Definitions (1,2,3)

* Blog page – a webpage with content organized like a blog. The page’s content is composed entirely of blog posts, ordered by published date with the most recent posts displayed first. Example blog pages include the following: Home page, Blog page, Q&A page. A blog page also contains a ‘keyword’ text field and a ‘search’ button, which allows a user to search for blog posts containing a user-entered keyword.
* Blog search results page – A special kind of blog page, which contains only BlogPosts that contain the keyword searched by the user.
* BlogPost – The content that goes into a blog page. BlogPosts can only be created and edited by administrators.

Input Domain (1,2,3)

* Blog page
  + Blog posts <Collection of BlogPost> The posts on the page, ordered by published date with the most recent posts displayed first
  + PostsPerPage <Number> The number of posts that can be viewed on a single screen. If multiple screens are required to view all content, the blog page UI will provide a mechanism for navigating between screens.
* BlogPost (Article?)
  + Required parameters
    - Published <Boolean> True if the blog post has been published to a blog page
    - Title <Text> The name of this blog post
    - Keywords <Collection of Text> A list of words which categorize this post
    - Text (Content?) <Text> The content of the blog post. <TODO: confirm> This will be html-formatted, so that additional media like pictures and video can be added.
  + Calculated parameters
    - PublishDate <DateTime> Set when the blog post is published by an administrator
    - Author <User> The administrator which created this BlogPost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** User | | | **ID:** 1 | **Importance Level:** Medium |
| **Use Case Name:** Search blog posts by keyword | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a user, I want to view a blog page, such as the home page, blog page, or Q&A page. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | |  | | |
| **Post-Conditions:** | |  | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user navigates to a blog webpage | | |
|  | | |
| **Normal Flow of Events:** | | 1. The blog webpage is shown. All blog posts for the page are displayed, with the most recent shown first. | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  Alt 1) If no posts match the user’s keyword, the ‘Blog Search Results’ webpage is shown, but instead of a listing of posts, a message displays indicating that no matching posts were found. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** User | | | **ID:** 2 | **Importance Level:** Medium |
| **Use Case Name:** Search blog posts by keyword | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a user, I want to search a blog for all posts containing some keyword. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User has navigated to a blog page | | |
| **Post-Conditions:** | |  | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user types a keyword into the ‘keyword’ text field on a blog page and issues the ‘search’ command | | |
|  | | |
| **Normal Flow of Events:** | | 1. The ‘Blog Search Results’ webpage is shown. All blog posts which contain the user’s keyword are displayed, with the most recent shown first. (See Alt 1) | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  Alt 1) If no posts match the user’s keyword, the ‘Blog Search Results’ webpage is shown, but instead of a listing of posts, a message displays indicating that no matching posts were found. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *4* | **Importance Level:** High/Medium/Low *High* |
| **Use Case Name:** *Customer Login* | | | **Use Case Type:**  Overview/Detail *A customer shall be able to login to the system.* | |
| **Stakeholders:** | | *Customer, Administrator* | | |
| **Goal:** | | *Allow a customer to login to the system.* | | |
| **Quality requirements:** | | *3. The system shall allow a user to have 5 incorrect logins before locking the user’s account for 3 hours.* | | |
| **Pre-Conditions:** | | *The Customer must not be logged in currently.* | | |
| **Post-Conditions:** | | *The Customer must be logged in.* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A User selects the Login option* | | |
| **Normal Flow of Events:** | | 1. *The User enters their email address and password* 2. *The User selects the submit option* 3. *The System validates the email address and password* 4. *The User is authenticated* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *3.a. Invalid Email Address*  *3.a.1 Systems displays error message “Invalid email address or password”*  *3.b. Invalid Password*  *3.b.1 System displays error message “Invalid email address or password”* | | | | |
| **Input Specifications:**  Email := RFC 2822 compliant | | | | |
| **Functional Specification:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *5* | **Importance Level:** High/Medium/Low *High* |
| **Use Case Name:** *Customer Create Account* | | | **Use Case Type:**  Overview/Detail *Overview – very high level view of the requirement (lacks detail)* | |
| **Stakeholders:** | | *Customer, Administrator* | | |
| **Goal:** | | *Allow a customer to create an account* | | |
| **Quality requirements:** | | *5. The system shall commit user updates to his/her account information within 3 seconds.*  *14. The system shall hash and salt passwords as follows, SHA-224((first 6 characters of password) append (MD5 of username) append (remaining characters of password in reverse order)).* | | |
| **Pre-Conditions:** | | *None* | | |
| **Post-Conditions:** | | *The Customer must have an account* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A Customer selects the create account option* | | |
| **Normal Flow of Events:** | | 1. *A Customer enters their email address and password, and they confirm their password.* 2. *The System validates that the given email address is not in use.* 3. *The System validates that the given password and confirmation password match.* 4. *The System updates persistent storage with the new account.* 5. *The Customer is authenticated.* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *2.a The email address is already in use*  *2.a.1 The System displays the error “This email address is already in use”*  *3.a The password and password confirmation do not match*  *3.a.1 The System displays the error “The password and password confirmation do not match”* | | | | |
| **Input Specification:**  Email := RFC 2822 compliant | | | | |
| **Functional Specification:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *6* | **Importance Level:** High/Medium/Low *High* |
| **Use Case Name:** *Customer View Order History* | | | **Use Case Type:**  Overview/Detail *Overview – very high level view of the requirement (lacks detail)* | |
| **Stakeholders:** | | *Customer* | | |
| **Goal:** | | *Allow a Customer to view their order history* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *The Customer must be logged in* | | |
| **Post-Conditions:** | | *None* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A Customer selects the view order history* | | |
| **Normal Flow of Events:** | | 1. *System displays the Customer’s order history. Each order includes the price, quantity, whether it has been paid for or not, the pickup date, and the total.* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *None* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *7* | **Importance Level:** High/Medium/Low *High* |
| **Use Case Name:** *Customer Reset Forgotten Password* | | | **Use Case Type:**  Overview/Detail *Overview – very high level view of the requirement (lacks detail)* | |
| **Stakeholders:** | | *Customer* | | |
| **Goal:** | | *A Customer shall be able to reset their forgotten password.* | | |
| **Quality requirements:** | | *4. When a user requests a password reset, the system shall reset the user’s password within one hour.*  *14. The system shall hash and salt passwords as follows, SHA-224((first 6 characters of password) append (MD5 of username) append (remaining characters of password in reverse order)).* | | |
| **Pre-Conditions:** | | *The Customer must have an account.* | | |
| **Post-Conditions:** | | *The Customer’s account password must be reset.* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *The Customer selects the Reset Forgotten Password Option* | | |
| **Normal Flow of Events:** | | 1. *The Customer enters their email address* 2. *The System validates the email address* 3. *The System sends and email with a link to reset the password for the account with the given email address to the given email address.* 4. *The Customer navigates to the link* 5. *The Customer enters their new password and password confirmation.* 6. *The System validates that the passwords match.* 7. *The Customer’s account is updated in persistent storage.* 8. *The Customer is authenticated.* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *2.a. Invalid email address*  *2.a.1. The System displays “Invalid email address”*  *6.a. Passwords do not match*  *6.a.1. The System displays “The passwords do not match”* | | | | |
| **Input Specifications:**  Email := RFC 2822 compliant | | | | |
| **Functional Specifications:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *State the product title* | | | **Product version:** *State the product version number* | |
| **Primary Actors:** Customer | | | **ID:** 8 | **Importance Level:**  *High – must be implemented* |
| **Use Case Name:** View Available Products | | | **Use Case Type:**  Detail | |
| **Stakeholders:** | | Customer | | |
| **Goal:** | | The customer has requested to view available products. | | |
| **Quality requirements:** | | The system shall display available products within 3 seconds of a customer requesting to view the available products. | | |
| **Pre-Conditions:** | | None | | |
| **Post-Conditions:** | | All available products will be displayed for the user to view. | | |
| **Relationships:** | |  | | |
|  | **Include:** | None | | |
| **Extend:** | None | | |
| **Generalization:** | None | | |
| **Trigger:** | | **Trigger Type:** External | | |
| 1. The customer issues a request to view available products. | | |
| **Normal Flow of Events:** | | 2. The system queries the data store for a list of available products.  3. The data store returns a list of all available products.  4. The system displays all available products for the customer to view. | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows: 3a. The data store returns an empty list of available products**  3a1.The system displays a message informing the user that no products are available. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *Use Case 4.9a* | **Importance Level:** High/Medium/Low *High – must be implemented* |
| **Use Case Name:** *Add Product to Shopping Cart* | | | **Use Case Type:**  Overview/Detail *Detail* | |
| **Stakeholders:** | | *Customers* | | |
| **Goal:** | | *The system shall allow a Customer to add products to a shopping cart* | | |
| **Quality requirements:** | | *Non-Functional Requirements Reference (refer to Jonesborough Farmer’s Market Top Level Requirements Document)*  *6. When a user adds an item to his/her shopping cart, the system shall update the user’s shopping cart within 2 seconds.*  *7. When a user removes an item from his/her shopping cart, the system shall update the user’s shopping cart within 2 seconds.*  *8. The system shall create a receipt for a completed checkout within 3 seconds of the transactions completion.*  *9. The system shall update the total for the user’s cart within 3 seconds of any change made to the cart.* | | |
| **Pre-Conditions:** | | *The product must exist in inventory with quantity greater than the quantity being purchased.* | | |
| **Post-Conditions:** | | *The quantity of the product in inventory is reduced by the quantity of product purchased.* | | |
| **Relationships:** | |  | | |
|  | **Include:** | * *Specify Quantity* | | |
| **Extend:** | * *Add to Shopping Cart* * *Login* * *Create Account* | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** *External* | | |
| 1. *The Customer clicks in the “Qty:” field, types in the number of product to add, and clicks the “Add to Cart” button* | | |
| **Normal Flow of Events:** | | 1. *The Item Quantity is updated on shopping cart display* | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** *EX1*  *2a. If the Customer is not logged in, the Login page is displayed.*  *2b. If the Customer is not registered, the Create Account page is displayed.*  *2c. If the specified quantity exceeds the quantity of product in inventory, the Insufficient Inventory page is displayed* | | | | |
| **UI: To be Added** | | | | |
| **Input Specifications:**  Qty ::= <quantity>  <quantity> ::= 0{integer}999 | | | | |
| **Functional Specifications:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *Use Case 4.9b* | **Importance Level:** High/Medium/Low *High – must be implemented* |
| **Use Case Name:** *View and Edit Shopping Cart* | | | **Use Case Type:**  Overview/Detail *Detail* | |
| **Stakeholders:** | | *Customers* | | |
| **Goal:** | | *The system shall allow a Customer to view their shopping cart and edit a shopping cart* | | |
| **Quality requirements:** | | *Non-Functional Requirements Reference (refer to Jonesborough Farmer’s Market Top Level Requirements Document)*  *6. When a user adds an item to his/her shopping cart, the system shall update the user’s shopping cart within 2 seconds.*  *7. When a user removes an item from his/her shopping cart, the system shall update the user’s shopping cart within 2 seconds.*  *8. The system shall create a receipt for a completed checkout within 3 seconds of the transactions completion.*  *9. The system shall update the total for the user’s cart within 3 seconds of any change made to the cart.* | | |
| **Pre-Conditions:** | |  | | |
| **Post-Conditions:** | | * *The quantity of the product in shopping cart is reduced by the quantity of product removed from the shopping cart* * *The quantity of the product in inventory is increased by the quantity of product removed from the shopping cart* | | |
| **Relationships:** | |  | | |
|  | **Include:** | * *View Shopping Cart* * *Remove Product from Shopping Cart* | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** *External* | | |
| 1. *The Customer clicks in the “Shopping Cart” button* | | |
| **Normal Flow of Events:** | | 1. *The Shopping Cart page is displayed* 2. *The Customer checks the checkbox to the left of the product to be removed from the Shopping Cart* 3. *The Customer clicks the “Remove” button to the right of the product to be removed from the Shopping Cart* 4. *The Shopping Cart page is displayed without the product that was removed* | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |
| **UI: To be Added** | | | | |
| **Input Specifications:** | | | | |
| **Functional Specifications:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *Use Case 4.9c* | **Importance Level:** High/Medium/Low *High – must be implemented* |
| **Use Case Name:** *Checkout* | | | **Use Case Type:**  Overview/Detail *Detail* | |
| **Stakeholders:** | | *Customers* | | |
| **Goal:** | | *The system shall allow a Customer place an order for the products in their shopping cart (Checkout)* | | |
| **Quality requirements:** | | *Non-Functional Requirements Reference (refer to Jonesborough Farmer’s Market Top Level Requirements Document)*  *6. When a user adds an item to his/her shopping cart, the system shall update the user’s shopping cart within 2 seconds.*  *7. When a user removes an item from his/her shopping cart, the system shall update the user’s shopping cart within 2 seconds.*  *8. The system shall create a receipt for a completed checkout within 3 seconds of the transactions completion.*  *9. The system shall update the total for the user’s cart within 3 seconds of any change made to the cart.* | | |
| **Pre-Conditions:** | | *At least one product exists in the Shopping Cart* | | |
| **Post-Conditions:** | | *A receipt is created for the Purchase* | | |
| **Relationships:** | |  | | |
|  | **Include:** | * *View Oder Total* * *Update Account Information* * *Cancel Purchase* | | |
| **Extend:** | * *Submit Oder* * *View Receipt* | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** *External* | | |
| 1. *The Customer clicks in the “Checkout” button* | | |
| **Normal Flow of Events:** | | 1. *The Checkout page is displayed* 2. *The Customer clicks the “Submit Order” button* 3. *A receipt is displayed for the purchase* 4. *Customer clicks “OK”* 5. *A receipt is emailed to the Customer* 6. *Purchase Total is sent to the Point of Sale system* 7. *The View Available Products page is displayed* | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** EX1  3a. The Customer clicks the “Cancel Purchase” button  3b. The Shopping Cart page is displayed  EX2  3a. The Customer clicks the “Update Account Information” button  3b. The Update Account Information page is displayed | | | | |
| **UI: To be Added** | | | | |
| **Input Specifications:** | | | | |
| **Functional Specifications:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *State the product title* | | | **Product version:** *State the product version number* | |
| **Primary Actors:** Customer | | | **ID:** 10 | **Importance Level:** High |
| **Use Case Name:** Search by product name | | | **Use Case Type:**  Detail | |
| **Stakeholders:** | | Customer | | |
| **Goal:** | | The customer searches for a product | | |
| **Quality requirements:** | | The system shall display the product to the customer within 2 seconds of a customer issuing a search request. | | |
| **Pre-Conditions:** | | The system is operational. | | |
| **Post-Conditions:** | | The customer is viewing the product he/she was searching for. | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External | | |
| 1. The customer issues a search for a product by the product’s name. | | |
| **Normal Flow of Events:** | | 1. The system queries the data store for products matching the customer’s product name. 2. The data store returns a list of matching products 3. The system displays the list of matching products to the customer. | | |
| **Sub-flows:** | | None | | |
| **Alternative/Exception flows:** 3a. The data store returns a null/empty list of matching products  3a1. The system displays a message to the customer that no matching products were found. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| **Product title:** *Jonesborough Farmer's Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *11* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Donating Anonymously* | | | **Use Case Type:**  Overview/Detail *Customer makes a donation without using an account* | |
| **Stakeholders:** | | *Customer, Administrator* | | |
| **Goal:** | | *Allow a customer to donate anonymously* | | |
| **Quality requirements:** | | *10. The system will complete a money transaction for a purchase or donation within 5 business days.* | | |
| **Pre-Conditions:** | | *Customer is not logged in* | | |
| **Post-Conditions:** | | *The donation will be processed* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A Customer selects donate* | | |
| **Normal Flow of Events:** | | *1. The Customer enters his or her credit card information and the amount to donate*  *2. The System validates the inputs*  *3. The System charges the customer's card*  *4. The System displays the message “Thank you for your donation!”* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *2.a. The field(s) is/are invalid*  *2.a.1 The system displays an error message indicating the invalid fields*  *3.a. The card is declined*  *3.a.1 The system displays the message “Unfortunately the payment could not be processed.”* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| **Product title:** *Jonesborough Farmer's Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *13* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Donating Anonymously* | | | **Use Case Type:**  Overview/Detail *Customer views his or her donation history* | |
| **Stakeholders:** | | *Customer, Administrator* | | |
| **Goal:** | | *Allow a customer to view donation history* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *Customer is logged in* | | |
| **Post-Conditions:** | | *The donation history will be displayed* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A Customer selects donation history* | | |
| **Normal Flow of Events:** | | *1. The System displays the customer's donation history* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *None* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| **Product title:** *Jonesborough Farmer's Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *13* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Donating Anonymously* | | | **Use Case Type:**  Overview/Detail *Customer makes a donation using an account* | |
| **Stakeholders:** | | *Customer, Administrator* | | |
| **Goal:** | | *Allow a customer to donate* | | |
| **Quality requirements:** | | *10. The system will complete a money transaction for a purchase or donation within 5 business days.* | | |
| **Pre-Conditions:** | | *Customer is logged in* | | |
| **Post-Conditions:** | | *The donation will be processed* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A Customer selects donate* | | |
| **Normal Flow of Events:** | | *1. The Customer enters his or her credit card information and the amount to donate*  *2. The System validates the inputs*  *3. The System charges the customer's card*  *4. The System displays the message “Thank you for your donation!”* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *2.a. The field(s) is/are invalid*  *2.a.1 The system displays an error message indicating the invalid fields*  *3.a. The card is declined*  *3.a.1 The system displays the message “Unfortunately the payment could not be processed.”* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** User | | | **ID:** 15 | **Importance Level:** Medium |
| **Use Case Name:** View information on becoming a volunteer | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a user, I want to learn more about becoming a volunteer. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | |  | | |
| **Post-Conditions:** | |  | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user navigates to the ‘View Volunteer Information’ webpage | | |
|  | | |
| **Normal Flow of Events:** | | 1. The webpage displays information about becoming a volunteer. It also includes contact information if more information is required.   <TODO: Is this Use case really necessary?> | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Name the primary actor(s) | | | **ID:** 16 | **Importance Level:** Medium |
| **Use Case Name:** Sign up as volunteer (The system shall allow a volunteer to sign up as a volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a user, I want to register as a volunteer. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is not already flagged as a volunteer.  User has navigated to the ‘View Volunteer Information’ webpage | | |
| **Post-Conditions:** | | User account is flagged as a volunteer. User can access volunteer-only portions of the website. | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: User issues the ‘sign up as a volunteer’ command | | |
|  | | |
| **Normal Flow of Events:** | | 1. User issues the ‘sign up as a volunteer’ command    1. The system flags the user as a volunteer, enabling access to volunteer-specific contact. E-mail confirmations are sent to the user and administrators. A confirmation is displayed, and the system returns to the previous webpage. | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 16a-1 | **Importance Level:** Medium |
| **Use Case Name:** Add a ‘reoccurring availability’ timeslot (The system shall allow a volunteer to enter the days and times they are available to volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a volunteer, I want to indicate my availability by specifying a time when I am/am not available. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘Edit Availability’ webpage | | |
| **Post-Conditions:** | | An availability timeslot for this user is saved to the system | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user issues the ‘add reoccurring availability timeslot’ command. | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user issues the ‘add reoccurring availability timeslot’ command. A form appears. 2. The user specifies the following required fields (see ex 2):    1. Day    2. StartTime    3. EndTime (see ex 2c)    4. Available 3. The user may additionally specify the following optional field:    1. Comment 4. The user issues the ‘confirm’ command (see Alt 4, Ex 4). The timeslot is committed to the system. A confirmation displays, and the ‘Edit Availability’ webpage is shown. | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** Ex 2) If any of the required fields is not filled in with a valid value, the ‘confirm’ command shall be disabled.  Ex 2c) If the EndTime is earlier in the day than the StartTime (e.g. StartTime is 1:15 PM and EndTime is 1:00 PM), the ‘confirm’ command shall be disabled.  Alt 4) The user issues the ‘cancel’ command. A confirmation indicates no changes are made to the system, and the ‘Edit Availability’ webpage is shown.  Ex 4) If the specified reoccurring availability timeslot overlaps with another reoccurring availability timeslot, the system shall report an error indicating the conflicting timeslot(s). The timeslot shall not be saved. The same form shall be displayed again. | | | | |

Definitions

Volunteers can indicate their availability with two types of timeslot:

* Timeslot: a period of time with a start time and an end time. Both times are bound to the same day. The start time must be earlier in the day than the end time.
* Reoccurring availability timeslot: a timeslot when a volunteer is or is not available which repeats every week. Examples: Bob is available on Tuesdays from 7:00-10:00 PM. Bob is unavailable on Sundays from 10:00 AM-10:00 PM.
* Exceptional availability timeslot: a timeslot with no reoccurrence. A volunteer may or may not be available at this time. Examples: Bob is available on December 13, 2013 from 5:00-10:00 PM. Bob is unavailable on December 14, 2013 from 12:00 AM-11:59 PM.

Input Domain (16a-1, 16a-2)

* Reoccurring availability timeslot
  + Required parameters
    - Day <DayOfWeek> - the day of week for this availability period
    - StartTime <Time> - the start time for this availability period
    - EndTime <Time> - the end time for this availability period
    - Available <Boolean> true if volunteer is available during this time span, false if unavailable
  + Optional parameters
    - Comment <Text> - a place for the volunteer to indicate a reason for the availability, e.g., “Tuesday nights I have school, so I cannot volunteer.”
* Exceptional availability timeslot
  + Required parameters
    - Date <Date> - a calendar date for this availability period, e.g. Jan 1 2014.
    - StartTime <Time> - the start time for this availability period
    - EndTime <Time> - the end time for this availability period
    - Available <Boolean> true if volunteer is available during this time span, false if unavailable
  + Optional parameters
    - Comment <Text> - a place for the volunteer to indicate a reason for the availability, e.g., “Tuesday nights I have school, so I cannot volunteer.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 16a-2 | **Importance Level:** Medium |
| **Use Case Name:** Add an ‘exceptional availability’ timeslot (The system shall allow a volunteer to enter the days and times they are available to volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a volunteer, I want to indicate my availability by specifying a time when I am/am not available. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘Edit Availability’ webpage | | |
| **Post-Conditions:** | | An availability timeslot for this user is saved to the system | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user issues the ‘add exceptional availability timeslot’ command. | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user issues the ‘add exceptional availability timeslot’ command. A form appears. 2. The user specifies the following required fields (see ex 2):    1. Date (see ex 2a)    2. StartTime    3. EndTime (see ex 2c)    4. Available 3. The user may additionally specify the following optional field:    1. Comment 4. The user issues the ‘confirm’ command (see Alt 4, Ex 4). The timeslot is commited to the system. A confirmation displays, and the ‘Edit Availability’ webpage is shown. | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** Ex 2) If any of the required fields is not filled in with a valid value, the ‘confirm’ command shall be disabled.  Ex 2a) If the specified date is prior to today’s date, the ‘confirm’ command shall be disabled.  Ex 2c) If the EndTime is earlier in the day than the StartTime (e.g. StartTime is 1:15 PM and EndTime is 1:00 PM), the ‘confirm’ command shall be disabled.  Alt 4) The user issues the ‘cancel’ command. A confirmation indicates no changes are made to the system, and the ‘Edit Availability’ webpage is shown.  Ex 4) If the specified reoccurring availability timeslot overlaps with another exceptional availability timeslot, the system shall report an error indicating the conflicting timeslot(s). The timeslot shall not be saved. The same form shall be displayed again. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 16a-3 | **Importance Level:** Medium |
| **Use Case Name:** Modify a ‘reoccurring availability’ timeslot (The system shall allow a volunteer to enter the days and times they are available to volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a volunteer, I want to indicate my availability by specifying a time when I am/am not available. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘Edit Availability’ webpage  At least one ‘reoccurring availability’ timeslot exists for this user | | |
| **Post-Conditions:** | | An availability timeslot for this user is saved to the system | | |
| **Relationships:** | |  | | |
|  | **Include:** | 16a-2 | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user issues the ‘modify reoccurring availability timeslot’ command. | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user selects an existing reoccurring availability timeslot. 2. The user issues the ‘modify reoccurring availability timeslot’ command. A form appears. 3. <Follow steps 2-4 in 16a-1> | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 16a-4 | **Importance Level:** Medium |
| **Use Case Name:** Modify an ‘exceptional availability’ timeslot (The system shall allow a volunteer to enter the days and times they are available to volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a volunteer, I want to indicate my availability by specifying a time when I am/am not available. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘Edit Availability’ webpage  At least one ‘exceptional availability’ timeslot exists for this user | | |
| **Post-Conditions:** | | An availability timeslot for this user is saved to the system | | |
| **Relationships:** | |  | | |
|  | **Include:** | 16a-2 | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user issues the ‘modify exceptional availability timeslot’ command. | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user selects an existing exceptional availability timeslot. 2. The user issues the ‘modify exceptional availability timeslot’ command. A form appears. 3. <Follow steps 2-4 in 16a-2> | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 16a-5 | **Importance Level:** Medium |
| **Use Case Name:** Delete a ‘reoccurring availability’ timeslot (The system shall allow a volunteer to enter the days and times they are available to volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a volunteer, I want to indicate my availability by specifying a time when I am/am not available. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘Edit Availability’ webpage  At least one ‘reoccurring availability’ timeslot exists for this user | | |
| **Post-Conditions:** | | An availability timeslot for this user is saved to the system | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user issues the ‘modify reoccurring availability timeslot’ command. | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user selects an existing reoccurring availability timeslot. 2. The user issues the ‘delete reoccurring availability timeslot’ command. A confirmation prompt appears. 3. The user confirms the prompt. The selected timeslot is removed from the system. A confirmation displays, and the ‘Edit Availability’ webpage is shown. (see Alt 3) | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  Alt 3) The user cancels the prompt. No changes are made to the system. A confirmation displays, and the ‘Edit Availability’ webpage is shown. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 16a-6 | **Importance Level:** Medium |
| **Use Case Name:** Delete an ‘exceptional availability’ timeslot (The system shall allow a volunteer to enter the days and times they are available to volunteer) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | As a volunteer, I want to indicate my availability by specifying a time when I am/am not available. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘Edit Availability’ webpage  At least one ‘exceptional availability’ timeslot exists for this user | | |
| **Post-Conditions:** | | An availability timeslot for this user is saved to the system | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user issues the ‘modify exceptional availability timeslot’ command. | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user selects an existing exceptional availability timeslot. 2. The user issues the ‘delete exceptional availability timeslot’ command. A confirmation prompt appears. 3. The user confirms the prompt. The selected timeslot is removed from the system. A confirmation displays, and the ‘Edit Availability’ webpage is shown. (see Alt 3) | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  Alt 3) The user cancels the prompt. No changes are made to the system. A confirmation displays, and the ‘Edit Availability’ webpage is shown. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer | | | **ID:** 17 | **Importance Level:** Medium |
| **Use Case Name:** View Scheduled Timeslots | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | A volunteer should be able to view the timeslots of their schedule. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and has logged in.  User is a volunteer.  User has navigated to the ‘View Volunteer Schedule’ webpage | | |
| **Post-Conditions:** | |  | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user navigates to the ‘View Volunteer Schedule’ webpage | | |
|  | | |
| **Normal Flow of Events:** | | 1. The schedule displays all scheduled timeslots assigned to this user.   <TODO: A UI prototype is really important here> | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer, Administrator | | | **ID:** 17a | **Importance Level:** Medium |
| **Use Case Name:** Modify availability (confirm) | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | Volunteers should be able to confirm availability. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and logged in.  User is a volunteer.  User has navigated to the ‘View Volunteer Schedule’ webpage  At least one scheduled timeslot exists for this user | | |
| **Post-Conditions:** | | Volunteer’s schedule is confirmed. | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user selects a timeslot and issues the ‘confirm timeslot’ command | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user selects an existing timeslot. 2. The user issues the ‘confirm timeslot’ command. A checkbox associated with the timeslot is checked to indicate the user has confirmed. (See Alt 1) | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  Alt 1) If the timeslot has already been confirmed, issuing the ‘confirm timeslot’ command will toggle the timeslot back to unconfirmed. The associated checkbox is unchecked. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| *Insert the UI prototype screenshot here* | | | | |
| **Product title:** Jonesborough Farmers Market | | | **Product version:** 1.0 | |
| **Primary Actors:** Volunteer, Administrator | | | **ID:** 17b | **Importance Level:** Medium |
| **Use Case Name:** Report a scheduling conflict | | | **Use Case Type:**  Overview | |
| **Stakeholders:** | |  | | |
| **Goal:** | | Volunteers shall be able to report scheduling conflicts to his or her supervisor. | | |
| **Quality requirements:** | |  | | |
| **Pre-Conditions:** | | User is registered and logged in.  User is a volunteer.  User has navigated to the ‘View Volunteer Schedule’ webpage  At least one scheduled timeslot exists for this user | | |
| **Post-Conditions:** | |  | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External: The user selects a timeslot and issues the ‘report conflict’ command | | |
|  | | |
| **Normal Flow of Events:** | | 1. The user selects an existing timeslot. 2. The user issues the ‘report conflict’ command. A form appears. 3. The user specifies the following required field (see Alt 3):    1. Reason <Text> 4. The user issues the ‘confirm’ command (see Alt 4). The ‘Reason’ is sent as a message to the Administrators. | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  Alt 3) If any of the required fields is not filled in with a valid value, the ‘confirm’ command shall be disabled.  Alt 4) The user issues the ‘cancel’ command. A confirmation indicates no changes are made to the system, and the ‘View Volunteer Schedule’ webpage is shown. | | | | |

Definitions

* Scheduled timeslot: A timeslot which indicates when a volunteer is expected to work. These timeslots are assigned to volunteers by administrators.
* Scheduling conflict: A scheduled timeslot that the assigned volunteer cannot keep. When a scheduling conflict occurs, the assigned volunteer needs to notify administrators.

Input Domain (17b)

* Scheduled timeslot
  + Required parameters
    - Date <DayOfWeek> - the day of week for this availability period
    - StartTime <Time> - the start time for this availability period
    - EndTime <Time> - the end time for this availability period
    - Confirmed <Boolean> true if volunteer has indicated they can work this timeslot, false if the volunteer has not indicated or cannot work this timeslot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *State the product title* | | | **Product version:** *State the product version number* | |
| **Primary Actors:** Grower  Administrator | | | **ID:** 18 | **Importance Level:** High |
| **Use Case Name:** Grower Adds New Product Type | | | **Use Case Type:** Detail | |
| **Stakeholders:** | | Growers | | |
| **Goal:** | | A Grower shall be able to add a new product type to the system. | | |
| **Quality requirements:** | | The product type must contain 5 to 15 characters and cannot contain special characters. | | |
| **Pre-Conditions:** | | The Grower has logged into the system. | | |
| **Post-Conditions:** | | The system will be updated to include the new product type. | | |
| **Relationships:** | |  | | |
|  | **Include:** | Use Case 23: Administrator Approves Product Type | | |
| **Extend:** | None | | |
| **Generalization:** | None | | |
| **Trigger:** | | **Trigger Type:** External | | |
| 1. A Grower issues an “add new product type” command | | |
| **Normal Flow of Events:** | | 1. The system places the new product type in a queue for administrator approval 2. The administrator approves the new product type 3. The new product type is added to the system | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** 3a. The administrator disproves the new product type  3a1. The system notifies the Grower that the new product type request has been denied. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *State the product title* | | | **Product version:** *State the product version number* | |
| **Primary Actors:** Grower | | | **ID:** 19 | **Importance Level:** High/Medium/Low High – must be implemented Medium – must be included in this version of the software Low – may be deferred until next version of the product |
| **Use Case Name:** Grower creates product invoice | | | **Use Case Type:**  Detail | |
| **Stakeholders:** | | Grower | | |
| **Goal:** | | The Grower shall be able to create a product invoice. | | |
| **Quality requirements:** | | None | | |
| **Pre-Conditions:** | | The Grower is logged into the system. | | |
| **Post-Conditions:** | | The Grower will have an invoice of products he/she has prepared to sell. | | |
| **Relationships:** | |  | | |
|  | **Include:** | None | | |
| **Extend:** | None | | |
| **Generalization:** | None | | |
| **Trigger:** | | **Trigger Type:** External | | |
| 1. The Grower issues the “create product invoice” command | | |
| **Normal Flow of Events:** | | 1. The system prompts the Grower for a product type 2. The Grower selects a product type 3. The system prompts the Grower for a product quantity 4. The Grower selects a product quantity 5. The system prompts the Grower to add more items to the invoice 6. If the Grower selects to add more items, go to step 2. Otherwise, continue to step 8. 7. The system prompts the Grower to enter a delivery date for the invoice 8. The Grower selects a delivery date 9. The system prompts the Grower to confirm the invoice 10. The Grower confirms the invoice. 11. The system submits the invoice to the data store | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** 11a. The Grower rejects the invoice  11a1. The system returns the Grower to the Grower’s main menu | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** State the product title | | | **Product version:** State the product version number | |
| **Primary Actors:** Grower | | | **ID:** 20 | **Importance Level:** High |
| **Use Case Name:** View Invoice History | | | **Use Case Type:**  Detail | |
| **Stakeholders:** | | Grower | | |
| **Goal:** | | The Grower shall be able to view his/her invoices | | |
| **Quality requirements:** | | The system will retrieve past invoices for the Grower within 5 seconds of a request. | | |
| **Pre-Conditions:** | | The Grower is logged into the system. | | |
| **Post-Conditions:** | | The system will display the Grower’s invoice history. | | |
| **Relationships:** | |  | | |
|  | **Include:** | None | | |
| **Extend:** | None | | |
| **Generalization:** | None | | |
| **Trigger:** | | **Trigger Type:** External | | |
| 1. The Grower issues a “view invoice history” command. | | |
| **Normal Flow of Events:** | | 1. The system queries the data store all invoices created by the Grower. 2. The data store returns a list of the Grower’s invoices. 3. The system displays the Grower’s invoice history. | | |
| **flows:** | | None | | |
| **Alternative/Exception flows:** 3a. The data store returns an empty list with no invoices for the Grower  3a1. The system displays a message informing the Grower that he/she has no invoices. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** State the product title | | | **Product version:** State the product version number | |
| **Primary Actors:** Grower | | | **ID:** 21 | **Importance Level:** High |
| **Use Case Name:** View Payment History | | | **Use Case Type:**  Detail | |
| **Stakeholders:** | | Grower | | |
| **Goal:** | | The Grower shall be able to view his/her payment history | | |
| **Quality requirements:** | | The system shall retrieve payment history for the Grower within 5 seconds of a request. | | |
| **Pre-Conditions:** | | The Grower is logged into the system. | | |
| **Post-Conditions:** | | The system will display the Grower’s payment history. | | |
| **Relationships:** | |  | | |
|  | **Include:** | None | | |
| **Extend:** | None | | |
| **Generalization:** | None | | |
| **Trigger:** | | **Trigger Type:** External | | |
| 1. The Grower issues a “view payment history” command. | | |
| **Normal Flow of Events:** | | 1. The system queries the data store all payments issued to the Grower. 2. The data store returns a list of the payments made to the Grower. 3. The system displays the Grower’s payment history. | | |
| **flows:** | | None | | |
| **Alternative/Exception flows:** 3a. The data store returns an empty list with no payments made to the Grower  3a1. The system displays a message informing the Grower that he/she has no payment history. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C:\Users\Jacob\Desktop\UseCase Diagram0.png** | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Administrator* | | | **ID:** *21* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Administrator Assigning Security Roles* | | | **Use Case Type:**  Overview/Detail | |
| **Stakeholders:** | | *Administrator, Customer/User* | | |
| **Goal:** | | *The administrator shall be able to Assign Security Roles to Users* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *User must be logged on as an Administrator*  *Administrator must be on “Administrative Settings” page* | | |
| **Post-Conditions:** | | *The selected User’s will have specified security roles assigned* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *Administrator issues the “Assign Security Roles” command* | | |
| **Normal Flow of Events:** | | 1. *Administrator selects user* 2. *System prompts users for security roles.* 3. *Administrator assigns security role* 4. *Administrator saves security role.* | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:**  **4a. Adminstrator issues “Cancel” command.**  **4b. System returns to** | | | | |
| **UI: To be added** | | | | |
| **Input Specifications:** | | | | |
| **Functional Specifications:** | | | | |
| **Out-Put Specifications:**  Message = “Information Successfully Saved” | | | | |
| **Behavioral Specifications:** | | | | |
| **Data Storage:** Database | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *C:\Users\Jacob\Desktop\UseCase Diagram 22a.jpg* | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version: 1.0** | |
| **Primary Actors:** *Administrator* | | | **ID:** *22a* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Administrator views Volunteer Schedules* | | | **Use Case Type:**  Overview/Detail | |
| **Stakeholders:** | | *Administrator, Volunteer* | | |
| **Goal:** | | *View the Schedule of Volunteers* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *User must be logged in as an Administrator*  *Administrator must be on “Schedule” page* | | |
| **Post-Conditions:** | | *Volunteer Schedules have been viewed* | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *Administrator issues “View Schedule” command* | | |
| **Normal Flow of Events:** | | *Administrator navigates to “Schedule” page*  *Administrator views Confirmed and Unconfirmed Schedules*  *Administrator exits “Schedule” page* | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** *None* | | | | |
| **Input Specifications**  **Date Scheduled:** <month><digit>/<day><digit>/<year>  **Volunteer Name:** <Last Name><space><First Name>  **Time Available:** <hour><digit>”:”<minute><digit>  **Hour:** 0 | 1  **Minute:** 0 | 1 | 2 | 3 | 4 | 5  **First Name:** 1(Alpha-Numeric Characters)25  **Last Name:** 1(Alpha-Numeric Characters)25  **Month:** 0 | 1  **Day:** 0 | 1 | 2 | 3  **Year:** 4(digit)\*  **Digit:** 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | |
| **Behavioral Specifications:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *C:\Users\Jacob\Desktop\UseCase Diagram22b.png* | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Adminstrator* | | | **ID:** *22b* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Administrator views Volunteer availability* | | | **Use Case Type:**  Overview/Detail | |
| **Stakeholders:** | | *Administrator, Volunteer* | | |
| **Goal:** | | *Administrator should be able to view Volunteer’s availability* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *User must be logged in as an Administrator*  *Administrator must be on “View Volunteer Availability” page* | | |
| **Post-Conditions:** | | *Administrator should have viewed Volunteer’s availability* | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *Administrator issues “View Volunteer Availability” command* | | |
| **Normal Flow of Events:** | | 1. *Administrator issues “View Volunteer Availability” command* 2. *Administrator views volunteers’ availability* 3. *Administrator exits the page* | | |
| **Sub-flows:** | |  | | |
| **Alternative/Exception flows:** | | | | |
| **Input Specifications:**  **Date(s) Available:** <month><digit>/<day><digit>/<year>  **Volunteer Name:** <Last Name><space><First Name>  **First Name:** 1(Alpha-Numeric Characters)25  **Last Name:** 1(Alpha-Numeric Characters)25  **Month:** 0 | 1  **Day:** 0 | 1 | 2 | 3  **Year:** 4(digit)\*  **Digit:** 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | |
| **Out-put Specifications:** Same as Input | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *C:\Users\Jacob\Desktop\UseCase Diagram 22c.png* | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Administrator* | | | **ID:** *22c* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Administrator schedules Volunteers* | | | **Use Case Type:**  Overview/Detail | |
| **Stakeholders:** | | *Administrator, Volunteer* | | |
| **Goal:** | | *Volunteers should be scheduled to work* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *User must be logged in as an Administrator*  *Administrator must be on “Schedule” page* | | |
| **Post-Conditions:** | | *None* | | |
| **Relationships:** | |  | | |
|  | **Include:** |  | | |
| **Extend:** |  | | |
| **Generalization:** |  | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *Administrator issues the “Edit Schedule” command* | | |
| **Normal Flow of Events:** | | 1. *Administrator issues the “Edit Schedule” command* 2. *Administrator inputs Volunteer name, date(s) scheduled, and time(s) scheduled* 3. *Administrator issues the “Save” command* 4. *Updated Schedule is saved* 5. *Administrator exits the “Schedule” page* | | |
| **Alternative/Exception flows:** *1. Administrator issues the “Cancel” command*  *2. Changes are lost*  *3. Administrator exits the page* | | | | |
| **Input Specifications:**  **Date Scheduled:** <month><digit>/<day><digit>/<year>  **Time Scheduled:** <hour><digit>”:”<minute><digit>  **Volunteer Name:** <Last Name><space><First Name>  **First Name:** 1(Alpha-Numeric Characters)25  **Last Name:** 1(Alpha-Numeric Characters)25  **Hour:** 0 | 1  **Minute:** 0 | 1 | 2 | 3 | 4 | 5  **Month:** 0 | 1  **Day:** 0 | 1 | 2 | 3  **Year:** 4(digit)\*  **Digit:** 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | |
| **Behavioral Specifications:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Administrator* | | | **ID:** *23* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Product Confirmation* | | | **Use Case Type:**  Overview/Detail *Administrator confirms product type.* | |
| **Stakeholders:** | | *Administrator, User/Customer* | | |
| **Goal:** | | *To allow Administrators to view and confirm varying product types* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *Administrator must be logged in.*  *At least one product type that needs to be confirmed* | | |
| **Post-Conditions:** | | *New product types confirmed* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *Administrator clicks “Confirm Product Type”* | | |
| **Normal Flow of Events:** | | *List the normal flow of the use case.* | | |
| **Sub-flows:** | | 1. Administrator views new product types 2. Administrator selects specified product types 3. Administrator clicks “Confirm” | | |
| **Alternative/Exception flows:** *Error Message* | | | | |
| **UI Design:** | | | | |
| **Input Specifications:**  **Product Type:** 1(Alpha-Numeric Characters)50  **Grower Name:** 1(Alpha-Numeric Characters)25 | | | | |
| **Functional Specifications:**  C:\Users\Jacob\Desktop\Activity Diagram 23.png | | | | |
| **Out-Put Specifications:**  **Saved Successfully Message =** “Information Successfully Saved”  **Error Message =** “Error: Information not Saved” | | | | |
| **Behavioral Specifications:** | | | | |
| **Data Storage Requirements:** Database | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Insert the use-case diagram here* | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Administrator* | | | **ID:** *22* | **Importance Level:** High/Medium/Low *Medium* |
| **Use Case Name:** *Updating Blog* | | | **Use Case Type:**  Overview/Detail *Administrator updates Blog* | |
| **Stakeholders:** | | *Administrator, Customer/User* | | |
| **Goal:** | | *To Update the Blog* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *Administrator must be logged in* | | |
| **Post-Conditions:** | | *Blog is updated* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *Administrator clicks “Update”* | | |
| **Normal Flow of Events:** | |  | | |
| **Sub-flows:** | | 1. *Administrator clicks “Update”* 2. *New Editable Blog Post Added* 3. *Administrator updates text* 4. *Administrator clicks “Save”* | | |
| **Alternative/Exception flows:** *List the steps that may happen but are not considered part of the normal flow.* | | | | |
| **Input Specifications:**  **Blog Post:** 1(Alpha-Numeric Characters)500  **Date Posted:** <month><digit>/<day><digit>/<year>  **Authors Name:** 1(Alpha-Numeric Characters)25  **Month:** 0 | 1  **Day:** 0 | 1 | 2 | 3  **Year:** 4(digit)\*  **Digit:** 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | |
| **Behavioral Specifications:**  While Administrator clicks “Update Blog”  New Blog Post Displayed with editable text region  If Administrator clicks “Confirm”  Blog Post Saved  Else If Administrator clicks “Cancel”  Blog post Deleted  End While | | | | |
| **Out-Put Specifications:** Same as Input | | | | |
| **Behavioral Specifications:** | | | | |
| **Data Storage Requirements:** Blog posts saved to Database. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *25* | **Importance Level:** High/Medium/Low *High* |
| **Use Case Name:** *Customer Update Account Info* | | | **Use Case Type:**  Overview/Detail *Overview – very high level view of the requirement (lacks detail)* | |
| **Stakeholders:** | | *Customer* | | |
| **Goal:** | | *Allow a Customer to update their account information* | | |
| **Quality requirements:** | | *5. The system shall commit user updates to his/her account information within 3 seconds.* | | |
| **Pre-Conditions:** | | *The Customer must be logged in.* | | |
| **Post-Conditions:** | | *The Customer’s account information must be updated.* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *A Customer selects the Update Account Info option* | | |
| **Normal Flow of Events:** | | 1. *The Customer changes their First Name, Last Name, Username, Address, Phone Number, or Email Address* 2. *The System validates the changed field* 3. *The System updates persistent storage with the changes* 4. *The System displays the message “Changes applied successfully”* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *2.a. The changed field(s) is/are invalid.*  *2.a.1. The System displays an error indicating the invalid fields.* | | | | |
| **Functional Specification:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Product title:** *Jonesborough Farmer’s Market Sales System and Website* | | | **Product version:** *1.0* | |
| **Primary Actors:** *Customer* | | | **ID:** *26* | **Importance Level:** High/Medium/Low *High* |
| **Use Case Name:** *Customer Delete Account* | | | **Use Case Type:**  Overview/Detail *Overview – very high level view of the requirement (lacks detail)* | |
| **Stakeholders:** | | *Customer* | | |
| **Goal:** | | *Allow a Customer to delete their account* | | |
| **Quality requirements:** | | *None* | | |
| **Pre-Conditions:** | | *The Customer must be logged in* | | |
| **Post-Conditions:** | | *The Customer’s account must be removed.* | | |
| **Relationships:** | |  | | |
|  | **Include:** | *None* | | |
| **Extend:** | *None* | | |
| **Generalization:** | *None* | | |
| **Trigger:** | | **Trigger Type:** External/Temporal *External* | | |
| *The Customer selects the Delete Account option* | | |
| **Normal Flow of Events:** | | 1. *The Customer enters their password* 2. *The System validates the password* 3. *The System updates persistent storage* | | |
| **Sub-flows:** | | *None* | | |
| **Alternative/Exception flows:** *2.a. Invalid password*  *2.a.1. The System displays the error “Invalid password”*  *2.a.2. The System does not delete the Customer’s account* | | | | |