**Chapter 3**

**Software Requirement Specification**

**Chiang Mai Red Taxi**

**Service Assistant**

**Project Proposal**

**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**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version**  **Number** | **Record Date** | **Prepared/Modified By** | **Reviewed**  **By** | **Change Details** |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

**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 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 Web 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 comprising of the main features and sub-features. The main features consist of:

• A registration and log in function for both drivers and passengers.  
• Passengers are able to set their destination and the number of passengers travelling.

• Passengers are able to send a request to a driver.

• Passengers are able to receive confirmation if the driver accepts the request.

• Passengers are able to view the booking information of red taxi.

• Passengers are able to view the current location of the red taxi that match their conditions.

• Drivers are able to update the availability of seats.

• Drivers are able to receive requests from passengers.

• Drivers are able to accept or decline the request.

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

• Passengers are able change the scope of searching for red taxis.

• Passengers are able to cancel the request.

• Passengers are able to receive notifications when the red taxi arrives.

• Drivers are able to enable or disable the service (e.g. off duty).

• Drivers are 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 must preset a general direction of driving using their mobile application before leaving.
* Only passengers are able to send a request to drivers (one driver per request).
* Drivers are unable to locate passengers unless there is a request from them.
* Services limited to the passengers and drivers in Chiang Mai only
* Services are limited to the registered taxis and passengers only.  
  (Registration required before using the app)

This application aims 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 an 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] |

**1.5.1 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 would be able to 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 drivers of the red taxi receive 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.

**2.3 Features**

Main features

• Both passengers and drivers can register themselves to the system.

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

• Passengers can send a request to a driver.

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

• Passengers can see booking information of red taxi.

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

• Drivers can update the number of available seats.

• Drivers can tell where the taxi is heading.

• Drivers can get the request from passenger.

• Drivers can either accept or decline the request.

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 taxis1.

• Drivers can receive the cancellation request.

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

**2.4 Functional & User Requirements**

**UC-01 Passengers can search for the red taxi which matches their conditions.**

**SRS-01** System provides the interface to the passenger to input the conditions and to search for a red taxi.

**SRS-02** System checks if passenger inputs all conditions. If not completed, system provides an error message, “Please input all conditions”.

**SRS-03** System processes the search conditions – (eg: number of seats and destination headed to) together with the taxi driving information which is stored in the database.

**SRS-04** System provides the results from the search that matches the conditions that display a list of up to 10 search results.

**UC-02** **Passengers can send a request to a driver.**

**SRS-05** System sends a request containing the conditions and taxi information to the server.

**SRS-06**System processes sending request form Server to Driver that Passenger selects.

**SRS-07** System displays “Message has been sent” if the request for the taxi has been processed.

**SRS-08**System displays the request information on a page after the passenger sent request.

**SRS-09**System displays “Message sending fail.” If the message cannot be sent to the Driver.

**UC-03** **Passengers can view booking information of red taxi.**

**SRS-10** System retrieves the booking information and displays it to passenger.

**UC-04** **Passengers can view the “view information” of red taxi that matches their conditions.**

**SRS-11** System displays the information of the red taxi that passenger selects.

**UC-05** **Driver can log out from the system.**

**SRS-12** System change status of the driver to offline in database.

**SRS-13** System displays login page after the driver logged out**.**

**UC-06 Passenger can log out from the system.**

**SRS-14** System change status of the driver to offline in database.

**SRS-15** System displays login page after passenger logged out.

**UC-07 Driver can update driving information.**

**SRS-16** System display the driving information, the driver can update driving information (e.g.: number of available seats, destination)

**SRS-17** System updates the driving information and displays “Update Successful” message.

**UC-08 Driver can view the requests from passenger.**

**SRS-18** System displays all requests sent by the passenger.

**UC-09 Drivers can respond to passenger’s request.**

**SRS-19** System displays the information of the passenger’s request.

**SRS-20** System updates the status of the passenger’s request to “Accepted” if the Driver clicks “accept”.

**SRS-21**System will remove the passenger’s request form the request page if the Driver denies the request.

**SRS-22**System will remove the passenger’s request form the request page if the Driver did not respond to the new request within 60 seconds.

**UC-10 The driver can register to the system**

**SRS-23** System displays the registration form.

**SRS-24** System will notify ‘Wrong format!’ if the driver inputs the wrong format.

**SRS-25** System will notify, ‘This username already exists.’ if the driver inputs an existing username.

**SRS-26** System will notify, ‘This license ID already exists.’ if the driver inputs an existing license ID.

**SRS-27** System creates the new user in the database and displays the   
“registration successful” message.

**SRS-28** System displays the log in page after driver registration.

**UC-11 The driver can log in to the system.**

**SRS-29** System displays the login page to the driver.

**SRS-30** System will notify, ‘This username or password is wrong.’ if the driver inputs the username or password incorrectly.

**SRS-31**System updates the driver status to online status in the database, the system displays the login successful message.

**UC-12 Passenger can register to the system**

**SRS-32**The system displays the registration form to the passenger**.**

**SRS-33** The system will notify ‘Wrong format!’ if the passenger inputs wrong format.

**SRS-34** The system will notify, ‘This username already exists.’ if the passenger inputs an existing username.

**UC-13 The Passenger can log in to the system.**

**SRS-35** System displays the login page to the passenger

**SRS-36** System will notify, ‘This username or password is wrong.’ if the passenger inputs the username or password incorrectly.

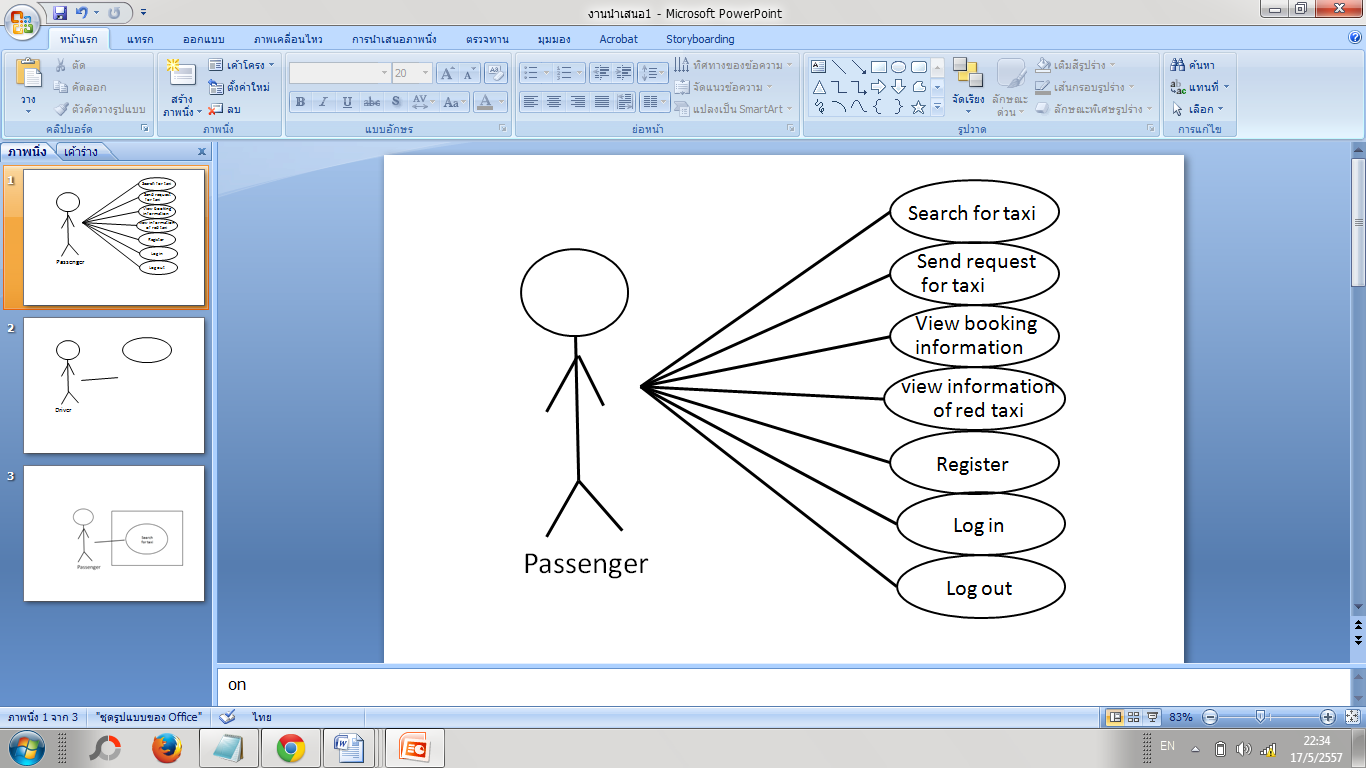
**SRS-37** System updates the passenger status to online status in the database.

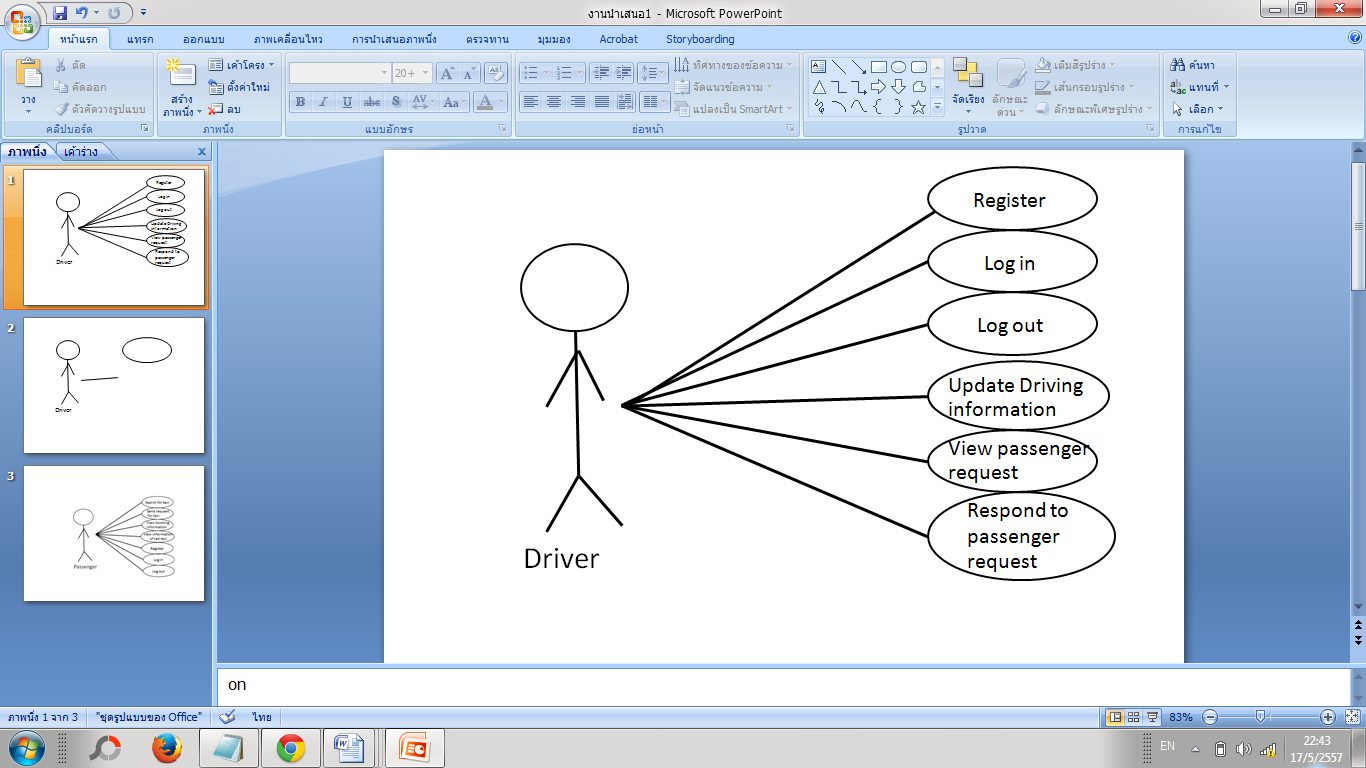
**SRS-38** System displays the login successful message to passenger.

**Chapter Three | Specific Requirement**

**3.1 Use Case Diagram**

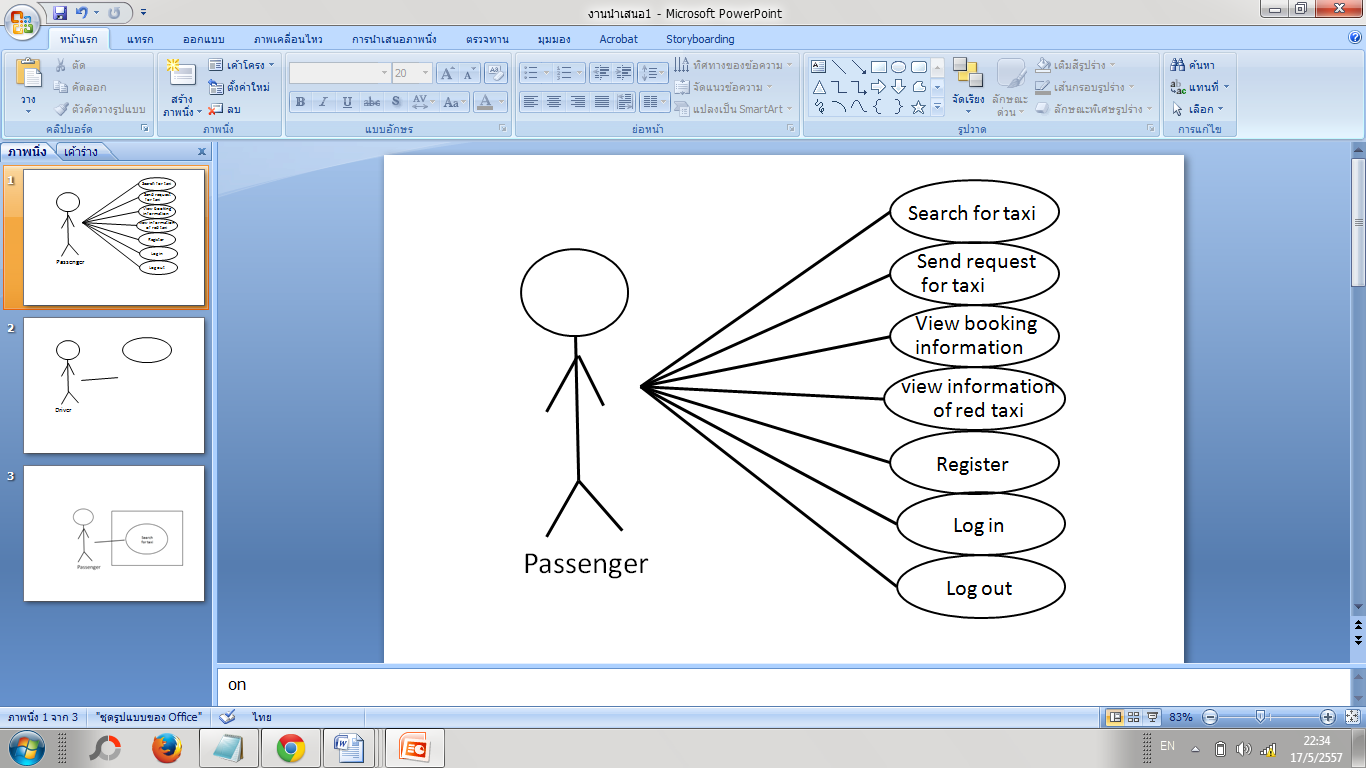
**3.1.1 The Overview Use Case Diagram of the System**

****

******

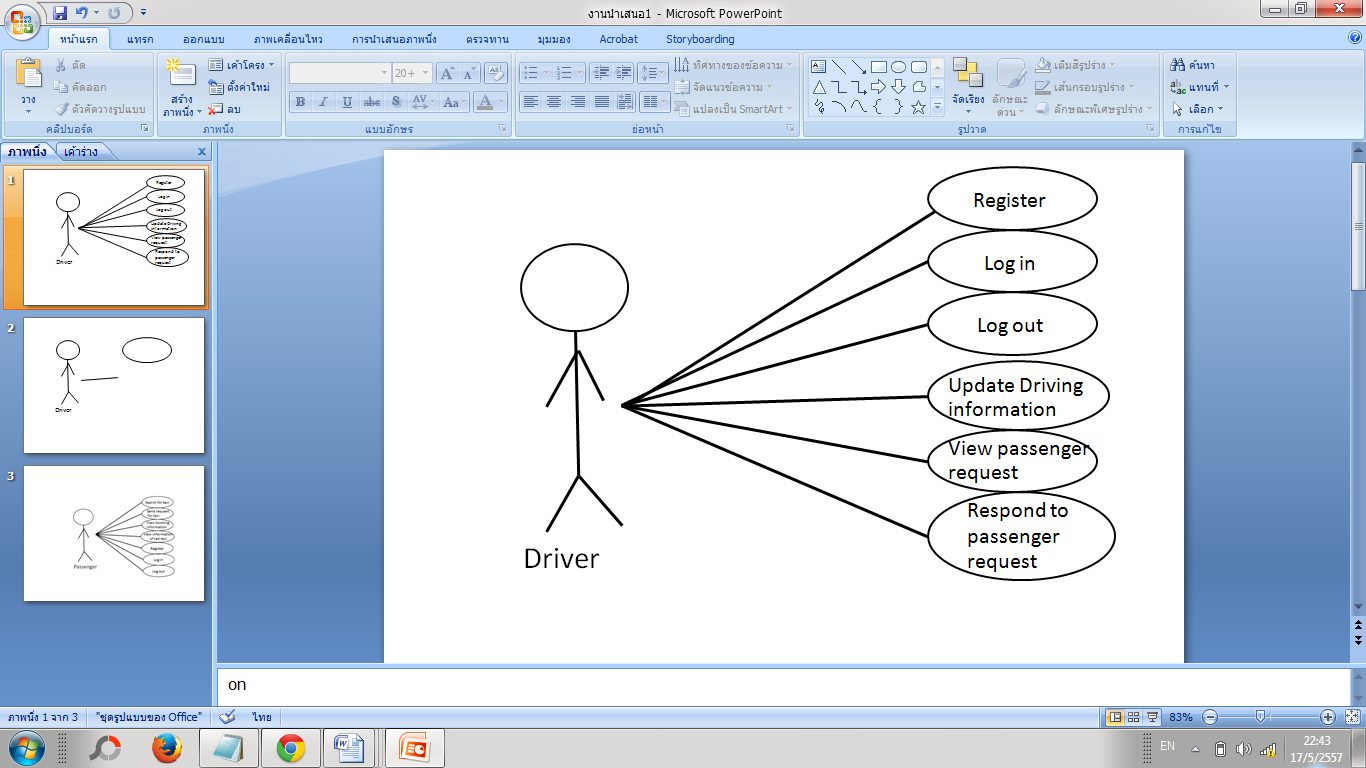
*Figure 1: The Overview Use Case Diagram of the System.*

**3.1.2 The Passenger Use Case Diagram**

****

*Figure 2: The PassengerUse Case Diagram.*

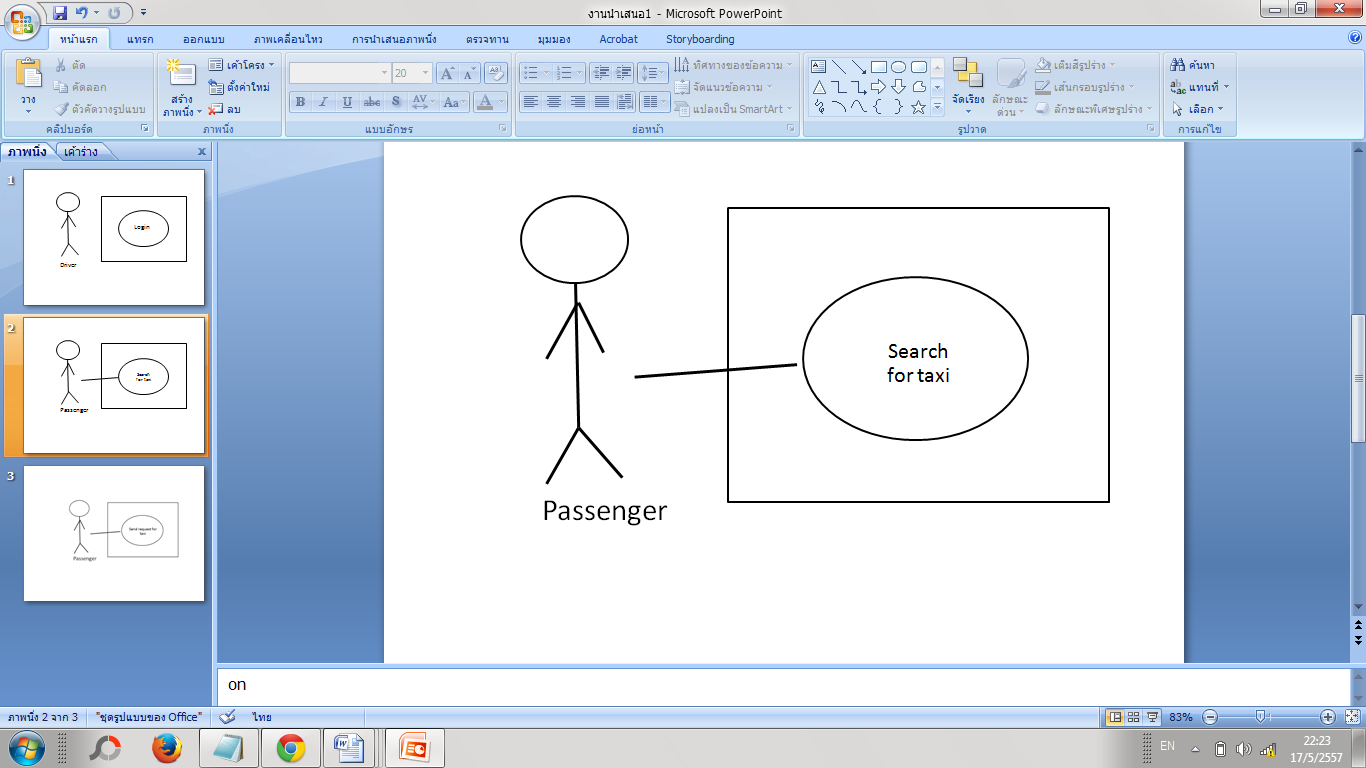
**3.1.2 The Driver Use Case Diagram**

****

*Figure 3: The Driver Use Case Diagram.*

**Chapter Four | Use case description**

4.1 Search for taxi Use Case (UC-01)



*Figure 4: Passengers can search for the red taxi which matches their conditions (UC-01)*

|  |  |
| --- | --- |
| Use Case ID | UC-01 |
| Use Case name: | Passengers can search for the red taxi which matches their conditions. |
| Actors: | Passenger |
| Description: | Passenger able to input their condition ,then search for the red taxi which is nearest to the user and match their condition. |
| Trigger: | Passenger selects the button in the menu bar called “Search for Taxi”. |
| Pre-conditions: | * Passenger logs into the Chiang Mai red taxi service assistant as a “passenger”. (UC-14) * System must have at least one driving information of the red taxi. |
| Post- Condition | * Passenger can view driving information of the taxi. (UC-04) |
| Normal Flows: | 1. Passenger selects the button in the menu bar called “Search for Taxi”. 2. System provides the interface to the passenger to input the conditions and to search for a red taxi. 3. Passenger inputs conditions – Eg: number of seats and destination headed to. Then passenger can select the ‘search’ button to proceed with searching for a red taxi. 4. System checks if passenger inputs all conditions. If not completed, system provides an error message, “Please input all conditions”. 5. System processes the search conditions – (eg: number of seats and destination headed to) together with the taxi driving information which is stored in the database. 6. System provides the results from the search that matches the conditions that display a list of up to 10 search results. |
| Alternative Flows: | 3A. Passenger can edit the scope of search. Default scope is 1km.  4A. If passenger did not input all the conditions, the system will display an error message, “Please input all conditions”. Passenger can edit the conditions and proceed on to step 5. |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: | If the passenger have not input the conditions yet? Should we provide any error? |

**UC-01 Passengers can search for a red taxi which matches their conditions.**

**Introduction:** This use case the passenger can be able to input their conditions, then search for the red taxi which is nearest to the user and match their condition.

**Input:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Example** | **Note** |
| Seat | 1 | List of integers1-11 |
| Destination | Doisuthep | Select form the destinationslist,which are provided in English.(only alphabet) |
| Scope | 1 | List of intergers 1-10 (Default is 1). |

**Action:**

**Output: If successful,** the system will provide the result page of the red taxi that match their conditions up to 10 taxis.

**System Requirement Specification**

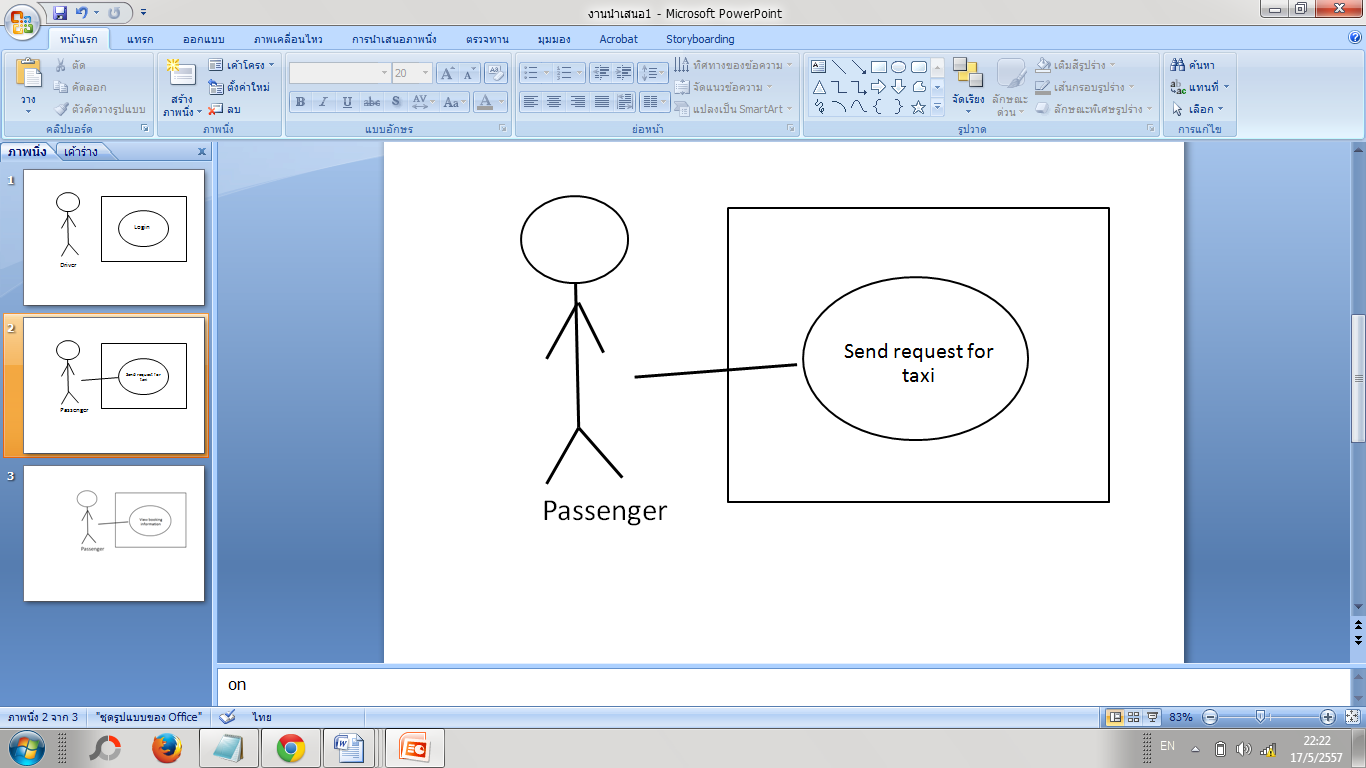
**SRS-01** System provides the interface to the passenger to input the conditions and to search for a red taxi.

**SRS-02** System checks if passenger inputs all conditions. If not completed, system provides an error message, “Please input all conditions”.

**SRS-03** System processes the search conditions – (eg: number of seats and destination headed to) together with the taxi driving information which is stored in the database.

**SRS-04** System provides the results from the search that matches the conditions that display a list of up to 10 search results.

4.2 Send request for taxi Use Case (UC-02)

**

*Figure 5: Passengers can send a request to a driver. (UC-02)*

|  |  |
| --- | --- |
| Use Case ID | UC-02 |
| Use Case name: | Passengers can send a request to a driver. |
| Actors: | Passenger |
| Stakeholders and Interests: |  |
| 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. (UC-01) |
| Post-conditions: | * Driver can view the request that the passenger sent. (UC-08) |
| Normal Flows: | 1. Passenger clicks “send request” button in the driving information page. 2. System sends a request containing the conditions and taxi information to the server. 3. System processes sending request form Server to Driver that Passenger selects. 4. System displays message “Message has been sent” if the request for the the taxi has been processed. 5. System displays the request information on a page after the passenger sent request. |
| Alternative Flows: | 4A. If the message cannot be sent to the Driver, the system will display “Messsage sending fail.” |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: | \*Do we have to mention about the response from the driver???? |

**UC-02** **Passengers can send a request to a driver.**

**Introduction:**In this use case the passenger can send a request to the driver after searching for the taxi process

**Input:** None

**Action:**

**Output:** If successful, the system will provide the interface and display message that the request has been sent.

**System Requirement Specification**

**SRS-05** System sends a request containing the conditions and taxi information to the server.

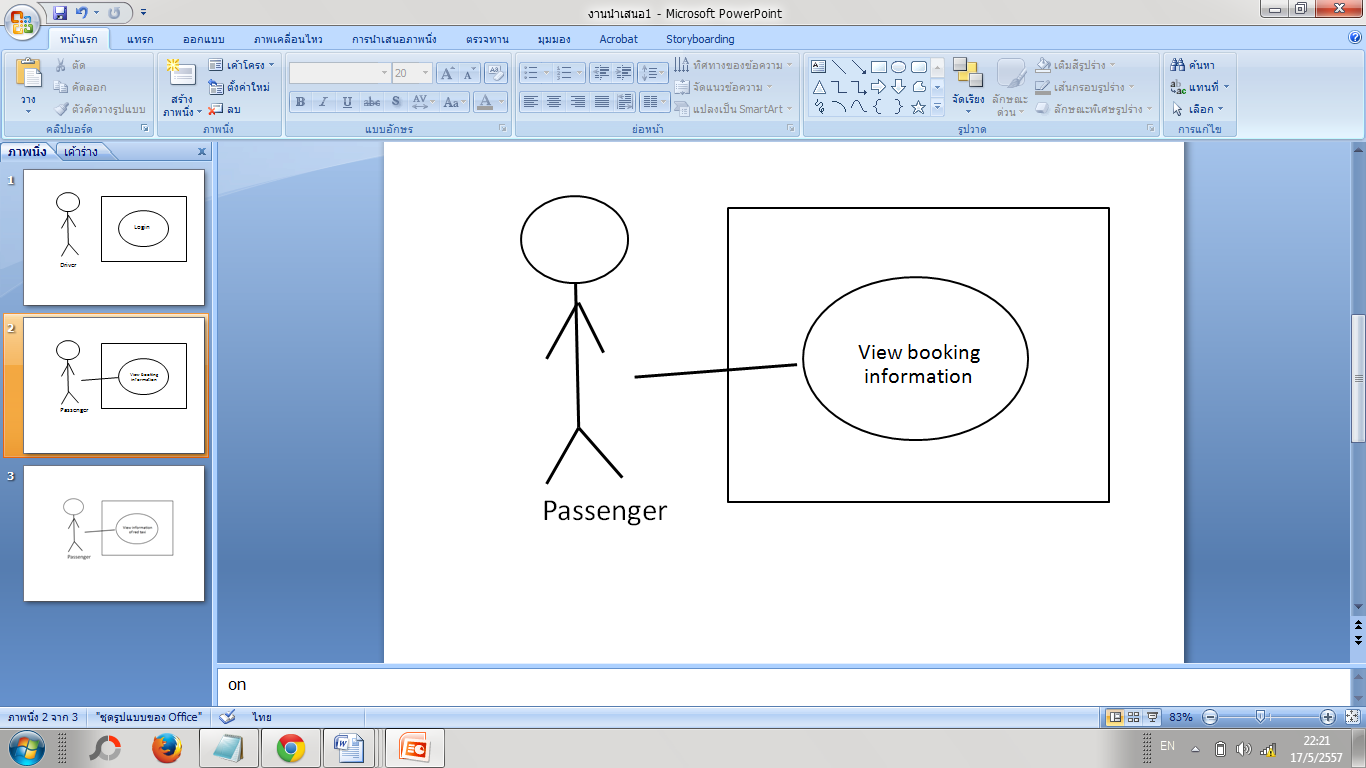
**SRS-06** System processes sending request form Server to Driver that Passenger selects.

**SRS-07** System displays message “Message has been sent” if the request for the taxi has been processed.

**SRS-08** System displays the request information on a page after the passenger sent request.

**SRS-09** System displays “Message sending fail.” if the message cannot be sent to the driver.

4.3 View booking information Use Case (UC-03)

****

*Figure 6: Passengers can view booking information of red taxi. (UC-03)*

|  |  |
| --- | --- |
| Use Case ID | UC-03 |
| Use Case name: | Passengers can view booking information of red taxi. |
| Actors: | Passenger |
| Description: | Passenger can view the request information that the driver already confirmed via the “booking” tab in the menu bar at the button of the screen. |
| Trigger: | Passenger click on the “booking” tab in the menu. |
| Pre-conditions: | * Passenger has already sent a request to a driver.  (UC-02) |
| Post-conditions: | N/A |
| Normal Flows: | 1. Passenger click on the “booking” tab in the menu bar at the button of the screen.  2.System retrieves the booking information from database and displays it to passenger. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |

**UC-03** **Passengers can view booking information of red taxi.**

In this use case the passenger can view the booking information that the driver already confirm the request via the booking tap in the menu bar at the button of the screen.

**Input:** None

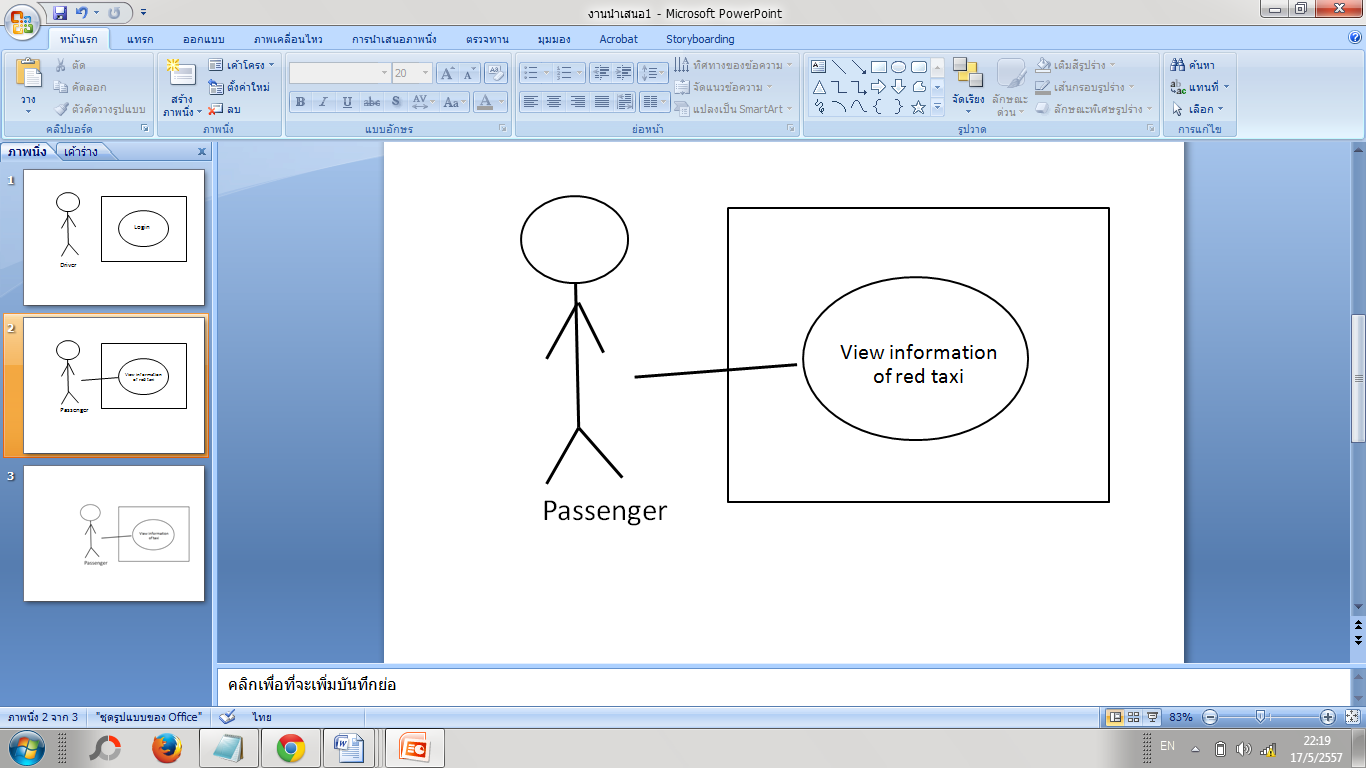
**Action:**

**Output:** If successful, the system will provide interface that display the booking information of passenger.

**System Requirement Specification**

**SRS-10** System retrieves the booking information and displays it to passenger.

4.4 View Information of red taxi Use Case (UC-04)

****

*Figure 7: Passengers can view the View Information of red taxi that matches their conditions. (URS-04)*

|  |  |
| --- | --- |
| Use Case ID | UC-04 |
| Use Case name: | Passengers can view the View Information of red taxi that matches their conditions |
| Actors: | Passenger |
| Description: | Passenger can view the information of red taxi that matches their condition after the search process. The information includes:  -the driver’s picture -the driver licence ID  -name of driver  -number of available seats -the map that allows passenger to trace the current location of the red taxi  -places that where they are heading to. |
| Trigger: | Passenger clicks on one of the results from the red taxi list. |
| Pre-conditions: | Passengers has found a red taxi which matches their conditions. (UC-01) |
| Post-conditions: | Passenger can send a request to the driver after searching for the taxi. (UC-02) |
| Normal Flows: | 1.Passenger click on one of the list of red taxi searching result.  2.The system displays the information of the red taxi that passenger selects. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |

**UC-04** **Passengers can view booking information of red taxi.**

In this use case the passenger can view the information of red taxi that matching their condition. The information is including the driver picture,the driver lisence ID, Name of driver,The number of availale seats,the map that passenger can trace the current location of that red taxi ,and the places that where they are heading to.

**Input:** None

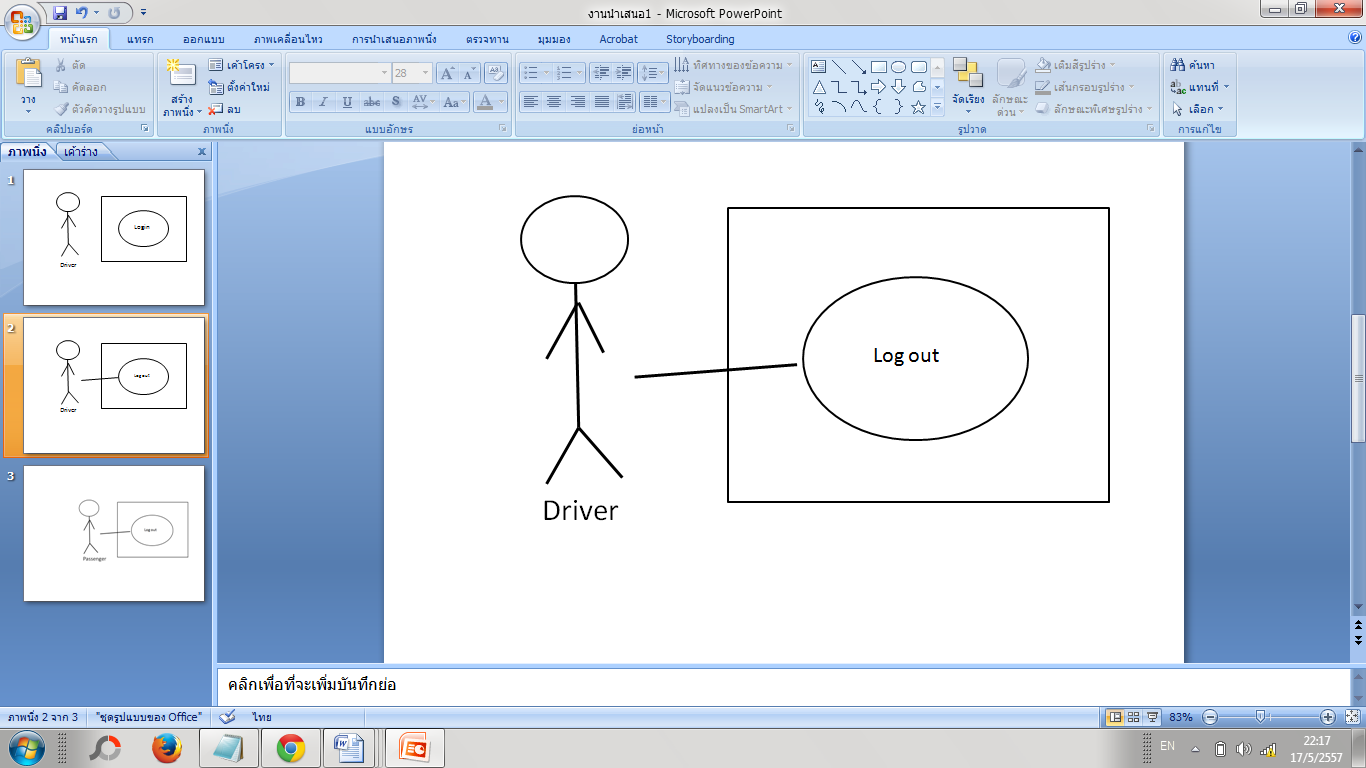
**Action:**

**Output:** If successful, the system will provide the interface that display the which are including the driver picture,the driver lisence ID, Name of driver,The number of availale seats,the map that passenger can trace the current location of that red taxi ,and the places that where they are heading to.

**System Requirement Specification**

**SRS-11** System displays the information of the red taxi that passenger selects.

4.5 Driver Logout Use Case (UC-05)

**

*Figure 8: Driver can log out from the system. (URS-05)*

|  |  |
| --- | --- |
| Use Case ID | UC-05 |
| 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”. (UC-11) |
| Post-conditions: | Driver can log into the system (UC-11) |
| Normal Flows: | 1. Driver click logout button. 2. The system change status of the driver to offline in database. 3. The system displays login page after Driver loged out. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |

**UC-05 Driver can log out from the system.**

**In this use case the driver can logout from the system if the driver ends of using service.**

**Input:**

**Action:**

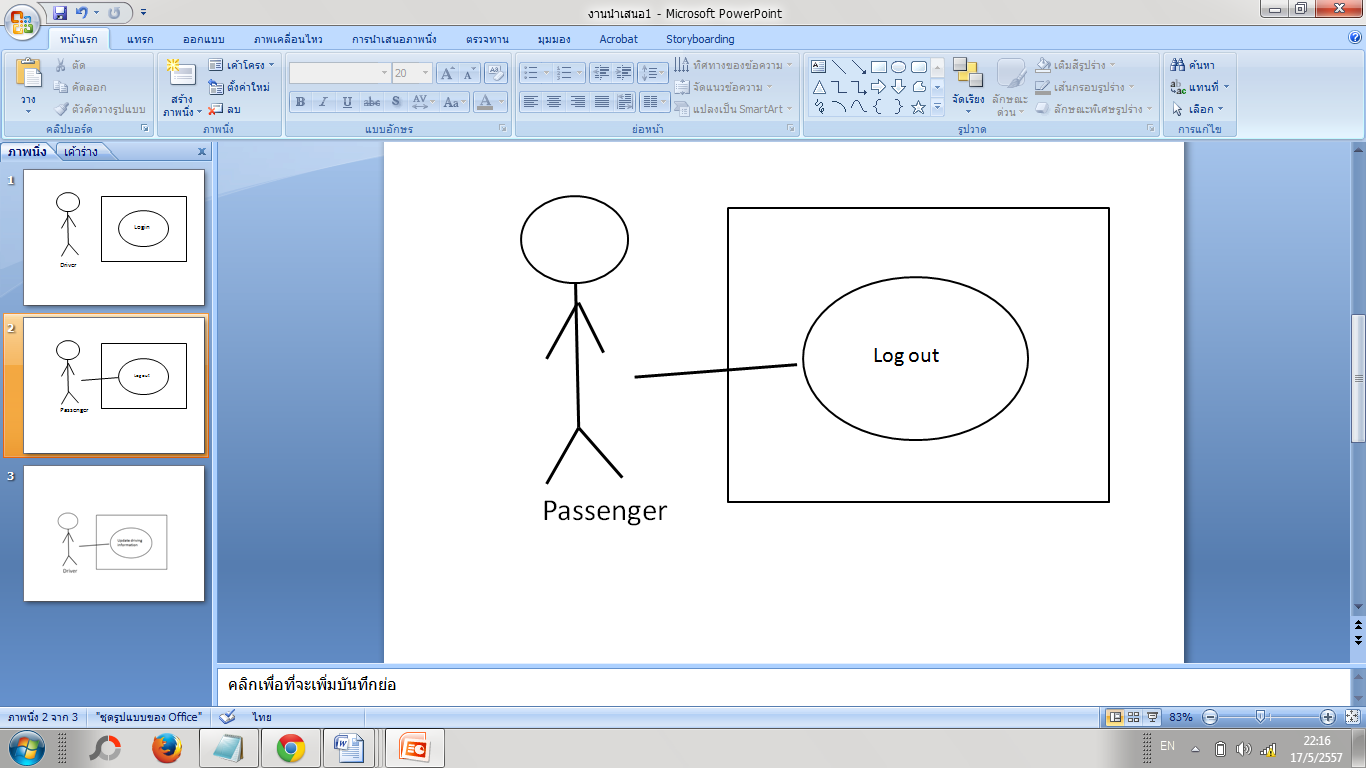
**Output:** If successful, the system will update the driver status to offline in the database and display the login page.

**System Requirement Specification**

**SRS-12** System change status of the driver to offline in database.

**SRS-13** System displays login page after Driver logged out**.**

4.6 Passenger Logout Use Case (UC-06)

**

*Figure 9: Passenger can log out from the system. (URS-06)*

|  |  |
| --- | --- |
| Use Case ID | UC-06 |
| Use Case name: | Passenger can log out from the system. |
| Actors: | Passenger |
| Description: | Passenger can logout from the system when the driver stops using the service. |
| Trigger: | Passenger clicks on the “logout” button in the settings page. |
| Pre-conditions: | The Passenger logs into the Chiang Mai Red Taxi Service Assistant as a “Passenger”. (UC-14) |
| Post-conditions: | Passenger can log into the system (UC-14) |
| Normal Flows: | 1. Passenger clicks on the “logout” button in the settings page. 2. The system change status of the driver to offline in database. 3. The system displays login page after passenger logged out. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: |  |
| Note and Issues: |  |

**UC-06 Passenger can log out from the system.**

**In this use case the passenger can logout from the system**

**Input:**

**Action:**

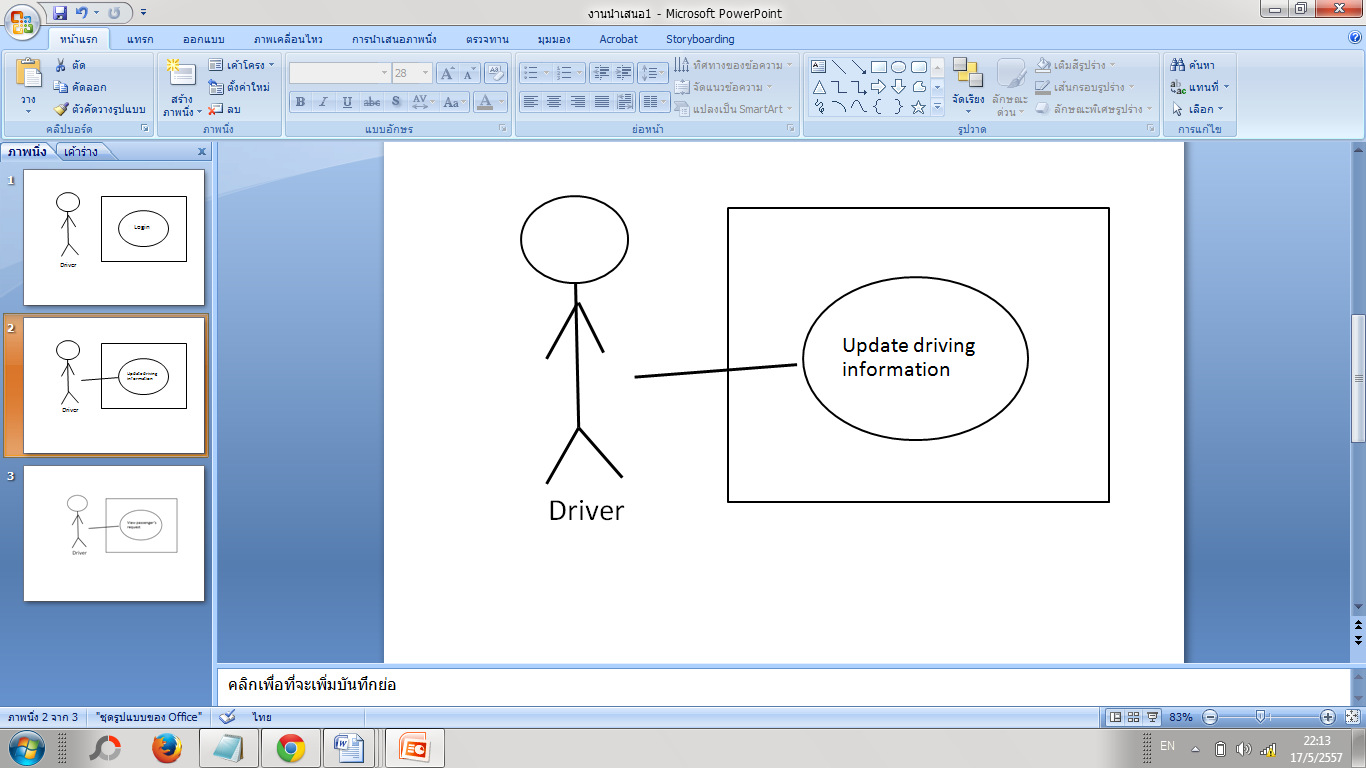
**Output:** If successful, the system will update the driver status to offline in the database and display the login page.

**System Requirement Specification**

**SRS-14** System change status of the driver to offline in database.

**SRS-15** System displays login page after passenger logged out.

4.8 Update driving information Use Case (UC-07)

**

*Figure 10: Driver can update driving information (URS-7)*

|  |  |
| --- | --- |
| Use Case ID | UC-07 |
| 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. And the passenger can also see these information. |
| Trigger: | Driver click on the “driving information menu” at the bottom of the screen . |
| Pre-conditions: | * The driver logs into the Chiang Mai Red Taxi Service Assistant as a “driver”.   (UC-11) |
| Post-conditions: | Passengers can view the “view information” of red taxi that matches their conditions. (UC-04) |
| Normal Flows: | 1. Driver click on the driving information menu. 2. The system display the driving information, the driver can update driving information(e.g.: number of available seats, destination) 3. The driver click on “save” button 4. The system updates the driving information and displays “Update Successful” message. |
| Alternative Flows: | N/A |
| Exceptions: |  |
| Includes: |  |
| Note and Issues: |  |

**UC-07 Driver can update driving information**

In this use case the driver can update driving information for the system can process macthing taxi and passenger. And the passenger can also see these information.

**Input:**

**Action:**

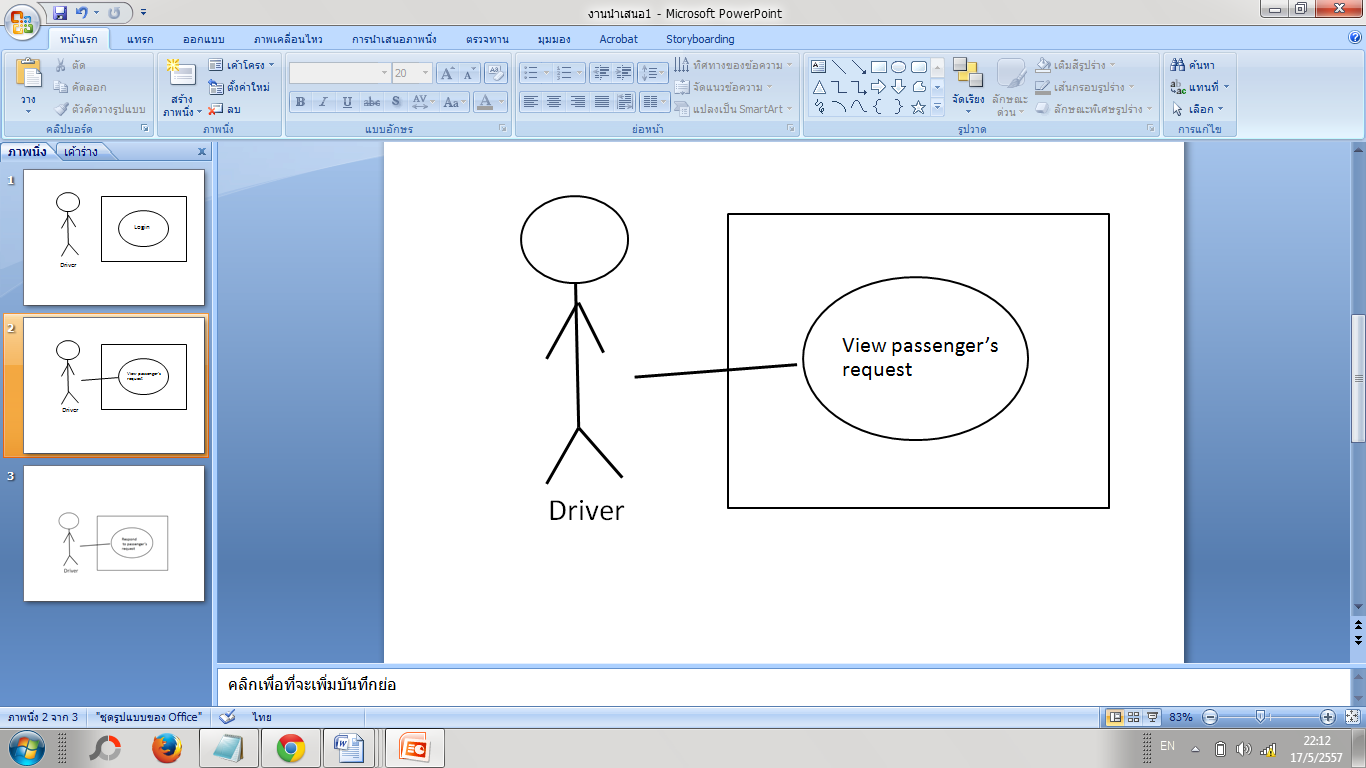
**Output:** If successful, the system will update the driving information and display the successful message.

**System Requirement Specification**

**SRS-16** System display the driving information, the driver can update driving information (e.g;: number of available seats, destination)

**SRS-17** System updates the driving information and displays “Update Successful” message.

4.9 View passenger’s request Use Case (UC-08)

**

*Figure 11: Drivers can view the requests from passenger. (URS-8)*

|  |  |
| --- | --- |
| Use Case ID | UC-08 |
| Use Case name: | Driver can view the requests from passenger. |
| Actors: | Driver |
| Description: | Driver can view the request that has been sent from the passenger. |
| Trigger: | Driver clicks on the message/request menu. |
| Pre-conditions: | * Driver logs into the Chiang Mai Red Taxi Service Assistant as a “driver”. (UC-11) |
| Post-conditions: | * Driver can respond to the request (UC-09) |
| Normal Flows: | 1. Driver clicks on the message/request menu. 2. The system displays all the requests sent by the passengers. |
| Alternative Flows: | N/A |
| Exceptions: |  |
| Includes: |  |
| Note and Issues: |  |

**UC-08 Drivers can view the requests from passenger.**

In this use case the driver can view the request that has been sent from the passenger.

**Input:**

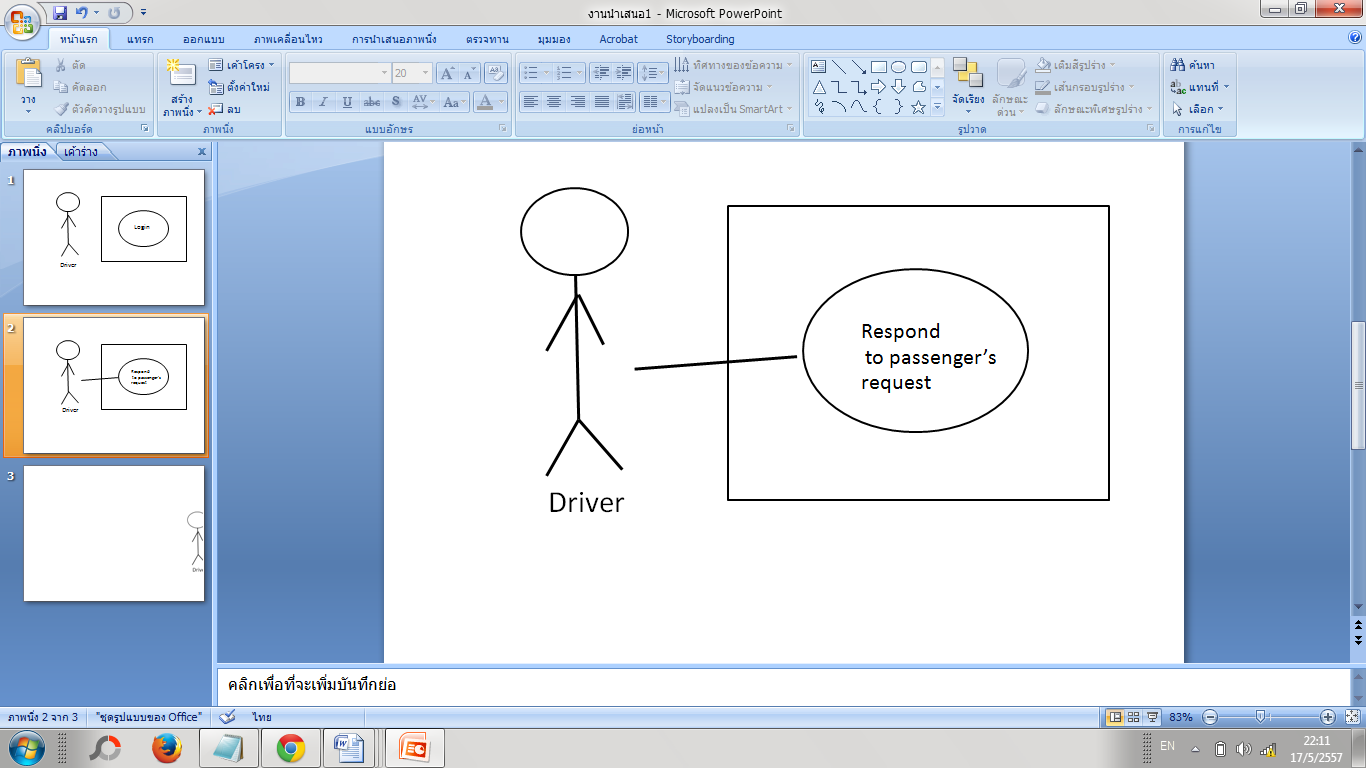
**Action:**

**Output:** If successful, the system will display all the request that has been sent from the passenger.

**System Requirement Specification**

**SRS-18** System displays all requests sent by the passenger.

4.10 Respond to passenger’s request Use Case (UC-09)

**

*Figure 12: Drivers can respond to passenger’s request (URS-09)*

|  |  |
| --- | --- |
| Use Case ID | UC-09 |
| Use Case name: | Drivers 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 click on each request in the “request page”. |
| Pre-conditions: | * Passenger’s request has already been sent to the driver. (UC-02) * The request status is listed as “new”. |
| Post-conditions: | * Passenger receives Driver’s response.   (UC-03) |
| Normal Flows: | 1. Driver click on each request in the “request page”. 2. The system displays the information of the request. 3. The Driver can choose to accept or deny the request by clicking the button. 4. The system updates the status of the request to “Accepted” if the Driver clicks “accept”. |
| Alternative Flows: | 4A. If the Driver denies the request, the request will disappear from the “request page”. |
| Exceptions: | If the Driver did not respond to the new request within 60 seconds, the request will disappear. |
| Includes: |  |
| Note and Issues: |  |

**UC-09 Respond to passenger’s request**

In this use case the driver can respond to the request that has been sent from the passenger.

**Input:**

**Action:**

**Output:** If successful, the system will forward the response from the Driver to the Passenger. If the Driver accepts the request, the status of the request is defined as ‘accepted’. If the Driver denies the request, the request disappears.

**System Requirement Specification**

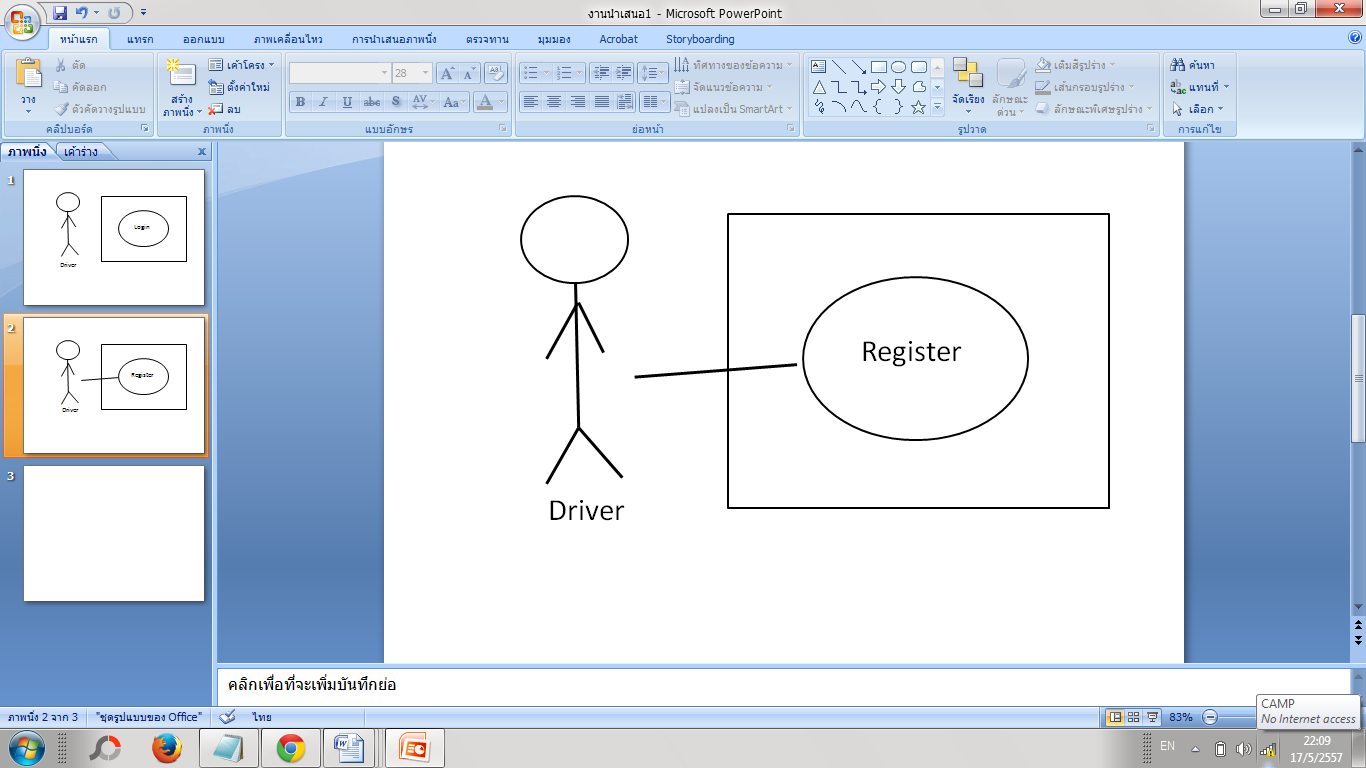
**SRS-19** System displays the information of the request.

**SRS-20** System updates the status of the request to “Accepted” if the Driver clicks “accept”.

**SRS-21**System will remove the request form the request page if the Driver denies the request.

**SRS-22**System will remove the request form the request page if the Driver did not respond to the new request within 60 seconds.

4.13 Driver Registration Use Case (UC-10)

**

*Figure 13: The driver can register to the system (URS-10)*

|  |  |
| --- | --- |
| Use Case ID | UC-10 |
| Use Case name: | The driver can register to the system |
| Actors: | Driver |
| Description: | Driver can register themself as a ‘Red taxi driver’. |
| Trigger: | Driver click on the register button when starting the application. |
| Pre-conditions: | N/A |
| Post-conditions: | The driver can log into the system. (UC-11) |
| Normal Flows: | 1. Driver click on the register button when starting the application. 2. The system displays the registration form. 3. Driver input their particulars. 4. Registration validation (to ensure no duplication). 5. The system creates the new user in the database and displays the” registration successful” message. 6. The system displays the log in page. |
| Alternative Flows: | N/A |
| Exceptions: | 4A If driver inputs wrong format, the system will notify ‘Wrong format!’ The driver can redo the 3rd step again.  4B If driver inputs an existing username, the system will notify, ‘This username already exists.’ The driver can redo the 3rd step again.  4C If the driver inputs an existing license ID, the system will notify, ‘This license ID already exists.’The driver can redo the 3rd step again. |
| Includes: |  |
| Note and Issues: |  |

**UC-10 The driver can register to the system.**

**Introduction:** In this use case the driver can register themself as a ‘Red taxi driver’.

**Input:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Example** | **Note** |
| Username | Somchai99 | Alphabet A-Z or Integer 0-9 The length is not more than 10 characters. |
| Password | Love66 | Alphabet A-Z or Integer 0-9 The length is 4-8 characters. |
| Name | Somchai | Alphabet A-Z. The length is 20 characters. |
| Last Name | Rakrian | Alphabet A-Z. The length is 20 characters. |
| License ID | กข5555 | Thai Alphabet. ก-ฮ and Integer 0-9. The length is 6 characters. |
| DriverPicture | aa.jpg | The image format should be in .jpg. |

**Action:**

**Output:** If successful, the system will display the messege that the registration is successful and display the log in page.

**System Requirement Specification**

**SRS-23** System displays the registration form.

**SRS-24** System will notify ‘Wrong format!’ if the driver inputs the wrong format.

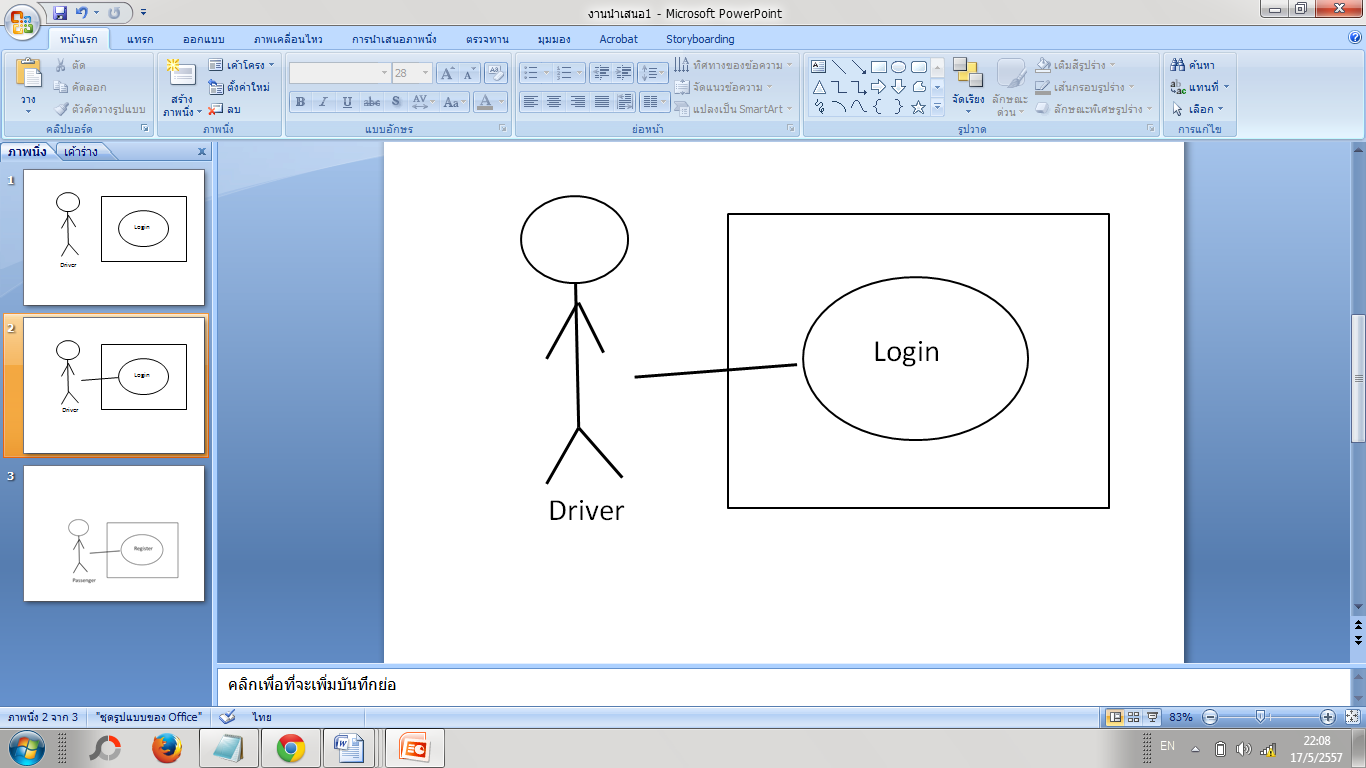
**SRS-25** System will notify, ‘This username already exists.’ if the driver inputs an existing username.

**SRS-26** System will notify, ‘This license ID already exists.’ if the driver inputs an existing license ID.

**SRS-27** System creates the new user in the database and displays the   
“registration successful” message.

**SRS-28** System displays the log in page after driver registration.

4.14 Driver Log In Use Case (UC-11)

**

*Figure 14: The driver can log in to the system (URS-11)*

|  |  |
| --- | --- |
| Use Case ID | UC-11 |
| Use Case name: | The 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: | The driver click on ‘Driver log in’ button in the login page. |
| Pre-conditions: | Driver is registered in the system. (UC-10) |
| Post-conditions: | * Driver can log out. (UC-05) * Driver can access the application |
| Normal Flows: | 1. The driver click on ‘Driver log in’ button in the login page. 2. The system displays the login page. 3. Driver input the username and password. 4. User validation (to ensure no duplication). 5. The system updates the driver status to online status in the database, the system displays the login successful message. |
| Alternative Flows: | N/A |
| Exceptions: | 4A If the driver inputs the username or password incorrectly, the system will notify, ‘This username or password is wrong.’ The driver re-enters the username and password again and can proceed on to step 5. |
| Includes: |  |
| Note and Issues: |  |

**UC-11 The driver can log in to the system.**

**Introduction:** In this use case the driver can log in to the system for accessing the system as a ‘red taxi driver’.

**Input:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Example** | **Note** |
| Username | Somchai99 | Alphabet A-Z or Integer 0-9 The length is not more than 10 characters. |
| Password | Love66 | Alphabet A-Z or Integer 0-9 The length is 4-8 characters. |

**Action:**

**Output:** If successful, the system will display the message that the log in is successful.

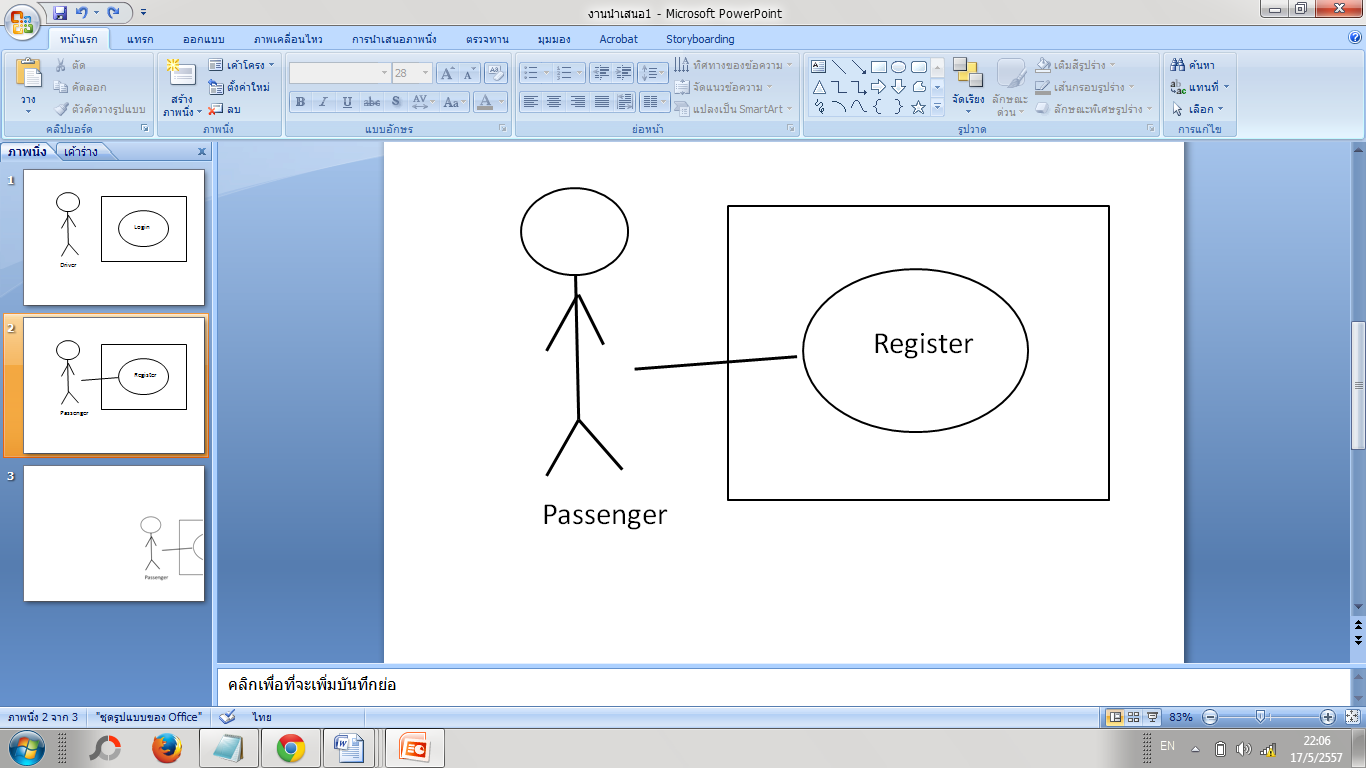
**System Requirement Specification**

**SRS-29** System displays the login page to driver.

**SRS-30** System will notify, ‘This username or password is wrong.’If the driver inputs the username or password wrongly.

**SRS-31**System updates the driver status to online status in the database, the system displays the login successful message.

4.13 Passenger Registration Use Case (UC-12)

**

*Figure 15: The passenger can register to the system (URS-12)*

|  |  |
| --- | --- |
| Use Case ID | UC-12 |
| Use Case name: | Passenger can register to the system |
| Actors: | Passenger |
| Description: | Passenger can register themself as a ‘Passenger’. |
| Trigger: | Passenger clicks on the register button when starting the application. |
| Pre-conditions: | N/A |
| Post-conditions: | Passenger can log into the system. (UC-14) |
| Normal Flows: | 1.Passenger clicks on the register button when starting the application.  2.The system displays the registration form.  3.Passenger inputs their particulars.  4.System validates the information.  5.The system creates the new user in the database and displays the” registration successful” message.  6.The system displays the log in page after the registration. |
| Alternative Flows: | N/A |
| Exceptions: | 4A If Passenger inputs wrong format, the system will notify ‘Wrong format!’ The Passenger can redo the 3rd step again.  4B If Passenger inputs an existing username, the system will notify, ‘This username already exists.’ The Passenger can redo the 3rd step again. |
| Includes: |  |
| Note and Issues: |  |

**UC-12 The Passenger can register to the system.**

**Introduction:** In this use case the Passenger can register themself as a ‘Passenger’.

**Input:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Example** | **Note** |
| Username | Somchai99 | Alphabet A-Z or Integer 0-9 The length is not more than 10 characters. |
| Password | Love66 | Alphabet A-Z or Integer 0-9 The length is 4-8 characters. |
| Name | Somchai | Alphabet A-Z. The length is 20 characters. |
| Last Name | Rakrian | Alphabet A-Z. The length is 20 characters. |
| PassengerPicture | aa.jpg | The image format should be in .jpg. |

**Action:**

**Output:** If successful, the system will display the message that the registration is successful and display the log in page.

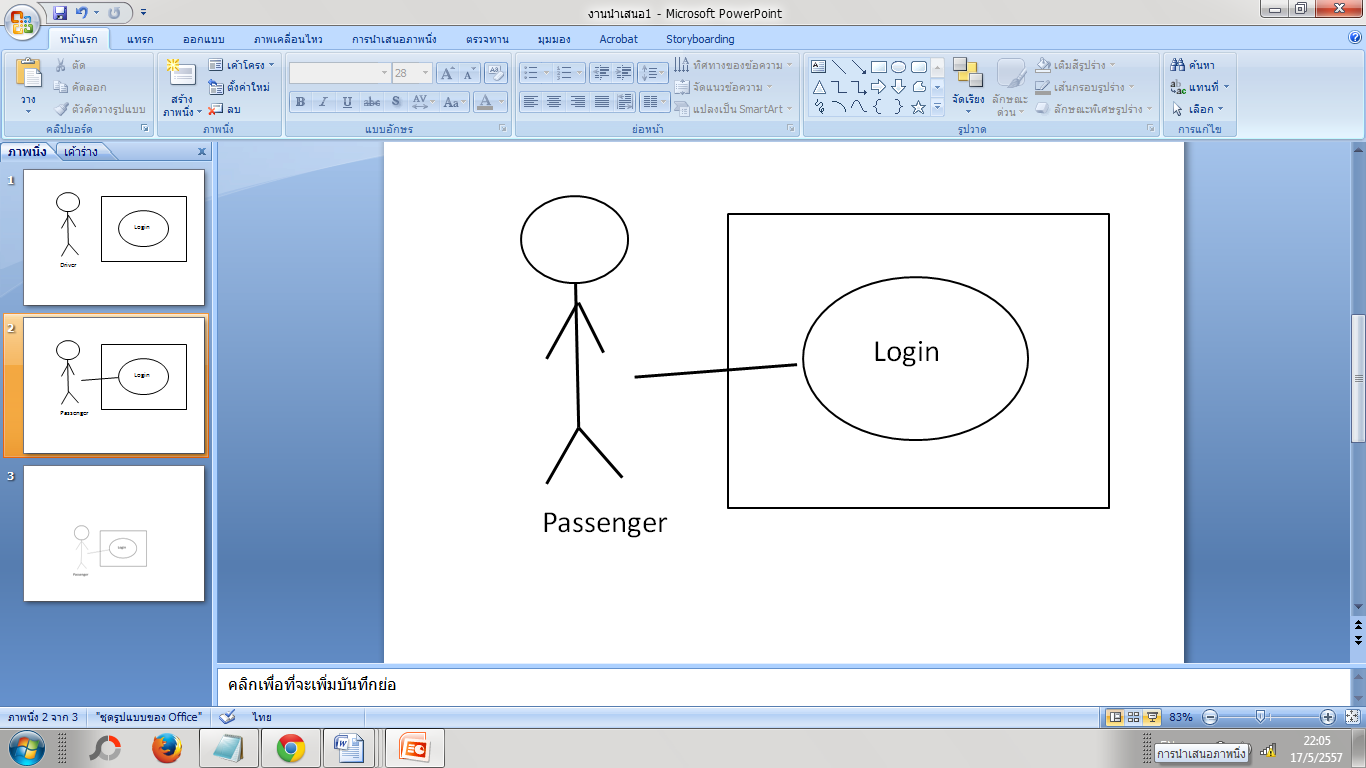
**System Requirement Specification**

**SRS-32**The system displays the registration form to Passenger

**SRS-33** The system will notify ‘Wrong format!’ If Passenger inputs wrong format

**SRS-34** The system will notify, ‘This username already exists.’ If Passenger inputs an existing username

4.14 Passenger Log In Use Case (UC-13)

**

*Figure 16: Passenger can log in to the system (URS-13)*

|  |  |
| --- | --- |
| Use Case ID | UC-13 |
| Use Case name: | Passenger can log in to the system |
| Actors: | Passenger |
| Description: | Passenger can log in to the system for accessing the system as a ‘Passenger’. |
| Trigger: | Passenger clicks on ‘Passenger log in’ button in the login page. |
| Pre-conditions: | Passenger registered in the system. (UC-12) |
| Post-conditions: | * Passenger can log out. (UC-06) * Passenger can access the application |
| Normal Flows: | 1. Passenger click on ‘Passenger log in’ button in the login page. 2. System displays the login page. 3. Passenger input the username and password. 4. Sytem verify the username and password. 5. System updates the passenger status to online status in the database. 6. System displays the login successful message. |
| Alternative Flows: | N/A |
| Exceptions: | 4A If Passengerinputs the username or password wrongly, the system will notify, ‘This username or password is wrong.’ Passenger re-enters the username and password again and can proceed on to step 5. |
| Includes: |  |
| Note and Issues: |  |

**UC-13 Passenger can log in to the system.**

**Introduction:** In this use case the Passenger can log in to the system for accessing the system as a ‘Passenger’.

**Input:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Example** | **Note** |
| Username | Somchai99 | Alphabet A-Z or Integer 0-9 The length is not more than 10 characters. |
| Password | Love66 | Alphabet A-Z or Integer 0-9 The length is 4-8 characters. |

**Action:**

*r*

**Output:** If successful, the system will display the message that the log in is successful.

**System Requirement Specification**

**SRS-35** System displays the login page to Passenger

**SRS-36** System will notify, ‘This username or password is wrong.’If Passengerinputs the username or password wrongly,

**SRS-37** System updates the passenger status to online status in the database.

**SRS-38** System displays the login successful message to passenger.