Milestone 2

Team Sriram
(Susi Cisneros, Eric Henderson, Taylor Purviance and Richard Thai)

7 October 2011

Contents

Changes (based off Git commits)

Date Time	Description

1 Executive Summary

This milestone, the second of a series, documents the context for a software project proposed by the client, Tim Ekl. This document also keeps track of the use cases involved with the final product which include adding an item, basic searching, advanced searching, sorting search results, viewing item details, editing item details, and generating inventory reports. All of the use cases satisfy a set of features and keep track of actors, basic/alternative flow of events, pre-conditions, post-conditions, extension points, and special requirements. In addition, this document storyboards the final product as well as keep track of the context/data flow diagrams.

2 Introduction

TODO: [Richard]

3 User Environment

- The client uses Chrome whenever possible and prefers that development support Chrome and Firefox browsers.
- The final product should operate on a Linux server with standard programming languages, programming frameworks, and Apache. Additional packages can be installed if necessary.

4 User Needs

ID	Need	Priority
N0	Search for parts based off their attributes	Primary
N1	Identify items via bar codes	Primary
N2	Keep track of the data associated with an asset	Primary
N3	Organize search results	Primary
N4	Insert objects in the system at any point; do not freeze the database	Secondary
N5	Modify objects; including adding notes to the objects	Secondary
N6	Access from a second physical location	Optional
N7	View most-recently acquired asset(s)	Optional
N8	View a summary of inventory data	Optional

5 Features

5.1 Feature Listing

ID	Feature	Priority	Effort	Risk	Stability	Target Release	Assigned To
F0	Online UI	Critical	High	High	Low	1.0	Eric
F1	Add assets to the inventory	Critical	Low	High	Low	1.0	Richard
F2	Modify assets in the inventory	Critical	Low	High	Low	1.0	Taylor
F3	The system keeps track of attributes based on category	Critical	Low	High	Low	1.0	Susi
F4	Use a UPC-A barcode as the unique identifier for each asset	Critical	Low	Medium	Low	1.0	Eric
F5	Provide an updated list of recently-added assets	Useful	Medium	Low	Medium	1.5	Richard
F6	Generate reports of asset inventory	Useful	High	Low	Medium	2.0	Taylor
F7	Sort search results based off of barcode, title, and modified/created timestamp	Useful	High	Low	Low	2.0	Susi
F8	Basic search for items based on name or UPC	Critical	High	High	Low	1.0	Eric
F9	Advanced search for items based on all fields related to the item and its category	Critical	High	High	Low	1.0	Richard
F10	Basic and Advanced searches allow the user to include wildcards in the query	Critical	High	High	Low	1.0	Taylor
F11	Basic and Advanced searches will search first by exact/wildcard match, then by fuzzy match	Critical	High	High	Low	1.0	Susi
F12	REST API	Critical	High	High	Low	1.0	Eric

5.2 Feature-to-Need Correspondence

	N0	N1	N2	N3	N4	N5	N6	N7	N8
F0							X		
F1			X		X				
F2			X			X			
F3			X		X	X			
F4	X	X	X						
F5								X	
F6									X
F7	X	X		X					
F8	X	X	X	X					
F9	X	X	X	X					
F10	X	X	X	X					
F11	X	X	X	X					
F12							X		

6 Client Background

Tim Ekl is a Rose-Hulman graduate student who possesses a significant amount of computer hardware. He plans on using this system to be able to quickly and easily locate the equipment he wants to use. Tim is an experienced developer and plans on maintaining the system after it is finished.

7 Current System

The client does not have a software solution in place. Currently, Tim has a primitive categorization system in place which involves labeling boxes and then trying to deduce the location of a desired component. The current system poses a few issues such as not always allowing him to find his items, i.e. there have been instances where an item was found after capital was spent to replace it.

8 Project Background

TODO: [Richard]

9 Use Cases

Use case syntax
Each use case is divided into 8 sections:

- A Basic Description section which gives an overview of what functionality the use case demonstrates.
- An Actors section to describe who or what interacts with the system in the use case.
- A Pre-conditions section which contains the assumptions made, especially those pertaining to the state of the system, prior to the start of the use case.
- A Basic Flow of Events section which details the order of events done by the actor(s) and the system under standard conditions. If a particular step in the basic flow has the possibility of failing to occur successfully, one or more alternative flow listings are presented in brackets following the possibility of failure. Each alternative flow listing matches to an alternative flow in the Alternative Flow section. If a basic flow step fails, the alternative flow that is listed with it is followed as the next step.

- An Alternative Flow of Events section which holds each alternative flow listed in the basic flow. Each alternative flow has a unique identifier used to reference it, a short name describing the condition that would cause the alternative flow to be followed, and the order of actions performed by the actor(s) and the system under the flows conditions. If an action has the possibility of failing, alternatives to that flow are presented just as they are in the basic flow with a reference to the alternative flow to take.
- A Post-conditions section which contains guarantees on the result of the use case.
- An Extension Points section which contains references to other use cases which the completion of the current use case might lead into.

9.1 Add an Item

Brief Description This use case shows the procedure for adding an item to the inventory.

Actors

- User

Pre-conditions

- The current page must have a link to the "Add Item" page.

Basic Flow of Events

1. The user clicks the "Add Item" link. [A0]
2. The user fills in the required fields. [A1]
3. The system dynamically adds the optional attribute fields for the category chosen.
4. The user fills in the appropriate optional attributes and "Notes" fields.
5. The user clicks the "Add Item" button. [A2]
6. The system adds the item to the inventory database.

Alternative Flows of Events

Alternative Flow A0: Server is Down

1. The user is notified that the server could not be reached.

Alternative Flow A1: Required Fields Ommitted

1. The user does not fill in the required title or UPC fields.
2. Steps 3, 4, and 5 from the basic flow.
3. The server notifies the user that the required fields were not completed.

Alternative Flow A2: UPC not Unique

1. The system notifies the user that the item cannot be added because the UPC is not unique.

Post-conditions

- The user is brought to the home page.
- The added item is displayed under the "Recently Modified" list.

Extension Points None

9.2 Basic Search for an Item

Brief Description This use case shows how a basic search from any page would proceed. This case will encompass the progression from the user entering the search term to the display of the search result(s).

Actors

- User

Pre-conditions

- The current page must have a basic search box.

Basic Flow of Events

1. The user types a query in the search box.
2. The user clicks the search button.
3. The browser sends a request to the server. [A0]
4. The server queries the database for items entered into the system that match the query. [A1]
5. The server responds with the formatted and sorted results of the database query. [A2]
6. The user is directed to the Advanced Search page with the results of the basic search displayed at the bottom.

Alternative Flows of Events

Alternative Flow A0: Empty Query

1. The user remains on the current page.

Alternative Flow A1: Server is Down

1. The user is notified that the server could not be reached.

Alternative Flow A2: No Matches

1. The server responds with the message "No results."
2. The results page displays the message returned by the server.

Post-conditions

- The results page will display the response from the server or that the server was unreachable.
- The search box will preserve the search term that was originally entered.
- The results will be prioritized and sorted according to exact matching, then fuzzy matching.
- The results will be matched using wildcards if they are present in the query. This will replace the exact matching, and fuzzy matching will then be done on the query with the wildcard characters removed.

Extension Points

- 5.4 Search Result Sorting
- 5.5 View Details of an Item

9.3 Advanced Search for an Item

Brief Description This use case shows how a search from the advanced search page would proceed. This case will encompass the progression from the user entering the search term to the display of the search result(s).

Actors

- User

Pre-conditions

- The current page must be the advanced search page.

Basic Flow of Events

1. The user chooses a category from a dropdown menu. [A0]
2. The category selection is sent to the server.
3. The server responds with the attribute fields related to the category. [A1]
4. The category's attribute fields are dynamically added to the page.
5. The user optionally fills in the attribute, name, and UPC fields appropriately.
6. The user clicks the search button.
7. The browser sends a request to the server.
8. The server queries the database for items entered into the system that match the query. [A1]
9. The server responds with the formatted and sorted results of the database query. [A3]
10. The list of candidates for the search parameter(s) provided is dynamically added to the bottom part of the page.

Alternative Flow of Events

Alternative Flow A0: No Category

1. The user fills in the name and/or UPC.
2. Return to basic flow step 6.

Alternative Flow A1: Server is Down

1. The user is notified that the server could not be reached.

Alternative Flow A2: Empty Query

1. The user remains on the current page.

Alternative Flow A3: No Matches

1. The server responds with the message "No results."
2. The results page displays the message returned by the server.

Post-conditions

- The results page will display the results of the search.
- The search fields will preserve the search parameter(s) that were originally entered.
- If appropriate, the results will be prioritized and sorted according to exact matching, then fuzzy matching.
- The results will be matched using wildcards if they are present in the query. This will replace the exact matching, and fuzzy matching will then be done on the query with the wildcard characters removed.
- If completed, the category field will only use exact matching.

Extension Points

- 5.4 Search Result Sorting
- 5.5 View Details of an Item

9.4 Search Result Sorting

Brief Description This use case shows how changing the sorting of the items on the search results page would proceed. This case will encompass the progression from the user clicking the sort type link to the display of the search result(s) in the new order.

Actors

- User

Pre-conditions

- The current page must be the search results page.

Basic Flow of Events

1. The user clicks on a sort type link (options are Name, UPC, Recently Created, and Recently Modified).
2. The browser sends a request to the server.
3. The server queries the database for items entered into the system that match the query. [A0]
4. The server responds with the formatted and sorted results of the database query. [A1]
5. The screen displays to the user a list of candidates for the search parameter(s) provided.

Alternative Flow of Events

Alternative Flow A0: Server is Down

1. The user is notified that the server could not be reached.

Alternative Flow A1: No Matches

1. The server responds with the message "No results."
2. The results page displays the message returned by the server.

Post-conditions

- The results page will display the results of the search.
- The search field(s) will preserve the search parameter(s) that were originally entered.
- The search fields will be sorted by match method first (wildcard, exact, or fuzzy), then by the selected sort type.

Extension Points

- 5.5 View Details of an Item

9.5 View Details of an Item

Brief Description This use case shows how a user would select an item from a list to view more detailed information about it.

Actors

- User

Pre-conditions

- The current page must have a link to the item which the user would like to view, as might happen in search results list or the recently modified items list.

Basic Flow of Events

1. The user clicks on an item (in search results, the recent changes list, or any other place with items listed).
2. The browser queries the server for the item using the UPC code to specify the item.
3. The server responds with all the data stored about the item. [A0]
4. The browser takes the user to a new page with all the data about the item that the server sent. It is displayed in elements labeled with Category:, Size:, etc. [A1]

Alternative Flow of Events

Alternative Flow A0: Server is Down

1. The user is notified that the server could not be reached.

Alternative Flow A1: Item not Found

1. The page displays a message saying that the item could not be found.

Post-conditions

- The page will display all data about the item that the system currently has logged for the item

Extension Points

- 5.6 Edit Details of an Item

9.6 Edit Details of an Item

Brief Description This use case shows how a user would modify an item that already exists in the server and save the edited/new field information for future viewing.

Actors

- User

Pre-conditions

- The current page must be an item's detailed description page.

Basic Flow of Events

1. The user clicks on the value of a field.
2. The page replaces the element with a text field containing the text that was in the element.
3. The user makes the desired changes to the value.
4. The user clicks outside of the field.
5. The browser sends a request to the server asking the value to be changed in the database.
6. The server changes the value in the database. [A0, A1]
7. The server responds that the value was changed.
8. The page changes the text field back into a div element that contains the updated value of the field.

Alternative Flow of Events

Alternative Flow A0: Server is Down

1. The user is notified that the server could not be reached.

Alternative Flow A1: Change Unsuccessful

1. The server responds that the value could not be changed.
2. The page notifies the user that the value could not be changed.
3. The user presses "Ok" in the message dialog.
4. The text field is focused again.

Post-conditions

- The data on the page will represent the updated state of the data in the system.

Extension Points None

9.7 Generate Inventory Report

Brief Description This use case shows the procedure for generating reports of the item inventory.

Actors

- User

Pre-conditions

- The current page must have a link to the "Generate Report" page.

Basic Flow of Events

1. The user clicks the "Generate Report" link. [A0]
2. The user is brought to the "Generate Report" page.
3. The user chooses what type of report to generate.
4. The user chooses appropriate report fields for the respective option.
5. The system generates the report accordingly.

Alternative Flows of Events

Alternative Flow A0: Server is Down

1. The user is notified that the server could not be reached.

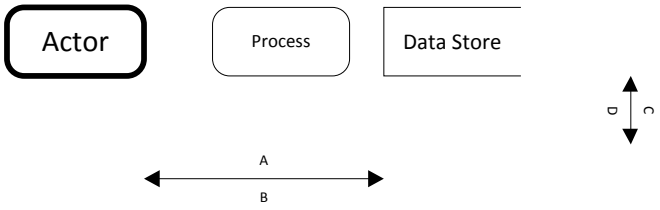
Post-conditions

- After basic step 5, the user is displayed the report which is generated as specified.

Extension Points None

10 Data Flow Diagrams

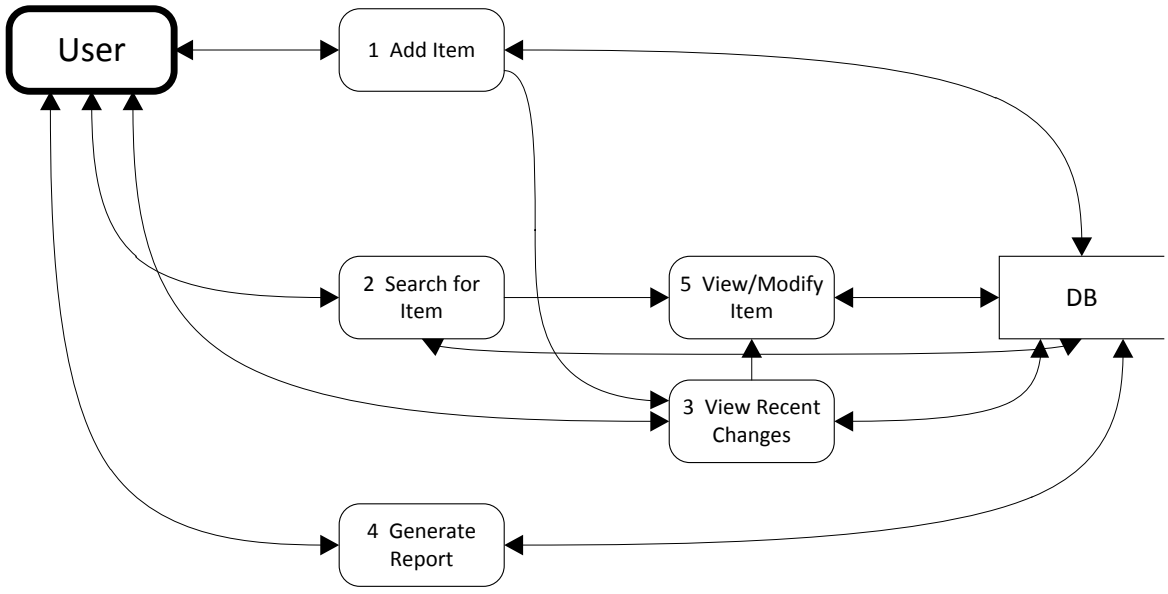
DFD Legend



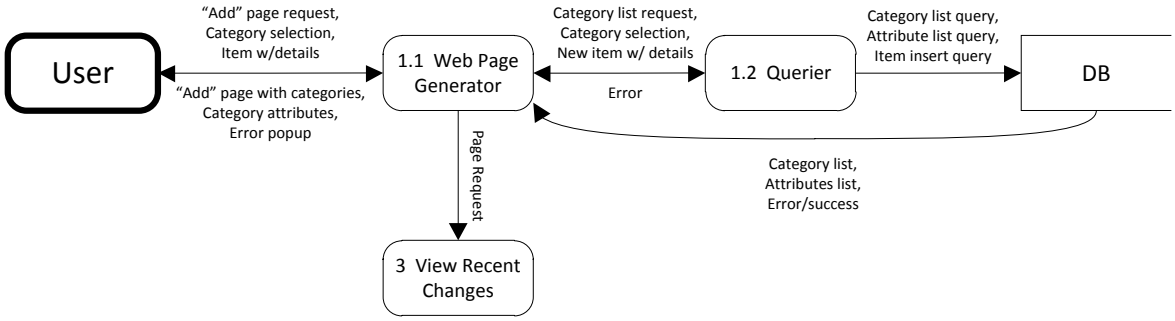
- Actor = some user or another system that interacts with the system.
- Process = name of some action that will be taken on the data
- Data Store = something like a file or database that stores data

With the bidirectional arrows the data listed on above the arrow (like A) flows left-to-right, data listed below the arrow (like B) flows right-to-left, data listed to the right of the arrow (like C) flows top-to-bottom, and data listed to the left of the arrow (like D) flows bottom-to-top.

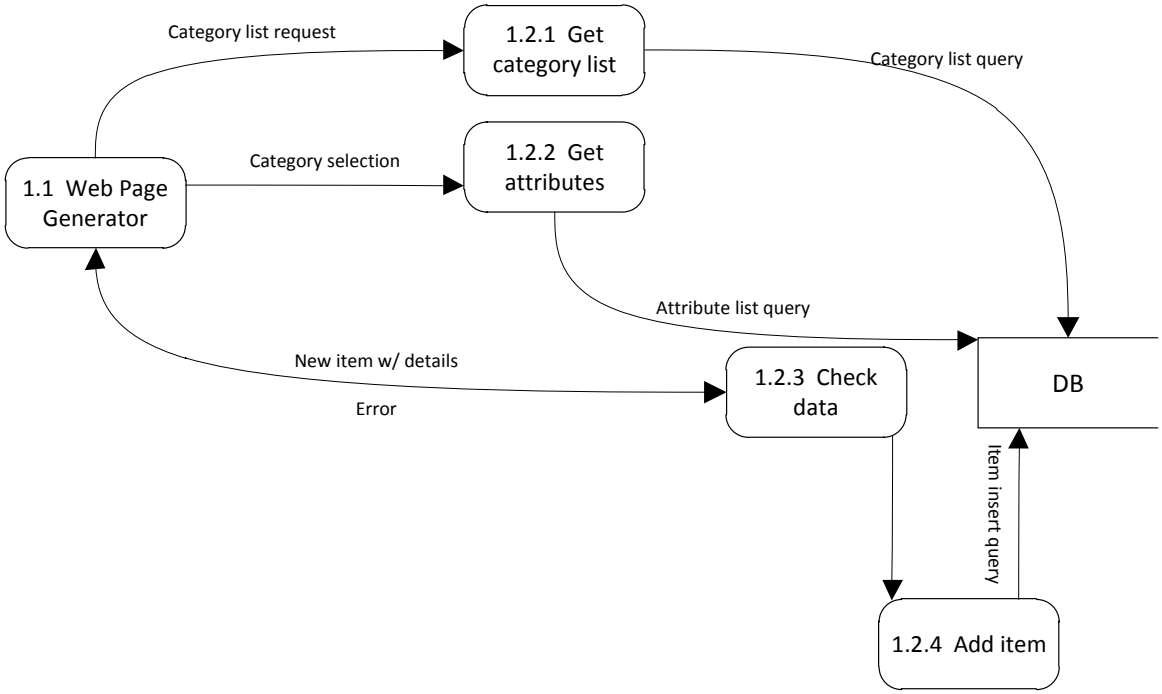
Context/Level 0



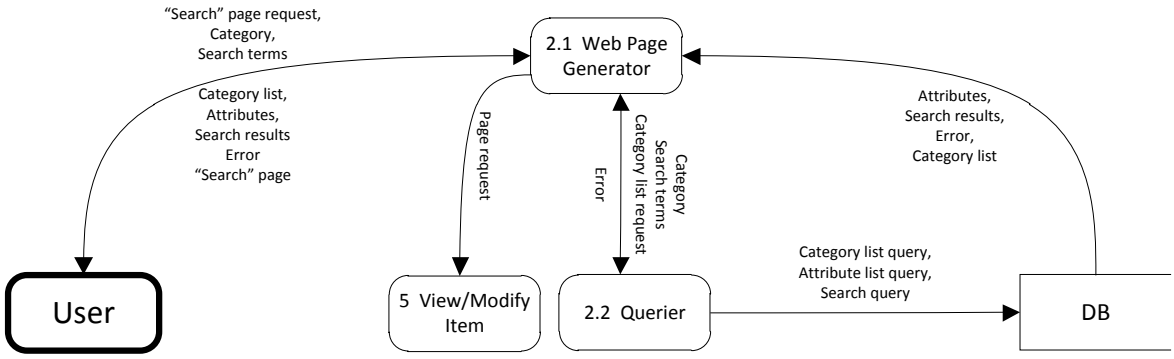
Level 1: Add Item



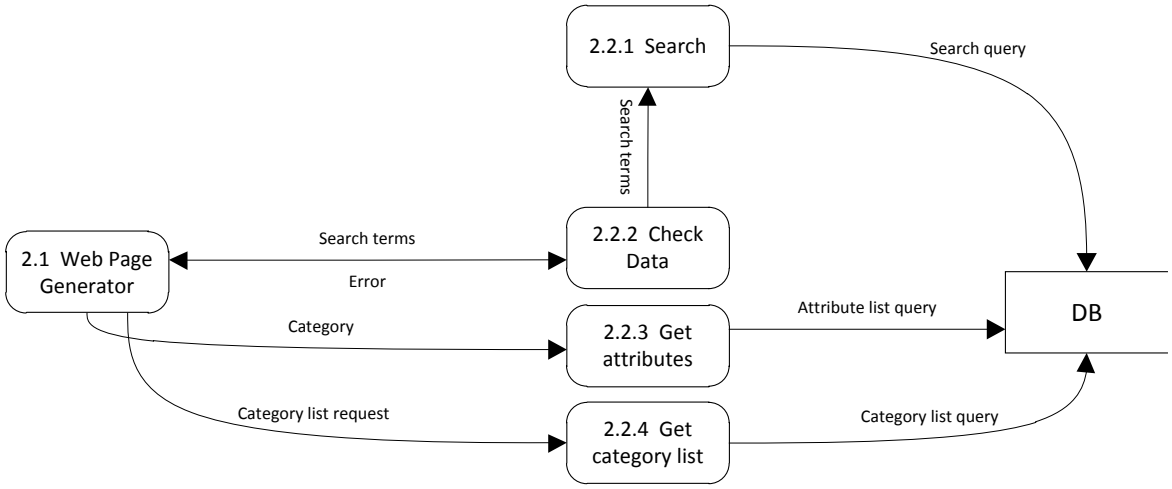
Level 2: Add Item–Querier



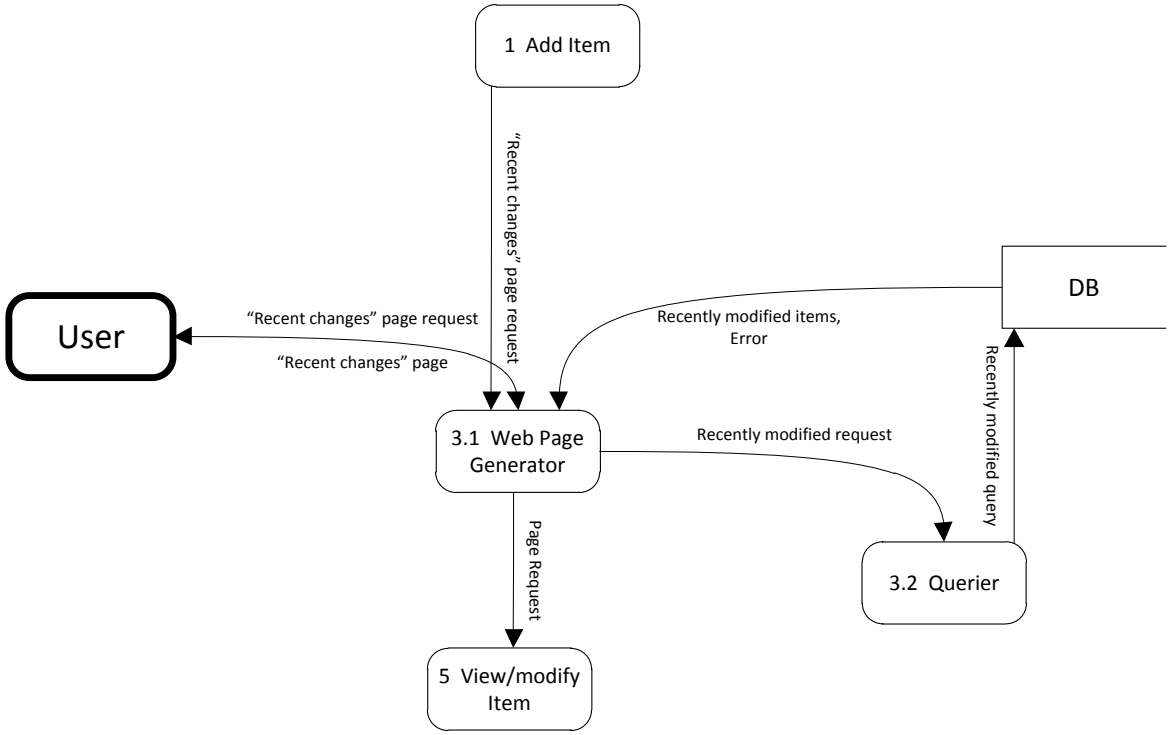
Level 1: Search for Item



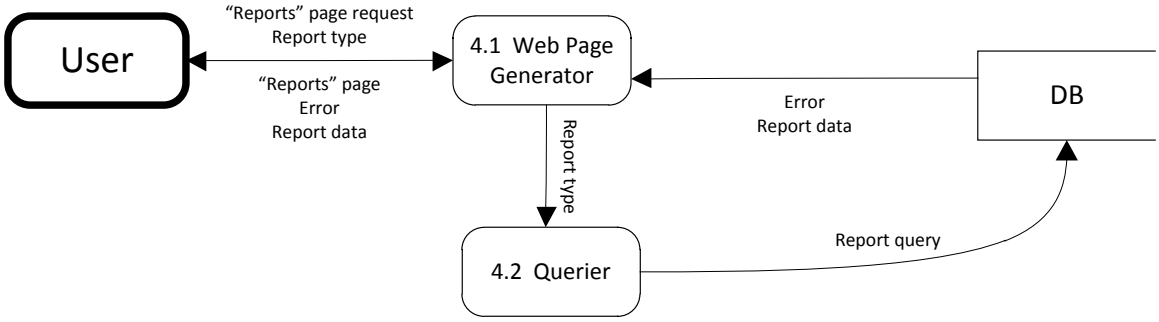
Level 2: Search for Item–Querier



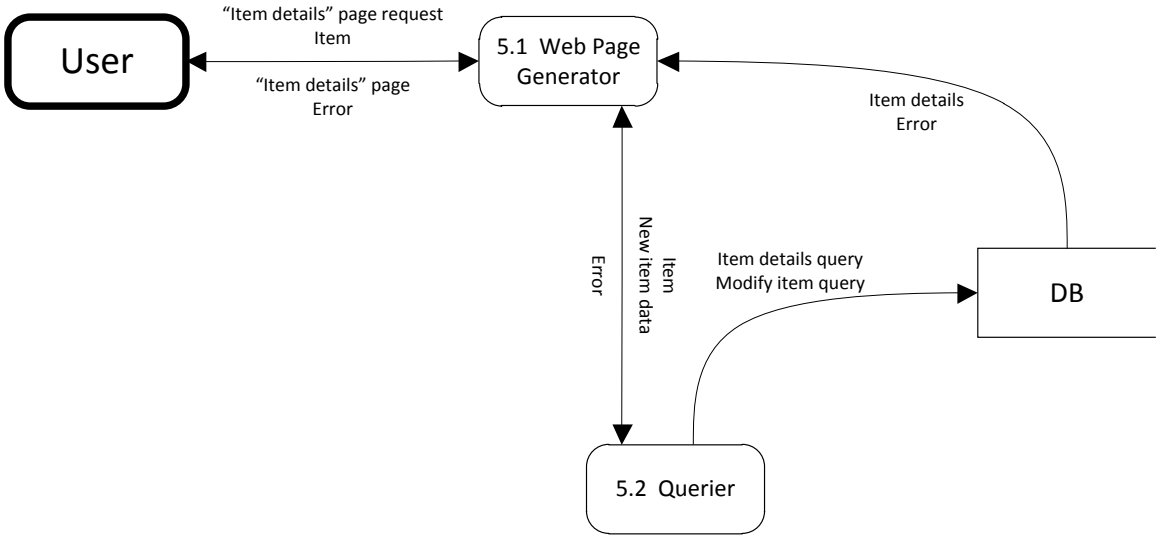
Level 1: View Recent Changes



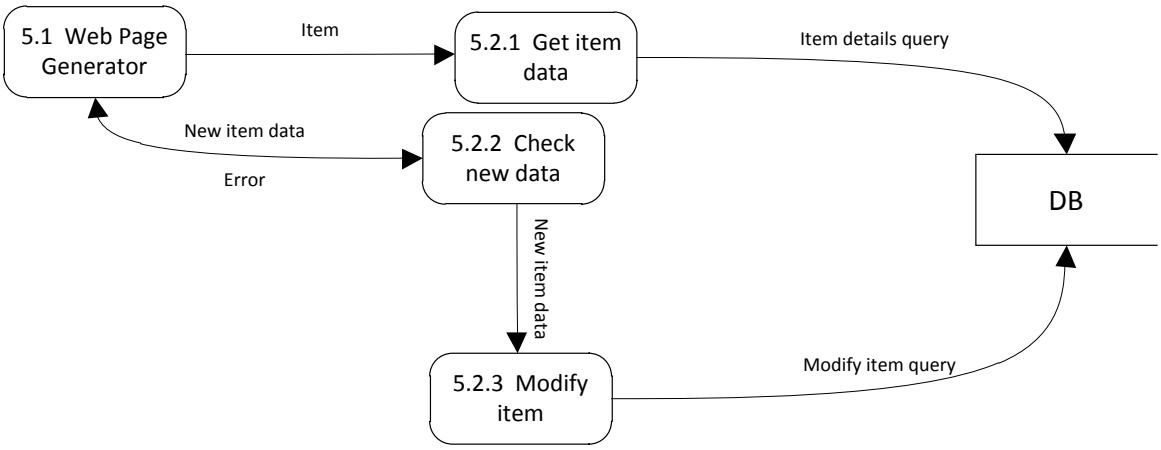
Level 1: Generate Report



Level 1: View/Modify Item



Level 2: View/Modify Item–Querier



11 Index and Glossary

12 References

(1) Leffingwell, Dean, and Don Widrig. *Managing Software Requirements: a Use Case Approach.* Addison-Wesley, Boston, 2nd Edition, 2003.