| **ID** | **Date Added** | | **Priority** | **Description** | **Use Case** | **Implemented in Iteration #** |
| --- | --- | --- | --- | --- | --- | --- |
| REQ1 | | 11/08/15 | ESSENTIAL | The system shall allow for anonymous users to view the location of food trucks on a map. | UC-1, AnyUser : ViewMap | 1 |
| REQ2 | | 11/08/15 | ESSENTIAL | The system shall allow for food truck companies to register and indicate that they are a truck owner. | UC-2, AnonymousUser : RegisterTruck | 2 |
| REQ3 | | 11/08/15 | ESSENTIAL | The system shall allow for the food truck companies to create a profile for their truck, including their truck’s name, website, and description. | UC-3, RegisteredTruck : UpdateTruckProfile | 2 |
| REQ4 | | 11/08/15 | LOW | The system should allow for the food truck companies to list upcoming and planned locations for their truck(s). | UC-4, RegisteredTruck : ListFutureLocations | future |
| REQ5 | | 11/08/15 | HIGH | The system shall allow for users to create accounts. | UC-5, AnonymousUser : RegisterUser | 2 |
| REQ6 | | 11/08/15 | LOW | The system shall allow logged in users to add trucks to a favorites listing. | UC-6, RegisteredUser : AddToFavorites | 4 |
| REQ7 | | 11/08/15 | LOW | The system shall allow anonymous or logged in users to search for a truck by truck name. | UC-7, All Users : SearchByName | future |
| REQ8 | | 11/08/15 | LOW | The system shall allow for anonymous or logged in users to center the map by address (full address; city and state or zip). | UC-8, All Users : AddressCenter | future |
| REQ9 | | 11/08/15 | ESSENTIAL | The system shall use Google Maps to display a map with identified truck locations. | UC-9, All Users : ViewTrucksOnMap | 1 |
| REQ10 | | 11/08/15 | LOW | The system should be able to detect the GPS location of users and center the map around the GPS location. | UC-10, All Users : AutoCenter | 2 |
| REQ11 | | 11/08/15 | ESSENTIAL | The system shall use user authentication with an encrypted password for all registered accounts. | N/A - not a separate use case  (part of UC-2 and UC-5) | 2 |
| REQ12 | | 11/08/15 | HIGH | The system’s user interface must follow Google’s material design rules. | N/A - not a separate use case | 3 |
| REQ13 | | 11/08/15 | HIGH | The system must incorporate periodic (daily) database backups. | N/A - not a separate use case | 3 |
| REQ14 | | 11/08/15 | ESSENTIAL | The system must be compatible with Android OS from Jelly Bean (API 16) to Marshmallow (API 23). | N/A - not a separate use case | 1 |
| REQ15 | | 11/23/15 | LOW | User is able to see profile for selected trucks,, including the current address and truck’s website. | UC-11, All Users : DisplayTruckInfo | Future |
| REQ16 | | 11/23/15 | LOW | When a truck is clicked on the map, the ability to add the truck to a favorites list appears. | UC-12, RegisteredUser : ClickTruck | 4 |
| REQ17 | | 11/23/15 | LOW | The system should allow for a list view of all trucks. | UC-13, All Users : ListView | future |
| REQ18 | | 11/23/15 | MEDIUM | The system should include a navigation drawer. | N/A - not a separate use case | 3 |
| REQ19 | | 12/7/15 | LOW | When a user attempts to register with an existing username, an error is transmitted to the user, and an account is not created. | UC-14, AnonymousUser: DuplicateUsernameReject | 4 |
| REQ20 | | 12/7/15 | MEDIUM | Logged in users can see a list of all trucks that they have added to their favorites list. | UC-15, RegisteredUser : FavoritesView | 4 |
| REQ21 | | 12/7/15 | LOW | Include a field for truck owners to select their food type. | N/A - not a separate use case | future |
| REQ22 | | 12/7/15 | LOW | Ability for users to filter by truck food type. | UC-16, AllUsers : FilterView | future |
| REQ23 | | 12/7/15 | HIGH | The system shall allow for truck owners to set their truck’s location by checking in at their current location. | UC-17, TruckUser : SetLocation | 4 |

**UML Scenarios and/or User Stories for Implemented Use Cases:**

| **Use Case** | **UML Scenario and/or User Story** |
| --- | --- |
| UC-1, AnyUser : ViewMap | As an anonymous user, I can view a map, and if food trucks are in the area of focus, they will be indicated. |
| UC-2, AnonymousUser : RegisterTruck | As a food truck company owner, I can register on the “Lunch on Wheels” application and configure a password for my account. |
| UC-3, RegisteredTruck : UpdateTruckProfile | As a food truck company owner, I can update a profile for my food truck. |
| UC-5, AnonymousUser : RegisterUser | As a user I want to be able to make an account, and be able to log in using a username and password combination that was created during the registration process. |
| UC-6, RegisteredUser : AddToFavorites |  |
| UC-9, All Users : ViewTrucksOnMap | As an anonymous user, I can view a map, and if food trucks are in the area of focus, they will be indicated. |
| UC-10, All Users : AutoCenter | As an anonymous user when I open the app it will use my GPS location, if activated, to center the screen. And will show me any nearby food trucks and their locations. |
| UC-12, RegisteredUser : ClickTruck |  |
| UC-14, AnonymousUser: DuplicateUsernameReject |  |
| UC-15, RegisteredUser : FavoritesView |  |
| UC-17, TruckUser : SetLocation | As a food truck company owner, I can list the current location of my truck. |

**Use Case Details for Key Use Cases:**

|  |
| --- |
| Use Case UC-#: UC-6 Add to Favorites |
| Related Requirements: REQ6, REQ16  Initiating Actor: Logged in user  Actor’s Goal: To add a truck and its information to a favorites list  Participating Actors: N/A  Pre-conditions: A truck has been clicked on.  Google API is connected.  Postconditions: The favorite is added to the Mysql database and can be seen in the favorites list in the app.  **Flow of Events for Main Success Scenario:**  → 1. The user clicks on the popup that is over the selected truck.  ← 2. The truck is added to the user’s favorites list in the MySQL database.  **Flow of Events for Extensions (Alternate Scenarios):**  What could go wrong? List the exceptions to the routine and describe how they are handled  → 1a. User clicks on the popup of a truck that is already in the user’s favorites list.  ← 1b. A duplicate favorites entry is not added to MySQL database. |

|  |
| --- |
| User Case UC#: UC-12 Click on Truck |
| Related Requirements: REQ16, REQ9  Initiating Actor: Logged in user  Actor’s Goal: To click on a truck and cause a pop-up to appear offering the option to add the truck to a favorites list.  Participating Actors: N/A  Pre-conditions: The app is open to the map activity. At least one truck is visible on the map. Google API is connected.  Postconditions: A pop-up is visible over the clicked truck. An option to add the truck to the user’s favorites appears.  Flow of Events for Main Success Scenario:  → 1.User clicks on truck.  ← 2. A pop-up appears over the truck, offering the option to add the truck to a favorites list.  Flow of Events for Extensions (Alternate Scenarios):  What could go wrong? List the exceptions to the routine and describe how they are handled  1a. No database connection  ← 1b. No list of trucks is displayed by the system.  1c. No trucks have their location listed.  ← 1d. No list of trucks is displayed by the system. |

|  |
| --- |
| User Case UC#: UC-14 Duplicate Username Reject |
| Related Requirements: REQ19, REQ2, REQ5  Initiating Actor: Anonymous user  Actor’s Goal: To attempt to create an account with an existing username  Participating Actors: N/A  Pre-conditions: The app is open to the registration activity. Google API is connected.  Postconditions: An error message is displayed.  **Flow of Events for Main Success Scenario:**  → 1.User completes profile with all required fields and with the username entered as an existing username.  ← 2. System displays an error message and does not create account.  **Flow of Events for Extensions (Alternate Scenarios):**  What could go wrong? List the exceptions to the routine and describe how they are handled  1a. No database connection  ← 1b. Unable to verify the information in the MySQL database. |

|  |
| --- |
| User Case UC#: UC-15 Favorites View |
| Related Requirements: REQ20, REQ18  Initiating Actor: Logged in user  Actor’s Goal: To click on the favorites link in the navigation drawer to view a list of the user’s favorites trucks.  Participating Actors: N/A  Pre-conditions: The navigation drawer is open.  Postconditions: A list of favorite trucks is displayed.  Flow of Events for Main Success Scenario:  → 1.User clicks on “Favorites” link.  2. include::NavDrawer (REQ18)  ← 3. System displays list of the user’s favorites trucks.  Flow of Events for Extensions (Alternate Scenarios):  What could go wrong? List the exceptions to the routine and describe how they are handled  3a. No database connection  ← 3b. No list of favorite trucks is displayed by the system.  3c. The user does not have any favorite trucks.  ← 3d. No list of favorite trucks is displayed by the system. |

|  |
| --- |
| User Case UC#: UC-17 Set Location |
| Related Requirements: REQ23,REQ18  Initiating Actor: Truck User  Actor’s Goal: To click on the “Set Location” option in the navigation drawer to set the truck’s current location.  Participating Actors: N/A  Pre-conditions: The navigation drawer is open.  Postconditions: The truck’s location is set in the database, and a location icon is visible on the map at the selected location.  **Flow of Events for Main Success Scenario:**  → 1.User clicks on “Set Location” link.  2. include::NavDrawer (REQ18)  ← 3. System adds truck’s location to database.  ← 4. System displays truck’s icon at current location.  **Flow of Events for Extensions (Alternate Scenarios):**  What could go wrong? List the exceptions to the routine and describe how they are handled  3a. No database connection  ← 3b. No location is written to database, and no truck icon is displayed at current location.  3c. GPS not enabled on device.  ← 3d. No location is written to database, and no truck icon is displayed at current location. |