Use Cases

for

CarePax

**Version 1.0**

**Prepared by CS 411 A1 Team 3**

**David Monti, Ryan King, Kevin Leung, Baiming Zhang**

**September 23, 2015**

1. **Guidance for Use Case Template**

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. **Use Case Identification**

## Use Case ID

Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X.Y. Related use cases can be grouped in the hierarchy.

## Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

* View part number information.
* Manually mark hypertext source and establish link to target.
* Place an order for a CD with the updated software version.

## Use Case History

### Created By

Supply the name of the person who initially documented this use case.

### Date Created

Enter the date on which the use case was initially documented.

### Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

### Date Last Updated

Enter the date on which the use case was most recently updated.

1. **Use Case Definition**

## Actors

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

## Trigger

Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.

## Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

## Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

## Normal Flow

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered “X.0”, where “X” is the Use Case ID.

## Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form “X.Y”, where “X” is the Use Case ID and Y is a sequence number for the alternative flow. For example, “5.3” would indicate the third alternative flow for use case number 5.

## Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form “X.Y.E.Z”, where “X” is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E.2” would indicate the second exception for the normal flow for use case number 5.

## Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

## Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## Business Rules

List any business rules that influence this use case.

## Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

## Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

## Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case List

|  |  |  |
| --- | --- | --- |
| ***ID*** | ***Primary Actor*** | ***Use Case Title*** |
| 1 | Product Purchaser | Search for Nearby Drivers |
| 2 | Product Purchaser | View available products and pricing |
| 3 | Product Purchaser | Send Order |
| 4 | Drivers (Suppliers) | Receive and process order |

Use Case 1

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Search for Nearby Drivers | | |
| Created By: | Team 3 | Last Updated By: | Team 3 |
| Date Created: | September 23, 2015 | Date Last Updated: | September 23, 2015 |

|  |  |
| --- | --- |
| Actors: | The purchaser of a convenience product. The driver of a convenience product car. |
| Description: | This use case exists to outline the process of a customer of CarePax searching for nearby vehicles that can provide him or her with the convenience goods they need. The outcome of this case is that the customer will know which cars are closest to him or her and be able to see it in a Map View. |
| Trigger: | Customer loads home page and clicks on the “Find Drivers Near Me!” Button to load the Driver Locator Page |
| Preconditions: | 1. The customer has internet access 2. The customer is using a browser that is optimized to load a Google Maps view of his or her current location 3. If location services are deactivated, the customer has the ability to type his or her address of delivery 4. The customer has an authenticated account with CarePax 5. The Driver is online and available to receive orders |
| Postconditions: | The user is aware of the number of drivers around them and their relative distance away from their location. They also know which cars have a low stock of items, as indicated by the car icon’s color or textual indication. |
| Normal Flow: | 1.1 The user navigates to the website  1.2 The user clicks on the button labeled “Find Drivers Near Me!”  1.3 The web browser navigates them to a page featuring a large map.  1.4 A pop up appears asking the user to approve sharing their location.  1.5 The user sees their own location along with images of cars driving around them. |
| Alternative Flows: | 1.1 The user navigates to the website.  1.2 The user clicks on the button labeled “Find Drivers Near Me!”  1.3 The web browser navigates them to a page featuring a larger map.  1.4 A pop up appears asking the user to approve sharing their location.  1.5.A The browser prompts them that their location could not be automatically detected.  1.5.B The customer does not approve the sharing of their location  1.6 The customer enters his or her address manually into a search bar above the map  1.7 The user sees the specified location along with images of cars driving around them. |
| Exceptions: | 1.0.E.1 Navigation failed  The website shows a message and requires users to refresh the website.  2.0.E.1 No drivers nearby  The website asks purchaser to wait few minutes and retry  while the server tries to find the nearest free driver.  5.1.E.1 Location unavailable  A prompt appears asking them to input their address.  The user fills out their address and hits okay.  The map snaps to their location.  The user sees their own location along with images of cars driving around them on the Map View.  5.1.E.2 Location out of service area  The website shows a message with apology and informs them of areas where CarePax operates.  5.1.E.3 Customers rejects the approval of location services  The website asks customer to manually enter address. |
| Includes: | 1.View available products and pricing  3.Send Order |
| Priority: | *Ranked on a priority scale of 1 lowest to 5 highest*  Google Maps API Integration: 4  Location Services for the Customer: 3  Location Services for the Driver: 5  Graphical Overlay of Vehicle positions: 1  Search bar feature for address: 4  Ability to list nearby drivers: 5  Indicate lower-inventory drivers: 2  Website Rendering: 5 |
| Frequency of Use: | The use case will be performed one time by the actor every time they wish to make a purchase or see if a car is close enough to them with inventory to decide whether or not they want to make the purchase at the current time. |
| Business Rules: | The driver must be verified through a background check. |
| Special Requirements: | Preferences and filterings based on the product and pricing are special requirements that need to be addressed. It should show how long the driver has been “away” from the app for his/her phone. Also, The browser must be able to support the Google Maps API. |
| Assumptions: | Assumes that the driver is online and is connected to Wifi and that the APIs used to locate the driver and the purchaser are correct. There is an assumption that the driver being shown as online actually makes them available to receive orders. |
| Notes and Issues: | How small or big should the scope of available drivers be? Should this be determined by the city or the state? Can drivers choose to be available only to people within a certain range (i.e. 5 miles maximum)? |

Use Case 2

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | View Available Products and Pricing | | |
| Created By: | Team 3 | Last Updated By: | Team 3 |
| Date Created: | October 9, 2015 | Date Last Updated: | October 23, 2015 |

|  |  |
| --- | --- |
| Actors: | The purchaser of a convenience product. The driver of a convenience product car. |
| Description: | After the customer generates a query for nearby cars, he or she will click on one of them to see what products are available in the driver’s inventory. If the customer does not see anything to his or her liking, he or she can simply click the X on the inventory window to go back to the map of cars. |
| Trigger: | Customer clicks on a car to open a page with the available items and pricing. |
| Preconditions: | 1. The customer has internet access 2. The customer is using a browser that is optimized to load a Google Maps view of his or her current location 3. There are cars on the map that the user can click on 4. The Driver is online and available to receive orders |
| Postconditions: | The customer sees which products are available within that particular car. |
| Normal Flow: | 1.1 The user navigates on the Google Maps API to find the desired vehicle.  1.2 Once the user sees a car that he or she is interested in, he or she will click on that car.  1.3 A window magnifies from the car as the background of the site blurs.  1.4 The window displays the to user a list of every item that is available in that car. Green, yellow, or red circles will appear next to the items to indicate inventory levels. Prices will also appear next to the items. |
| Alternative Flows: | 1.1 The user navigates on the Google Maps API to find the desired vehicle.  1.2 The user searches for a specific convenience item of interest.  1.3 The user is shown the inventory of the closest car with that item.  1.4 The window displays the to user a list of every item that is available in that car. Green, yellow, or red circles will appear next to the items to indicate inventory levels. Prices will also appear next to the items. |
| Exceptions: | 1.0.E.1 Navigation failed  The website shows a message and requires users to refresh the website.  2.0.E.1 No drivers nearby  The website asks purchaser to wait few minutes and retry  while the server tries to find the nearest free driver.  5.1.E.1 Location unavailable  A prompt appears asking them to input their address.  The user fills out their address and hits okay.  The map snaps to their location.  The user sees their own location along with images of cars driving around them on the Map View.  5.1.E.2 Location out of service area  The website shows a message with apology and informs them of areas where CarePax operates.  5.1.E.3 Customers rejects the approval of location services  The website asks customer to manually enter address.  5.1.E.4 No driver has the item searched |
| Includes: | 1.Search for Nearby Drivers  3.Send Order |
| Priority: | *Ranked on a priority scale of 1 lowest to 5 highest*  Google Maps API Integration: 4  Location Services for the Customer: 3  Location Services for the Driver: 5  Graphical Overlay of Vehicle positions: 1  Search bar feature for items: 3  Ability to list nearby drivers: 5  Indicate lower-inventory drivers: 5  Inventory window appears correctly: 5  Website Rendering: 5 |
| Frequency of Use: | The use case will be performed one time by the actor every time they wish to make a purchase or see if a car is close enough to them with inventory to decide whether or not they want to make the purchase at the current time. |
| Business Rules: | The driver must be verified through a background check. The driver must have appropriate inventory levels. |
| Special Requirements: | Preferences and filterings based on the product and pricing are special requirements that need to be addressed. It should show how long the driver has been “away” from the app for his/her phone. Also, The browser must be able to support the Google Maps API. |
| Assumptions: | Assumes that the driver is online and is connected to Wifi and that the APIs used to locate the driver and the purchaser are correct. There is an assumption that the driver being shown as online actually makes them available to receive orders. There is also an assumption that the driver has at least some items that can be ordered by the user. Also, the user must have an account to be able to see the items in the cars. |
| Notes and Issues: | How small or big should the scope of available drivers be? Should this be determined by the city or the state? Can drivers choose to be available only to people within a certain range (i.e. 5 miles maximum)? How much inventory is too little inventory to mark an item as red? If a driver has very little inventory will we even show them? |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Team 3 | 9/23/15 | Creation | 1 |
|  |  |  |  |