passenger Use Case Diagram 17  
 3.1.3 Driver Use Case Diagram 18  
 3.1.4 Administrator’s Use Case Diagram 19

Chapter 4: Use Case Descriptions20  
4.1 Passenger can register to the system (URS-01) 20  
4.2 Passenger can login to the system (URS-02) 22  
4.3 Passenger can logout of the system (URS-03) 24  
4.4 Passenger can search for taxi (URS-04) 26  
4.5 Passenger can send request for taxi (URS-05) 28  
4.6 Passenger can chat with driver (URS-06) 30  
4.7 Driver can register into the system (URS-07)32  
4.8 Driver can log in to the system (URS-08)34  
4.9 Driver can logout from the system (URS-09)36  
4.10 Driver can update driving information (URS-10)38  
4.11 Driver can respond to passenger’s request (URS-11)40  
4.12 Driver can chat with passenger (URS-012)42  
4.13 Administrator can login to the Administration system (URS-013) 44  
4.14 Administrator can logout (URS-14) 46  
4.15 Administrator can add destinations (URS-15)48  
4.16 Administrator can browse the destination (URS-16)50  
4.17 Administrator can edit destinations (URS-17)52  
4.18 Administrator can delete destinations (URS-18)54  
4.19 Administrator can search destinations (URS-19) 56  
4.20 Administrator can clear data (URS-20)58

Chapter One | Introduction

* 1. **Purpose**

This Software Requirement Specification (SRS) is used to explain and describe the functions in the Chiang Mai Red Taxi Service Assistant, which is a mobile application that allows red taxi drivers to find potential passengers and vice versa, in a convenient and efficient manner. This document aims to guide and allow both developers to develop this application and for users to better understand features and functions in the system using instructional diagrams, explaining design constraints and solutions, and also elaborating interface details.

**1.2 Software Scope**

The Chiang Mai Red Taxi Service Assistant will be developed using the Android operating system. There are four core features in the Chiang Mai Red Taxi Service Assistant namely, web service, Google maps, and the Android operating system on both driver’s and passenger’s systems. In this application, the Driver has to report his travel conditions and current location to the server. The Passenger will then detect his current location from Google Maps, and send the location and travel conditions (e.g. the number of passengers and the destination) to the server. Consequently, the service finds the taxi drivers near the Passenger. The web service will process the matching conditions and send the information of the available red taxis back to the Passenger. Thereafter, the Passenger can send a request to one of the available taxis. The Driver that receives the request can either accept or decline the request to the Passenger though the web service.

The features of software the Chiang Mai Red Taxi Service Assistant is divided into two categories that consist of the main features and sub-features. The main features are:

• Passengers can register to the system.

• Passengers can log into the system.

• Passengers can set their destination and the number of passengers travelling.

• Passengers can send a request to a driver.

• Passengers can receive confirmation if the driver accepts the request.

• Passengers can view the booking information of red taxi.

• Passengers can view the current location of the red taxi that match their conditions.

• Passengers can log out of the system.

• Drivers can register to the system.

• Drivers can log into the system

• Drivers can update the availability of seats.

• Drivers can receive requests from passengers.

• Drivers can accept or decline the request.

• Drivers can log out of the system.

• Administrator can login to the Administration system

• Administrator can logout from the Administration system

• Administrator can add destinations to system

• Administrator can browse the destination list

• Administrator can edit destinations in the system

• Administrator can delete destinations from the system

• Administrator can search for destinations in the system

The sub features of the application are:   
  
• Passengers can create schedules and plan their routes.

• Passengers can change the scope of searching for red taxis.

• Passengers can cancel the request.

• Passengers can receive notifications when the red taxi arrives.

• Drivers can enable or disable the service (e.g. off duty).

• Drivers can be notified when the request is cancelled.

However, there are some limitations with the Chiang Mai Red Taxi Service:

* The application requires an Internet connection.
* The application can only be used on Android smartphones that run on Android OS 4.3 or later.
* The application cannot be supported on an Android tablet.
* Drivers have to preset a general direction of driving using their mobile application before leaving.
* Only passengers can send a request to drivers (one driver per request).
* Drivers cannot locate passengers unless there is a request from them.
* Services are limited to the passengers and drivers in Chiang Mai only.
* Services are limited to the registered taxis and passengers only.

(Registration is required before using the application).

This application aims to provide passengers with a more convenient and efficient method of getting a red taxi in Chiang Mai and also to reduce exisiting public transportation problems faced in Chiang Mai by creating a network of red taxis and offering passenger-driver matching services.

**1.3 Operating Environment**

The Chiang Mai Red Taxi Service is developed on the Android operating system. Users of this application are required to have an active Internet connection and have an Android smart phone that runs on Android OS 4.3 or later.

|  |  |
| --- | --- |
| **Technology Used** | |
| Android Operating System | Version 4.3 or later |
| Google Maps | Android API |
| JSON | - |
| Java | - |
| MySQL | - |
| Yii Framework | PHP framework |
| XAMPP | Opensource webserver package |

**1.4 Definition, Acronyms and Abbreviation   
  
1.4.1 Key Definitions**

|  |  |
| --- | --- |
| **IEEE** | Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and for computer scientists. [IEEE90] |
| **Feature** | Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a product in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90] |
| **Requirement** | 1. A condition or capability needed by a user to solve a problem or achieve an objective. 2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. 3. A documented representation of a condition or capability as in definition (1) or (2). [IEEE90] |
| **Use Case** | 1. Concept to describe a system based on usage of system resources by its environment. Characterized by an objective set of interactions within and at the borders of that system. 2. Notation from UML for describing a scenario (Usage approach, operational scenario) from the perspective of its users. [IEEE90] |
| **Specification** | Precise description of an activity or work product which serves as basis or input for further activities or work product. A specification can comprise requirements to a product and how they will be solved. Different parts of a specification (e.g., what is to be done, how it will be done) must not be mixed. [IEEE90] |
| **Software Engineering** | The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. [IEEE90] |

**1.4.2 Key Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **URS-XX** | User Requirement Specification number XX |
| **SRS-XX** | Software Requirement Specification number XX |
| **AD-XX** | Activity Diagram number XX |
| **UC-XX** | Use Case Description number XX |

**Chapter Two | Overall Description & Requirement**

**2.1 Product Perspective**

The Chiang Mai Red Taxi Service Assistant will be developed on the Android operating system and it will require an active Internet connection. It is a web service that will assist users in getting a red taxi and aims to reduce transportation issues in Chiang Mai. The system will integrate Google-Maps that can show the current locations of drivers and passengers on the map in real-time.

**2.2 User Characteristics**

The Chiang Mai Red Taxi Service Assistant is an application that offers efficient passenger-driver matching services that will provide passengers with a more convenient method of getting a red taxi in Chiang Mai. The Chiang Mai Red Taxi Service Assistant is aimed at assisting red taxi drivers in receiving notifications and booking requests from potential passengers and also helping passengers track, locate and book the services of red taxis in the vicinity.

* Driver: Red Taxi Drivers that own an Android smartphone and are able to use 3G connection when driving on the road.
* Passenger: Users that own an Android smartphone with an active 3G connection and require transportation from one destination to another.
* Administrator: Administrator that can access the system’s database to update or remove information.

**2.3 Features**

**Main features**

• Passenger can logout of the system

• Passenger can register to the system

• Passenger can login to the system

• Passenger can search for taxi

• Passenger can send request for taxi

• Driver can logout from the system

• Driver can update driving information

• Driver can respond to passenger’s request

• Driver can chat with passenger

• Administrator can login to the Administration system

• Administrator can logout

• Administrator can add destinations

• Administrator can browse the destination

• Administrator can edit destinations

• Administrator can delete destinations

• Administrator can search destinations

• Administrator can clear data

**Sub features**

• Passengers can cancel the request.

• Passengers can get the notification when the red taxi arrives.

• Passengers can create schedules (for planning of routes).

• Passengers can change the scope of searching for red taxis.

• Drivers can receive the cancellation request.

• Drivers can choose to enable or disable the service (e.g. off duty).

**2.4 Functional & User Requirements**

**2.4.1 Passenger can register to the system (URS-01)**SRS-01 System shall display the registration form for the passenger.

SRS-02 System shall ensure no duplication of information by validating the passenger registration.

SRS-03 System shall create the new Passenger in the database.

SRS-04 System shall display the message “Registration complete” and the button “Go to login...” for link to login page.

SRS-05 System shall display the message in sequence “Enter your \_\_\_\_”, if the input is equals to null.

**2.4.2 Passenger can login to the system (URS-02)**SRS-07 System shall display the Passenger login page.

SRS-08 System shall ensure no duplication of information by validating the phone number and password.

SRS-09 System shall update the Passenger’s status to “Online” in the database.

SRS-10 System shall display the message “Welcome+Name of Passenger!”

SRS-11 System shall display the Home page that provides the search function

SRS-12 System shall display the map that shows the passenger’s current location.

SRS-13 System shall display message “This username or password is wrong” if the Passenger inputs the username or password incorrectly.

**2.4.3 Passenger can logout of the system (URS-03)**SRS-14 System shall change status of the Driver to offline in database.

SRS-15 System shall display the Passenger login page after passenger logs out.

**2.4.4 Passenger can search for taxi (URS-04)**

SRS-16 System shall display the map that shows the current location of red taxi from the search result.

SRS-17 System shall process the search conditions – (eg: number of seats and destination headed to) together with the taxi driving information which is stored in the database.

SRS-18 System shall provide the results from the search that matches the conditions.

SRS-19 System shall display a list of up to 10 search results.

**2.4.5 Passenger can send request for taxi (URS-05)**

SRS-20 System shall display the pop up message to show Driver’s details and request for confirmation to send the request.

SRS-21 System shall send a request containing the conditions and taxi information to the server and save to database.

SRS-22 System shall display the request information on a page after the passenger sent request. (Including the request’s status and the chat box)

**2.4.6 Passenger can chat with driver (URS-06)**SRS-23 System shall save the message that Passenger enters in the chat box to database.

SRS-24 System shall display the chat messages between Driver and Passenger.   
  
**2.4.7 Driver can register into the system (URS-07)**   
SRS-25 System shall display the registration form for the Driver.

SRS-26 System shall ensure no duplication of information by validating the Driver registration.

SRS-27 System shall create the new Driver in the database.

SRS-28 System shall display the message “Driver Registration complete”.

SRS-29 System shall display the button “Go to login...” for link to the login page.

SRS-30 System shall display message ‘Wrong format!’ if Driver inputs the wrong format.

SRS-31 System shall display message “This username already exists” if Driver inputs an existing username.

SRS-32 System shall display message “This license ID already exists.” if the Driver inputs an existing license ID.

**2.4.8 Driver can log in to the system (URS-08)**

SRS-33 System shall display the Driver’s login page.

SRS-34 System shall ensure no duplication of information by validating the phone number and password.

SRS-35 System shall update the Driver’s status to “Online” in the database.

SRS-36 System shall display the message “welcome+ name of driver”

SRS-37 System shall display the Home page that provide the updated driving information function and the map that can view the Driver’s current location.

SRS-38 System shall display the message ‘This username or password is wrong’ if the driver inputs the username or password incorrectly.  
  
**2.4.9 Driver can logout from the system (URS-09)**   
SRS-39 System shall change status of the Driver to “offline” in the database.

SRS-40 System shall display login page after Driver logs out.  
  
**2.4.10 Driver can update driving information (URS-10)**   
SRS-41 System shall update the driving information into database.

SRS-42 System shall display the page that shows all requests from passenger.  
  
**2.4.11 Driver can respond to passenger’s request (URS-11)**

SRS-43 System shall display pop up that request to confirm or decline the Passenger’s request.

SRS-44 System shall update the status of the request to “Accepted” if the Driver clicks “Accept”.

SRS-45 System shall display message ‘Request Accepted’ to the Driver.

SRS-46 System shall display message ‘Accepted by Driver’ to the Passenger.  
  
**2.4.12 Driver can chat with passenger (URS-012)**   
SRS-47 System shall save the message that Driver enters in the chat box to database.

SRS-48 System shall display the chat messages between Driver and Passenger.  
  
**2.4.13 Administrator can login to the Administration system (URS-013)**

SRS-49 System shall display the login interface to Administrator.

SRS-50 System shall verify Administrator username & password from database.

SRS-51 System shall display Administration homepage after successful verification.

SRS-52 System shall display the login interface, if Administrator enters username or password wrongly.

**2.4.14 Administrator can logout (URS-14)**   
SRS-53 System shall process Administrator logging out.

SRS-54 System shall display the Administrator login page after logging out.  
  
**2.4.15 Administrator can add destinations (URS-15)**   
SRS-55 System shall display interface for adding new destination’s information.

SRS-56 System shall create and save the new destination in database.

SRS-57 System shall display updated destination lists.

SRS-58 System shall display the destination page and save the information to database.  
  
**2.4.16 Administrator can browse the destination (URS-16)**   
SRS-59 System shall retrieve data from database.

SRS-60 System shall display the list of destinations.

**2.4.17 Administrator can edit destinations (URS-17)**SRS-61 System shall display interface for editing destination’s information.

SRS-62 System shall update destination’s information to database.

SRS-63 System shall display updated destination lists.

SRS-64 System shall display the destination page with updated information to database.  
 **2.4.18 Administrator can delete destinations (URS-18)**

SRS-65 System shall display the pop up message “Do you want to DELETE selected item? Delete or Cancel”.

SRS-66 System shall display updated destination lists.

SRS-67 System shall remove that destination from database.

SRS-68 System shall not remove that destination from database.  
  
**2.4.19 Administrator can search destinations (URS-19)**   
SRS-69 System shall process searching that keyword with database.

SRS-70 System shall display searching destinations result in a list.

SRS-71 System shall display “not found” if destination not stored in the database.

**2.4.20 Administrator can clear data (URS-20)**

SRS-72 System shall display updated table.

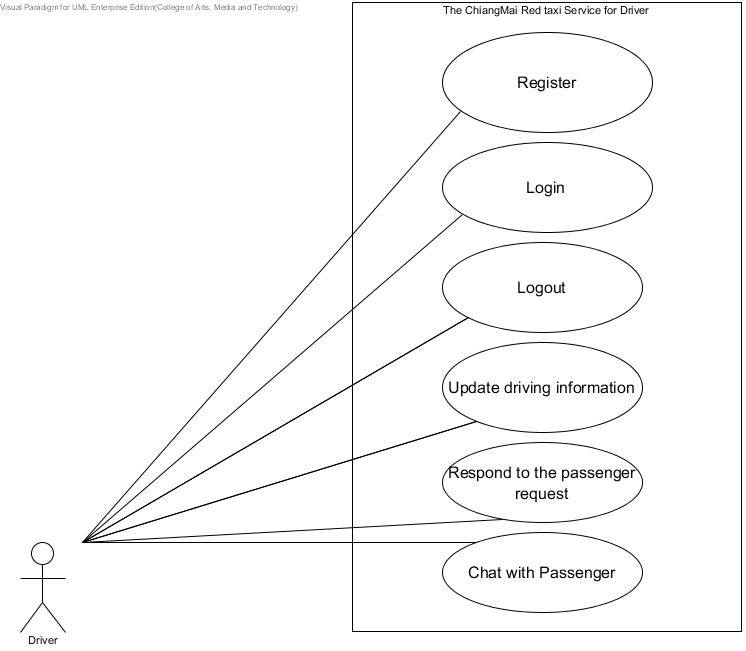
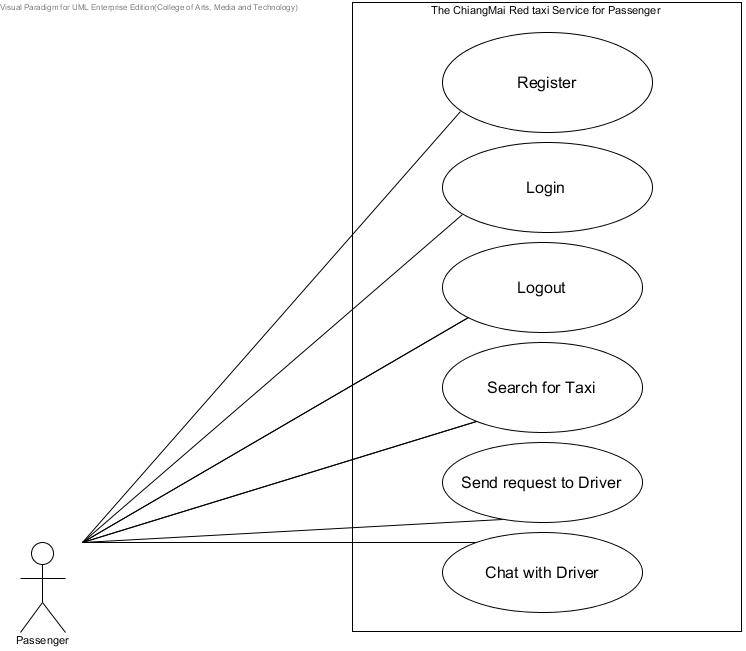
SRS-73 System shall ask for confirmation to clear data.

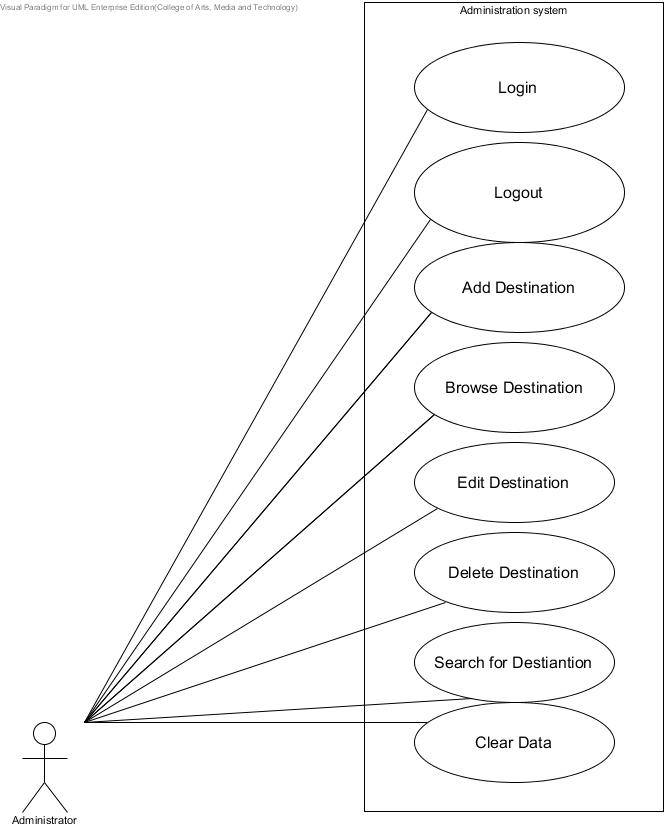
SRS-74 System shall display the table that shows the name of the table that Administrator can delete.

**Chapter Three | Specific Requirement**s

**3.1 Use Case Diagram**

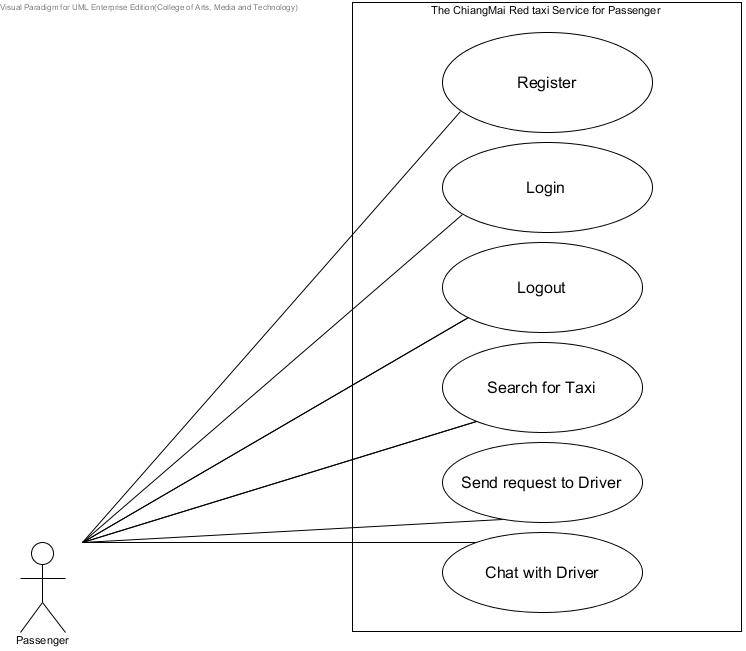
**3.1.1 Use Case Diagram Overview**





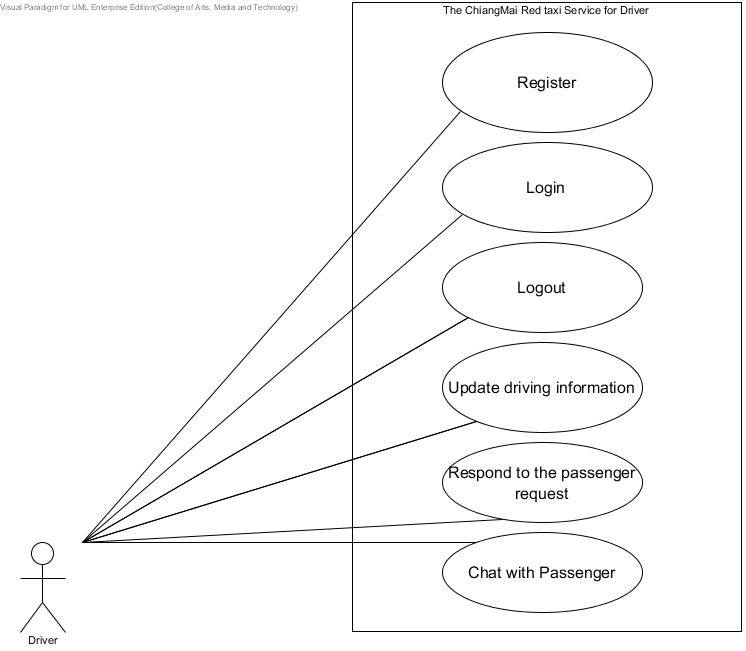
*Figure 1: Overview diagram of the System’s use case.*

**3.1.2 Passenger Use Case Diagram**



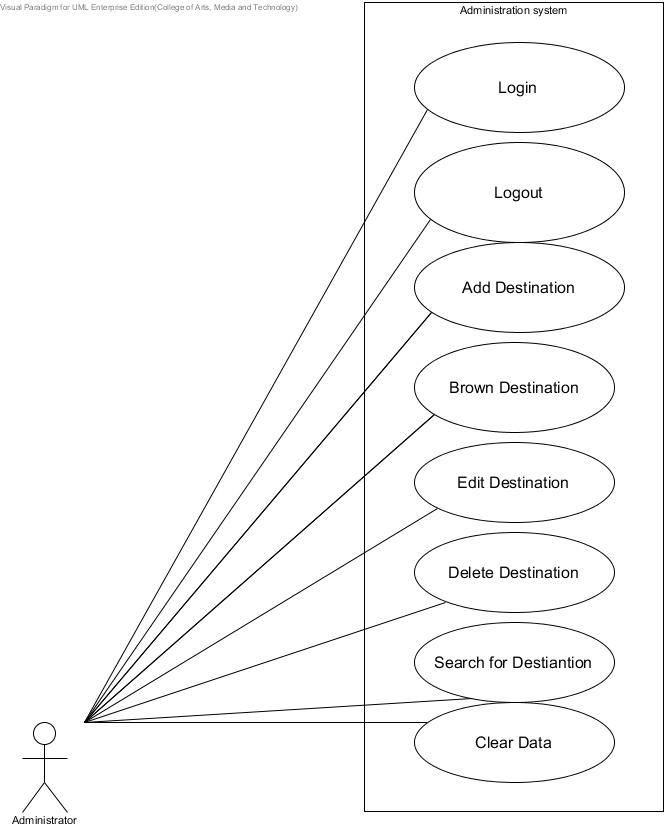
*Figure 2: PassengerUse Case Diagram.*

**3.1.3 Driver Use Case Diagram**

****

*Figure 3: Driver Use Case Diagram.*

**3.1.4 Administator’s Use Case Diagram**

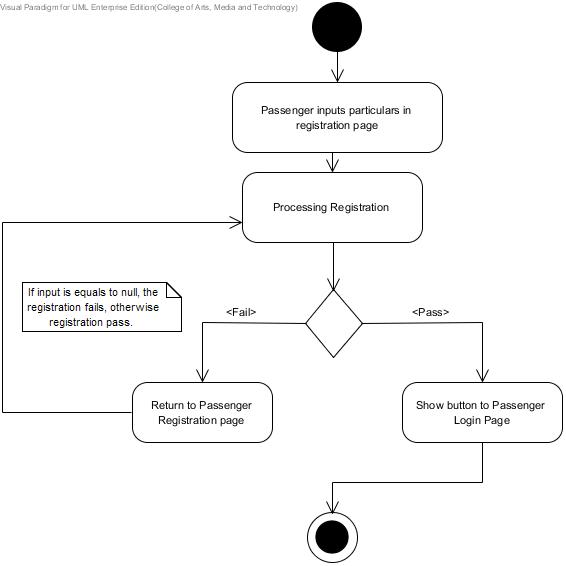


*Figure 4: Administator Use Case Diagram.*

**Chapter Four | Use Case Descriptions**

4.1 Passenger can register to the system (URS-01)

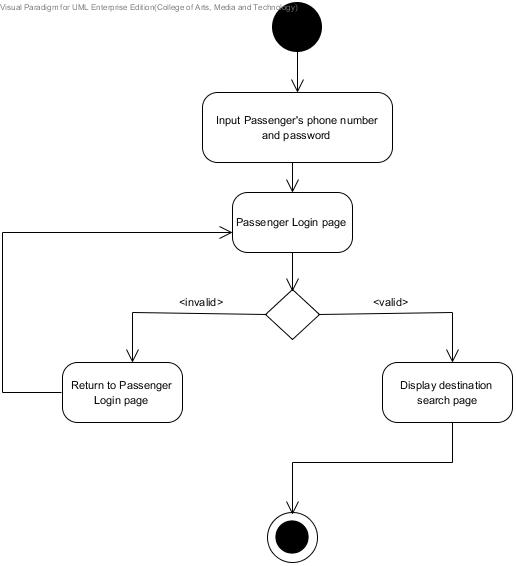
|  |  |
| --- | --- |
| Use Case ID | URS-01 |
| Use Case name: | Passenger can register to the system |
| Actors: | Passenger |
| Description: | Passenger can register themselves as a ‘Passenger’. |
| Trigger: | Passenger clicks on the register button when starting the application. |
| Pre-conditions: | N/A |
| Post-conditions: | A user account has been created for Passenger.  Passenger can log into the system. (URS-02) |
| Normal Flows: | 1. Passenger clicks on the register button when starting the application. 2. System shall display the registration form for the passenger. 3. Passenger inputs their particulars.  * Name * ID card * Mobile * Password  1. System shall ensure no duplication of information by validating the passenger registration. 2. System shall create the new Passenger in the database. 3. System shall display the message “Registration complete” and the button “Go to login ..” for link to login page. |
| Alternative Flows: | N/A |
| Exceptions: | 4A.   1. System shall display the message in sequence “Enter your \_\_\_\_\_”, if the input is equals to null. 2. Resume 3rd step in Normal Flows. |
| Includes: |  |
| Note and Issues: |  |



*Figure5: AD-* *Passenger can register to the system (URS-01)*

4.2 Passenger can login to the system (URS-02)

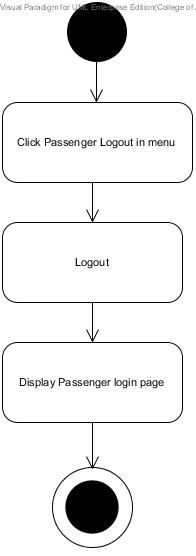
|  |  |
| --- | --- |
| Use Case ID | URS-02 |
| Use Case name: | Passenger can login to the system |
| Actors: | Passenger |
| Description: | Passenger can log in to the system for accessing the system as a ‘Passenger’. |
| Trigger: | Passenger clicks on log in button |
| Pre-conditions: | Passenger registered in the system. (URS-01) |
| Post-conditions: | * Passenger is logged in and ready to perform actions that require user authentication. * Passenger can log out. (URS-03) |
| Normal Flows: | 1. Passenger accesses the application. 2. System shall display the Passenger login page. 3. Passenger inputs the phone number and password. 4. System shall ensure no duplication of information by validating the phone number and password. 5. System shall update the Passenger’s status to “Online” in the database. 6. System shall display the message “Welcome+Name of Passenger!” 7. System shall display the Home page that provides the search function 8. System shall display the map that shows the passenger’s current location. |
| Alternative Flows: | N/A |
| Exceptions: | 4A.  1. System shall display message “This username or password is wrong” if the Passenger inputs the username or password incorrectly.  2. Resume 3rd step in Normal Flows. |
| Includes: |  |
| Note and Issues: |  |



*Figure6: AD-* *Passenger can login to the system (URS-02)*

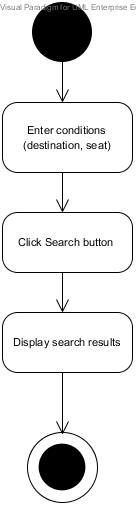
4.3 Passenger can logout of the system (URS-03)

|  |  |
| --- | --- |
| Use Case ID | URS-03 |
| Use Case name: | Passenger can log out from the system. |
| Actors: | Passenger |
| Description: | Passenger can logout from the system |
| Trigger: | Passenger clicks on the “logout” button in the settings page. |
| Pre-conditions: | Passenger is logged into the Chiang Mai Red Taxi Service Assistant as a “Passenger” (URS-02). |
| Post-conditions: | - Passenger is logged out and cannot perform the actions that require user  - Passenger can log into the system (URS-02). |
| Normal Flows: | 1. Passenger clicks on the “logout” button in the settings page. 2. System shall change status of the Driver to offline in database. 3. System shall display the Passenger login page after passenger logs out. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



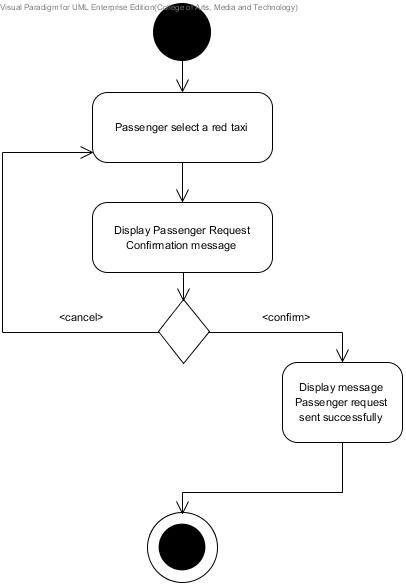
*Figure7: AD- Passenger can logout of the system (URS-03)*4.4 Passenger can search for taxi (URS-04)

|  |  |
| --- | --- |
| Use Case ID | URS-04 |
| Use Case name: | Passenger can search for the red taxi which matches the conditions. |
| Actors: | Passenger |
| Description: | Passenger can input their condition,then search for the red taxi which is nearest to the user and matches their condition. |
| Trigger: | Passenger inputs conditions, then click “search”. |
| Pre-conditions: | * Passenger logs into the Chiang Mai red taxi service assistant as a “passenger” (URS-02). * System shall have at least one driving information of the red taxi. |
| Post- Condition | * Passenger can send request to Driver (URS-05). |
| Normal Flows: | 1. Passenger inputs conditions – i.e.: number of seats and destination headed to. 2. Passenger selects the ‘search’ button to proceed with searching for a red taxi. 3. System shall process the search conditions – (eg: number of seats and destination headed to) together with the taxi driving information which is stored in the database. 4. System shall provide the results from the search that matches the conditions. 5. System shall display a list of up to 10 search results. |
| Alternative Flows: | 3A. Passenger can view the current location of red taxi from the searching result.  1. Passenger clicks “View on map” button.  2. System shall display the map that shows the current location of red taxi from the search result. |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure8: AD- Passenger can search for taxi (URS-04)*4.5 Passenger can send request for taxi (URS-05)

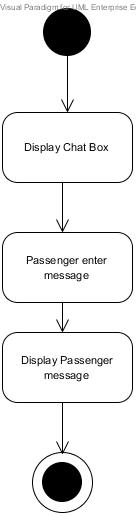
|  |  |
| --- | --- |
| Use Case ID | URS-05 |
| Use Case name: | Passenger can send a request to a driver. |
| Actors: | Passenger |
| Description: | Passenger can send a request to the driver after searching for the taxi. |
| Trigger: | Passenger clicks “send request” button in the driving information page. |
| Pre-conditions: | * Passenger has found the red taxi which matches her conditions (URS-04). |
| Post-conditions: | * Passenger views that request information on a page after the passenger send request. * Passenger views the current location of the red taxi on the map (ProgressII) * Driver views the request that the passenger sent (URS-11). * Passenger views the request information on a page after sending the request. (Including the request’s status and the chat box) (URS-05) |
| Normal Flows: | 1. Passenger selects on the red taxi that required in the result list. 2. System shall display the pop up message to show driver’s details and request for confirmation to send the request. 3. System shall send a request containing the conditions and taxi information to the server and save to database. 4. System shall display the request information on a page after the passenger sent request. (Including the request’s status and the chat box) |
| Alternative Flows: |  |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure9: AD- Passenger can send request for taxi (URS-05)*

4.6 Passenger can chat with driver (URS-06)

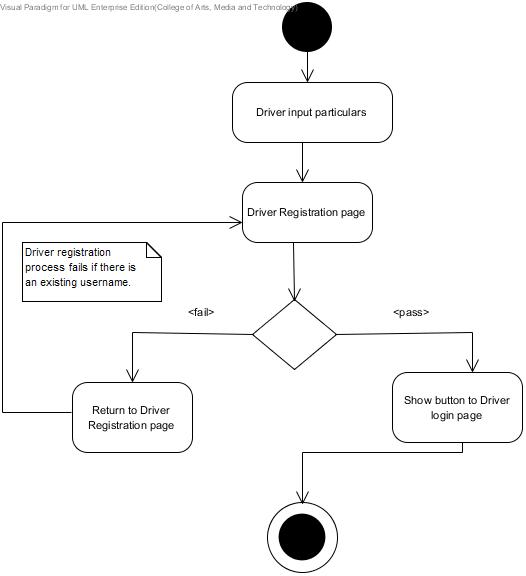
|  |  |
| --- | --- |
| Use Case ID | URS-06 |
| Use Case name: | Passenger can chat with driver |
| Actors: | Passenger, Driver |
| Stakeholders and Interests: |  |
| Description: | Passenger can chat with the driver after sending the request. This can help both Driver & Passenger negotiate the price or asking questions. |
| Trigger: | The request has been sent. |
| Pre-conditions: | * Passenger has sent the message to Driver.(URS-06) |
| Post-conditions: | N/A |
| Normal Flows: | 1. Passenger inputs a message in to the chat box which will be displayed after Passenger send request to Driver, then clicks on the “send” button. 2. System shall save the message that Passenger enters in the chat box to database. 3. System shall display the chat messages between Driver and Passenger. |
| Alternative Flows: |  |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure10: AD- Passenger can chat with driver (URS-06)*

4.7 Driver can register into the system (URS-07)

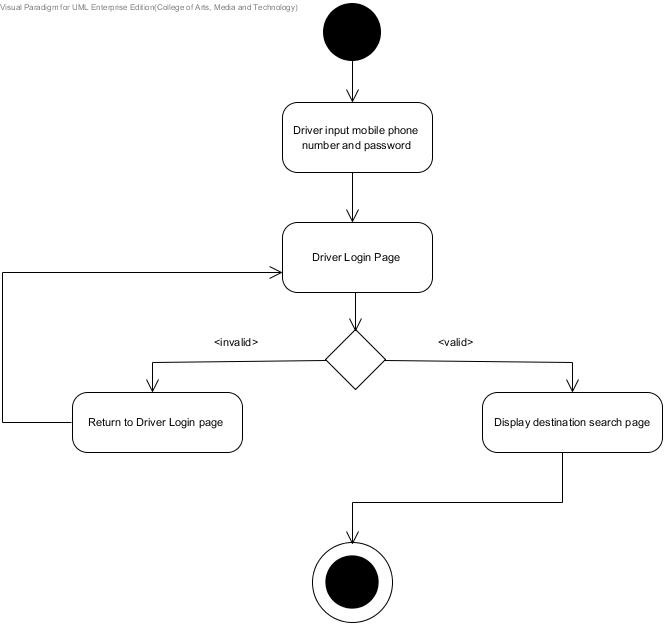
|  |  |
| --- | --- |
| Use Case ID | URS-07 |
| Use Case name: | Driver can register to the system |
| Actors: | Driver |
| Description: | Driver can register themselves as a ‘Red taxi Driver’. |
| Trigger: | Driver clicks on the “register” button when starting the application. |
| Pre-conditions: | N/A |
| Post-conditions: | Driver can log into the system. (URS-08) |
| Normal Flows: | 1. Driver clicks on the register button when starting the application. 2. System shall display the registration form for the Driver. 3. Driver inputs their particulars.   - Name  - ID card  - Mobile Number  - Password   1. System shall ensure no duplication of information by validating the Driver registration. 2. System shall create the new Driver in the database. 3. System shall display the message “Driver Registration complete”. 4. System shall display the button “Go to login ..” for link to the login page. |
| Alternative Flows: | N/A |
| Exceptions: | 4A.   1. System shall display message ‘Wrong format!’ if Driver inputs the wrong format. 2. Resume 3rd step in Normal Flows.   4B.   1. System shall display message “This username already exists” if driver inputs an existing username. 2. Resume 3rd step in Normal Flows.   4C.   1. System shall display message “This license ID already exists.” if the driver inputs an existing license ID. 2. Resume 3rd step in Normal Flows. |
| Includes: |  |
| Note and Issues: |  |



*Figure11:AD- Driver can register into the system (URS-07)*

4.8 Driver can log in to the system (URS-08)

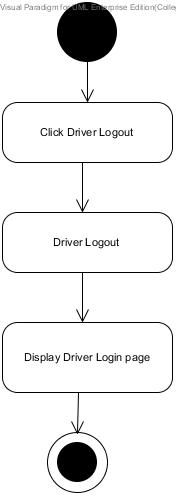
|  |  |
| --- | --- |
| Use Case ID | URS-08 |
| Use Case name: | Driver can log in to the system |
| Actors: | Driver |
| Description: | Driver can log in to the system for accessing the system as a ‘Red Taxi Driver’. |
| Trigger: | Driver clicks on ‘Driver log in’ button in the login page. |
| Pre-conditions: | Driver is registered in the system. (URS-07) |
| Post-conditions: | * Driver can log out. (URS-09) * Driver can access the application |
| Normal Flows: | 1. Driver starts application. 2. System shall display the Driver login page. 3. Driver inputs the phone number and password. 4. System shall ensure no duplication of information by validating the phone number and password. 5. System shall update the Driver’s status to “Online” in the database. 6. System shall display the message “welcome+ name of Driver” 7. System shall display the Home page that provide the updated driving information function and the map that can view the Driver’s current location. |
| Alternative Flows: | N/A |
| Exceptions: | 4A.   1. System shall display the message ‘This username or password is wrong’ if the driver inputs the username or password incorrectly. 2. Resume 3rd step in Normal Flows. |
| Includes: |  |
| Note and Issues: |  |



*Figure12: AD- Driver can log in to the system (URS-08)*

4.9 Driver can logout from the system (URS-09)

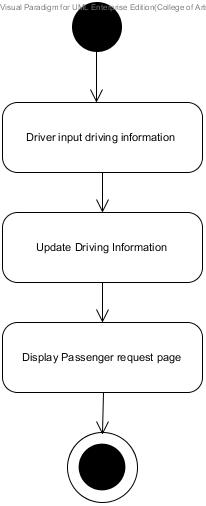
|  |  |
| --- | --- |
| Use Case ID | URS-09 |
| Use Case name: | Driver can log out from the system. |
| Actors: | Driver |
| Description: | Driver can logout from the system when the driver stops using the service. |
| Trigger: | The Driver clicks on the “logout” button in the settings page. |
| Pre-conditions: | The Driver logs into the Chiang Mai Red Taxi Service Assistant as a “Driver”. (URS-08) |
| Post-conditions: | Driver can log into the system (URS-08) |
| Normal Flows: | 1. Driver clicks logout button. 2. System shall change status of the Driver to “offline” in the database. 3. System shall display login page after Driver logs out. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure13: AD- Driver can logout from the system (URS-09)*

4.10 Driver can update driving information (URS-10)

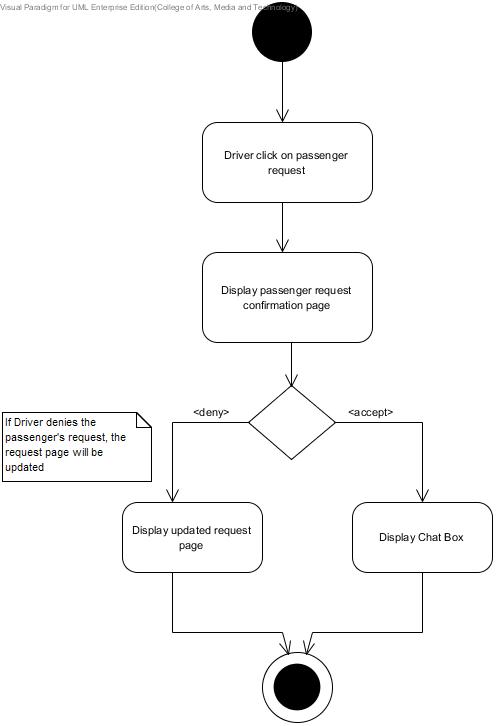
|  |  |
| --- | --- |
| Use Case ID | URS-10 |
| Use Case name: | Driver can update driving information |
| Actors: | Driver |
| Description: | Driver can update driving information for the system to process matching taxi and passenger. The information would be visible to the passenger. |
| Trigger: | Driver click on “Update” button |
| Pre-conditions: | * Driver logs into the Chiang Mai Red Taxi Service Assistant as a “Driver”. (URS-10) * Driver’s driving information has been provided and exists in the system. |
| Post-conditions: | * Passengers can view the “view information” of red taxi that matches their conditions. (URS-05) |
| Normal Flows: | 1. Driver opens the existing driving information in the system. 2. Driver updates driving information (E.g.; number of available seats, destination) 3. Driver clicks on “update” button 4. System shall update the driving information into database. 5. System shall display the page that shows all requests from passenger. |
| Alternative Flows: | N/A |
| Exceptions: |  |
| Includes: |  |
| Note and Issues: |  |



*Figure14: AD- Driver can update driving information (URS-10)*

4.11 Driver can respond to passenger’s request (URS-11)

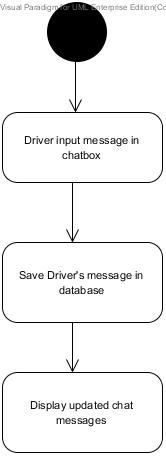
|  |  |
| --- | --- |
| Use Case ID | URS-11 |
| Use Case name: | Driver can respond to passenger’s request |
| Actors: | Driver |
| Description: | Driver can respond to the passenger’s request by choosing to accept or decline the request. |
| Trigger: | * Driver clicks on each request in the “request page”. |
| Pre-conditions: | * Passenger’s request has already been sent to the Driver. (URS-05) * The request status is listed as “new”. |
| Post-conditions: | * Passenger receives Driver’s response. (URS-11) |
| Normal Flows: | 1. Driver clicks on each request in the “request page”. 2. System shall display pop up that request to confirm or decline the Passenger’s request. 3. Driver clicks “yes” to accept the request. 4. System shall update the status of the request to “Accepted” if the Driver clicks “Accept”. 5. System shall display message ‘Request Accepted’ to the Driver. 6. System shall display message ‘Accepted by Driver’ to the Passsenger. |
| Alternative Flows: | 3A.  1. If the Driver denies the request, the request will disappear from the “request page”. |
| Exceptions: | If the Driver does not respond to the new request within 60 seconds, the request will disappear. |
| Includes: |  |
| Note and Issues: |  |



*Figure15: AD- Driver can respond to passenger’s request (URS-11)*

4.12 Driver can chat with passenger (URS-012)

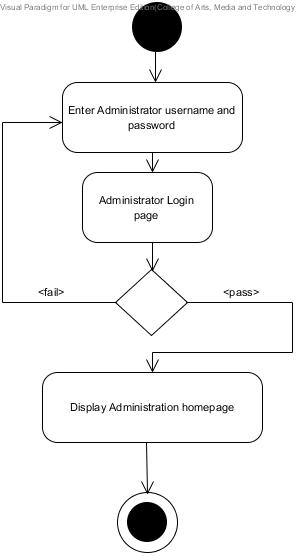
|  |  |
| --- | --- |
| Use Case ID | URS-12 |
| Use Case name: | Driver can chat with passenger |
| Actors: | Driver, Passenger |
| Description: | Driver can chat with the passenger after sending the request. This can help both Driver & Passenger negotiate the price or asking any questions related to the destination. |
| Trigger: | Driver selects and clicks on Passenger chatbox |
| Pre-conditions: | * There must be at least one request sent from Passenger to Driver in database. |
| Post-conditions: | N/A |
| Normal Flows: | 1. Driver inputs the message in to the chat box, then clicks on the “send” button. 2. System shall save the message that Driver enters in the chat box to database. 3. System shall display the chat messages between Driver and Passenger. |
| Alternative Flows: |  |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure16: AD- Driver can chat with passenger (URS-012)*

4.13 Administrator can login to the Administration system (URS-013)

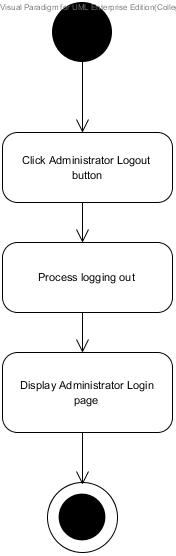
|  |  |
| --- | --- |
| Use Case ID | URS-13 |
| Use Case name: | Administrator can login to the Administration system. |
| Actors: | Administrator |
| Description: | Administrator can login to the Administration system to manage the destination. |
| Trigger: | Administrator accesses [http://theredtaxiservice.com/admin](http://theredtaxiservice.com/admin%20) |
| Pre-conditions: | Administrator’s information (username&password) already exists in database. |
| Post-conditions: | Administrator can browse the destination. |
| Normal Flows: | 1. Administrator accesses website [http://theredtaxiservice.com/admin](http://theredtaxiservice.com/admin%20) 2. System shall display the login interface to Administrator. 3. Administrator enters username & password. 4. Administrator clicks “login”. 5. System shall verify Administrator username & password from database. 6. System shall display Administration homepage after successful verification. |
| Alternative Flows: | N/A |
| Exceptions: | 4A.  1. System shall display the login interface, if Administrator enters username or password wrongly.  2. Resume 3rd step in Normal Flows. |
| Includes: |  |
| Note and Issues: |  |



*Figure17: AD- Administrator can login to the Administration system (URS-013)*

4.14 Administrator can logout (URS-14)

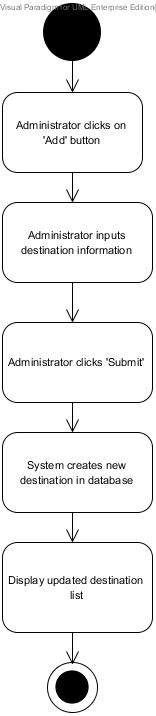
|  |  |
| --- | --- |
| Use Case ID | URS-14 |
| Use Case name: | Administrator can logout. |
| Actors: | Administrator |
| Description: | Administrator can logout from the Administration site. |
| Trigger: | Administrator clicks the button “logout” |
| Pre-conditions: | Administrator can log in to the system. |
| Post-conditions: | Administrator can log in to the system. |
| Normal Flows: | 1. Administrator clicks the button “logout”. 2. System shall process Administrator logging out. 3. System shall display the Administrator login page after logging out. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure18: AD- 4.14 Administrator can logout (URS-14)*

4.15 Administrator can add destinations (URS-15)

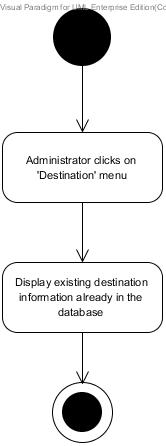
|  |  |
| --- | --- |
| Use Case ID | URS-15 |
| Use Case name: | Administrator can add destinations. |
| Actors: | Administrator |
| Description: | Administrator can add destination for providing to the Chiang Mai Red Taxi Service Assistant. Then Passenger and Driver are able to select from the popular or frequently visited destinations list. |
| Trigger: | Administrator clicks on “Add” button that is in the destination table. |
| Pre-conditions: | 1. Administrator login to the Administration system. (URS-13)  2. Administrator browse destinations list. (URS-16) |
| Post-conditions: | 1. Administrator can Edit destinations. (URS-17)  2. Administrator can delete destinations.(URS-18) |
| Normal Flows: | 1. Administrator clicks on “Add” button that is in the destination table. 2. System shall display interface for adding new destination’s information. 3. Administrator fills in destination’s information. 4. Administrator sets the location point of destination. 5. Administrator clicks on “submit” button. 6. System shall create and save the new destination in database. 7. System shall display updated destination lists. |
| Alternative Flows: | N/A |
| Exceptions: | 3A.  1. Administrator clicks on “back” button.  2. System shall display the destination page and save the information to database. |
| Includes: |  |
| Note and Issues: |  |



*Figure19: AD- Administrator can add destinations (URS-15)*

4.16 Administrator can browse the destination (URS-16)

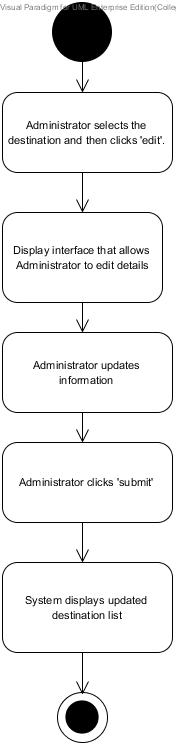
|  |  |
| --- | --- |
| Use Case ID | URS-16 |
| Use Case name: | Administrator can browse the destination. |
| Actors: | Administrator |
| Description: | Administrator can browse popular tourist destinations from the database. |
| Trigger: | Administrator clicks on “Destination”menu from the left-menu bar. |
| Pre-conditions: | 1. Administrator login to the Administration system.  2. There is destination information that has already been added to the database (URS-15). |
| Post-conditions: | 1. Administrator can add destinations (URS-15).  2. Administrator can Edit destinations (URS-17).  3. Administrator can delete destinations (URS-18). |
| Normal Flows: | 1. Administrator clicks on “Destination” menu from the left-menu bar.  2. System shall retrieve data from database  3. System shall display the list of destinations. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure20: AD- Administrator can browse the destination (URS-16)*

4.17 Administrator can edit destinations (URS-17)

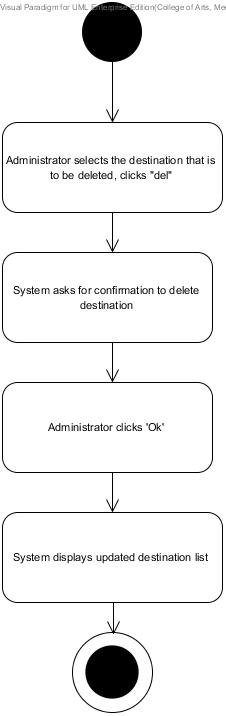
|  |  |
| --- | --- |
| Use Case ID | URS-17 |
| Use Case name: | Administrator can edit destinations. |
| Actors: | Administrator |
| Description: | Administrator can edit destination that already added in database. |
| Trigger: | Administrator clicks on “Edit” button at the last column of the destination that needed. |
| Pre-conditions: | * Administrator can browse the destination. (URS-16) * Administrator can add the destination. (URS-15) |
| Post-conditions: | * Administrator can delete destnations. (URS-18) |
| Normal Flows: | * 1. Administrator clicks on “Edit” button at the last column of the destination that needed.   2. System shall display interface for editing destination’s information.   3. Administrator updates information.   4. Administrator clicks “submit” button.   5. System shall update destination’s information to database.   6. System shall display updated destination lists. |
| Alternative Flows: | N/A |
| Exceptions: | 3A.  1. Administrator clicks on “back” button.  2. System shall display the destination page with updated information to database. |
| Includes: |  |
| Note and Issues: |  |



*Figure21: AD- Administrator can edit destinations (URS-17)*

4.18 Administrator can delete destinations (URS-18)

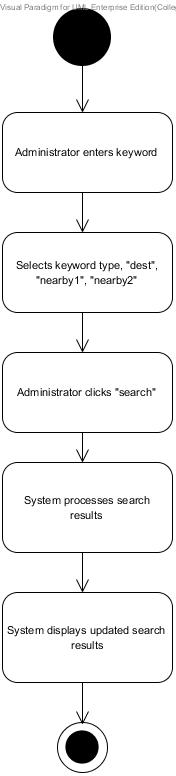
|  |  |
| --- | --- |
| Use Case ID | URS-18 |
| Use Case name: | Administrator can delete destinations. |
| Actors: | Administrator |
| Description: | Administrator can delete destination that is redundant from the database, the Passenger and Driver will not be able to see it in the destination list. |
| Trigger: | After Administrator clicks on check box of each destination row that needs to be deleted, Administrator clicks on the “Delete” button. |
| Pre-conditions: | * Administrator can add the destination. (URS-15) * Administrator can browse the destination. (URS-16) |
| Post-conditions: | N/A |
| Normal Flows: | 1. Administrator clicks on check box of each destination row that needs to be deleted.  2. Administrator clicks on the “Delete” button.  2. System shall display the pop up message “Do you want to DELETE selected item? Delete or Cancel”.  3. Administrator clicks on “Delete” button.  4. System shall remove that destination from database.  5. System shall display updated destination lists. |
| Alternative Flows: | N/A |
| Exceptions: | 2A.  1. Administrator clicks on “Cancel” button.  2. System shall not remove that destination from database.  3. Resume step 5. |
| Includes: |  |
| Note and Issues: |  |



*Figure22: AD- Administrator can delete destinations (URS-18)*

4.19 Administrator can search destinations (URS-19)

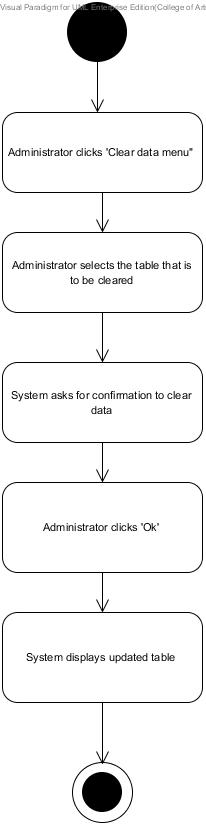
|  |  |
| --- | --- |
| Use Case ID | URS-19 |
| Use Case name: | Administrator can search destinations. |
| Actors: | Administrator |
| Description: | Administrator can search destination from destination lists that already exist in database. |
| Trigger: | Administrator clicks on “Search” button. |
| Pre-conditions: | * Administrator can browse the destination. (URS-16) * If there is already existing destination information in database, Administrator can add new destinations. (URS-15) |
| Post-conditions: | - Administrator can add destinations. (URS-15)  - Administrator can edit destinations. (URS-17)  - Administrator can delete destinations. (URS-18) |
| Normal Flows: | 1. Administrator enters a keyword. 2. Administrator can choose that search from “Destinations” “Nearby1” or “Nearby2” . 3. Administrator clicks on “Search” button. 4. System shall process searching that keyword with database. 5. System shall display searching destinations result in a list. |
| Alternative Flows: | N/A |
| Exceptions: | 3A.  1. If there is no matching keyword related to the destination.  2. System shall display “not found” if destination not stored in the database. |
| Includes: |  |
| Note and Issues: |  |



*Figure23: AD- Administrator can search destinations (URS-19)*

4.20 Administrator can clear data (URS-20)

|  |  |
| --- | --- |
| Use Case ID | URS-20 |
| Use Case name: | Administrator can clear data. |
| Actors: | Administrator |
| Description: | Administrator clears data in the database. Administrator can select the table that is to be deleted. Including:  -tdestination -tdriver -tdriverposition -tpassenger -tpassengerposition -trequest -trequestmsg |
| Trigger: | Administrator clicks on “clear data” button at the menu bar in home page. |
| Pre-conditions: | * Administrator can login to the Administration system (URS-013) |
| Post-conditions: | N/A |
| Normal Flows: | 1. Administrator clicks on “clear data” button at the menu bar in home page. 2. System shall display the table that shows the name of the table that Administrator can delete. 3. Administrator can select the table that is to be deleted, then click “del”. 4. System shall ask for confirmation to clear data. 5. Administrator clicks “OK”. 6. System shall display updated table. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |



*Figure24: AD- Administrator can clear data (URS-20)*