

Derivation and Evaluation of Business Blueprints

Zi Zhou Wang (zw3948)

EE 382C Software Architectures
Dr. Barber

March 23, 2018

Table of Contents

Table of Contents	2
Part 1: Prioritize Stakeholder Qualities/Constraints and Associate them with Quality Categories	3
Table 1: Prioritized Stakeholder Needs	3
Part 2: Derive a Business Blueprint	5
Figure 1: UML-Based depiction of BB	5
Table 2: Allocation of functions and data to components	5
Table 3: Component-to-Component I/O Dependencies	7
Table 4: Component-to-External I/O Dependencies	9
Table 5: I/O Dependencies within components	11
Table 6: Derivation Plan	12
Discussion of Stakeholder needs	14
Table 7: Potential conflicts and impact on derivation plan	14
Description of the bootstrap given derivation plan	15
Part 3: Evaluate Business Blueprint Structure	16
Table 8: Coupling and Cohesion Metrics	16
Table 9: Size and Complexity Metrics	16
Support for Applied Heuristic	17

Part 1: Prioritize Stakeholder Qualities/Constraints and Associate them with Quality Categories

Table 1: Prioritized Stakeholder Needs

Priority	Need/Quality	Classification	Priority Justification
1	The system should be robust and allow customers to browse product catalog, add items to cart, and successfully complete transactions	Reliability	The main purpose of the system is to provide a convenient and reliable shopping experience to users.
2	The system should protect sensitive store and customer data from hacking while also maintaining a secure payment system for the brick and mortar stores	Security	Loss of reputations of not only the system but also its associated stores can have catastrophic business consequences if the security of the system is compromised.
3	The system should be able to recover from database corruption or deletion with minimal loss of productivity	Recoverability	In the case of an unforeseen event such as the loss of power, it would be unacceptable for the system to be severely compromised for an extended period of time.
4	The system should be hosted on platforms familiar to users, while maintaining a desired workflow that should be intuitive	Usability	User experience is essential for the success of any product. As such, the intuitiveness or usability of the system is an essential priority.
5	The system should always be up and available, except for rare, appropriate circumstances	Availability	Any and all downtime of the system experiences by customers compromises the reputation and perception of the system, damaging business sales.
6	The system should offer low latency for all user requests over	Response time	Response time is a key measure of user experience.

	common network speeds		The loss of which can be extremely damaging.
7	The system should be able to be maintained easily and quickly, minimizing downtime	Maintainability	Changes to the system needs to be easily and quickly administered as delays will be noticed by customers and damage long term performance.
8	The system should be able to be evaluated and critiqued, by both customers and employees	Evaluability	After the deployment of the system, updates and changes should be based off of feedback and critiques given by stakeholders.
9	The system should be easily extended to support future brick and mortar stores, a larger product catalog, and a larger user base in the case of widespread adoption	Scalability	As the system grows, its design needs to support future augmentations that allow it to expand and incorporate additional stakeholders.
10	The system should be able to adapt to specific themes and cultures of specific companies or holidays and seasons	Adaptability	Companies or stores who employ the system should be able to differentiate themselves from others using the same system through themes and skins.

Part 2: Derive a Business Blueprint

Figure 1: UML-Based depiction of BB

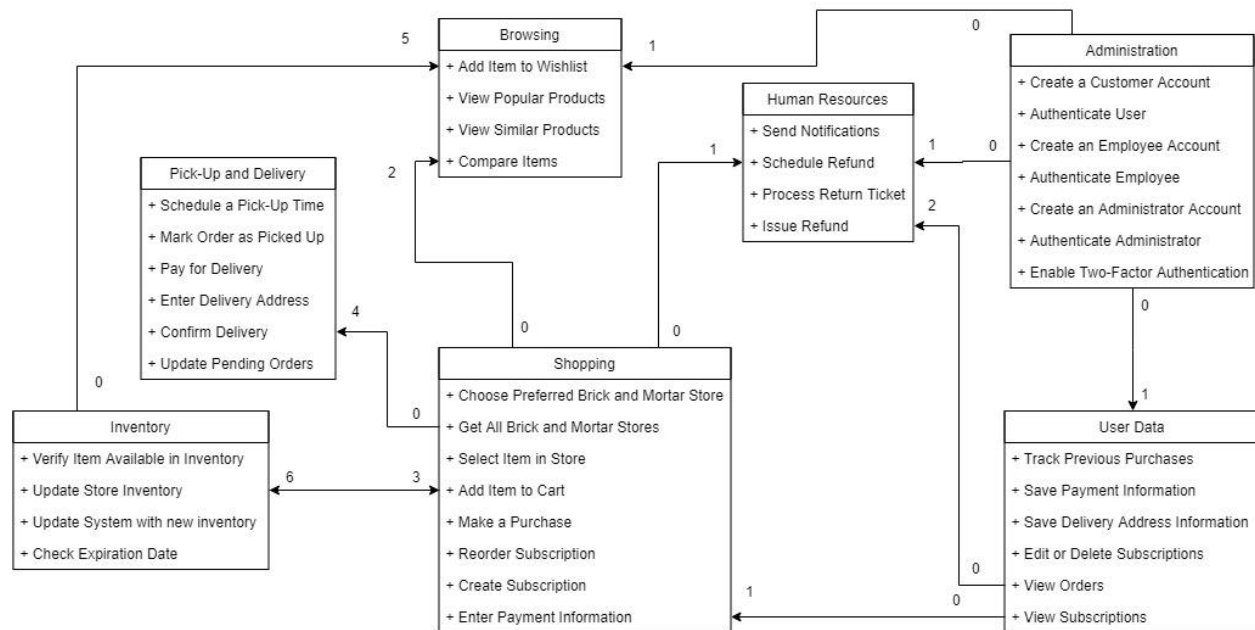


Table 2: Allocation of functions and data to components

Component	Functions and Data
Administration	<p>Functions:</p> <ul style="list-style-type: none"> - Create a Customer Account - Authenticate User - Create an Employee Account - Authenticate Employee - Create an Administrator Account - Authenticate Administrator - Enable Two-Factor Authentication <p>Data:</p> <ul style="list-style-type: none"> - Employee ID - User ID - Administrator ID - Passwords - Accounts - Login Sessions

	<ul style="list-style-type: none"> - Two-Factor Authentication Enable - User Account - Employee Account - Administrator Account
Shopping	<p>Functions</p> <ul style="list-style-type: none"> - Choose Preferred Brick and Mortar Store - Get All Brick and Mortar Stores - Select Item in Store - Add Item to Cart - Make a Purchase - Reorder Subscription - Create Subscription - Enter Payment Information <p>Data:</p> <ul style="list-style-type: none"> - Stores - Item Selected - Shopping Cart - Payment information - Confirmation/Order Numbers - Items successfully purchased - Subscription orders
Inventory	<p>Functions:</p> <ul style="list-style-type: none"> - Verify Item Available in Inventory - Update Store Inventory - Update System with new inventory - Check Expiration Date <p>Data:</p> <ul style="list-style-type: none"> - Item Inventory - Item View Counts - Item Purchase Counts - Expiration date of inventory items
Human Resources	<p>Functions:</p> <ul style="list-style-type: none"> - Send Notifications - Schedule Refund - Process Return Ticket - Issue Refund <p>Data:</p> <ul style="list-style-type: none"> - User contact information - Items returned for post-returning processing - Return Tickets - Repayment/Refund history
User Data	<p>Functions:</p> <ul style="list-style-type: none"> - Track Previous Purchases - Save Payment Information - Save Delivery Address Information - Edit or Delete Subscriptions

	<ul style="list-style-type: none"> - View Orders - View Subscriptions <p>Data:</p> <ul style="list-style-type: none"> - User Subscriptions - Order History - List of Orders the user has placed - Items from Past Order - User Payment information - User Delivery information - User Subscriptions - Previous
Pick-Up and Delivery	<p>Functions:</p> <ul style="list-style-type: none"> - Schedule a Pick Up Time - Mark Orders as Picked Up - Pay for Delivery - Enter Delivery Address - Confirm Delivery - Update Pending Orders <p>Data:</p> <ul style="list-style-type: none"> - Order - Pending orders - Delivery payment information - Delivery Address information - Delivery history
Browsing	<p>Functions:</p> <ul style="list-style-type: none"> - Add Item to Wishlist - View popular Products - View Similar Products - Compare Items <p>Data:</p> <ul style="list-style-type: none"> - User wishlist - Popular products - List/map or similar products - Comparison information of similar products

Table 3: Component-to-Component I/O Dependencies

Components	Functions and Data
FROM: Inventory TO: Shopping	<ul style="list-style-type: none"> - Select Item in Store requires Item Inventory from Verify Item Available in Inventory - Add Item to Cart requires Item Can be Added to Cart from Verify Item Available in Inventory - Reorder Subscription required Item Inventory retrieved

	directly from Inventory where it is allocated
FROM: Shopping TO: Inventory	<ul style="list-style-type: none"> - Verify Item Available in Inventory requires Item from Add Item to Cart - Verify Item Available in Inventory requires Store from Choose a Preferred Brick and Mortar Store - Update Store Inventory requires Items from Add Items to Cart - Update Store Inventory requires Store from Choose a Preferred Brick and Mortar Store - Update Store Inventory requires List of Items Successfully Purchased from Make a Purchase - Check Expiration Date requires Selected Item from Select Item in Store
FROM: Shopping TO: Pick-Up and Delivery	<ul style="list-style-type: none"> - Schedule a Pick Up Time requires Store from Choose a Preferred Brick and Mortar Store - Confirm Delivery requires Request Delivery from Make a Purchase - Pay for Delivery requires Payment Information from Enter Payment Information - Enter Delivery Address requires Order Number generated from Make a Purchase
FROM: Administration TO: User Data	<ul style="list-style-type: none"> - Track Previous Purchases requires User Account retrieved directly from Administration where it is allocated
FROM: Administration TO: Human Resources	<ul style="list-style-type: none"> - Send Notification requires User Account retrieved directly from Administration where it is allocated
FROM: User Data TO: Human Resources	<ul style="list-style-type: none"> - Schedule Refund requires Item from Past Order from View Orders - Issue Refund requires User Payment Info which is retrieved directly from User Data where it is allocated
FROM: Shopping TO: Human Resources	<ul style="list-style-type: none"> - Process Return Ticket requires Payment Information which is generated from Enter Payment Information
FROM: Administration TO: Browsing	<ul style="list-style-type: none"> - Add Item to Wishlist requires User Account retrieved directly from Administration where it is allocated
FROM: Shopping TO: Browsing	<ul style="list-style-type: none"> - Add Item to Wishlist requires Selected Item from Select Item in Store - Compare Items required Items which is retrieved directly from Shopping where it is allocated
FROM: Inventory TO: Browsing	<ul style="list-style-type: none"> - View Popular Products requires Item Inventory retrieved directly from Inventory where is is allocated - View Popular Products requires Item View Counts retrieved directly from Inventory where is is allocated

	<ul style="list-style-type: none"> - View Popular Products requires Item Purchase Counts retrieved directly from Inventory where it is allocated - View Similar Products requires Item inventory retrieved directly from Inventory where it is allocated - Compare Items requires Item Inventory which is retrieved directly from Inventory where it is allocated
FROM: User Data TO: Shopping	<ul style="list-style-type: none"> - Reorder Subscriptions requires User Subscriptions which is allocated in User Data

Table 4: Component-to-External I/O Dependencies

Components	Functions and Data
FROM: External TO: Administration	<ul style="list-style-type: none"> - Create Customer Account requires User ID as input from an external producer - Create Customer Account requires Password as input from an external producer - Create Customer Account requires Request to create account from an external producer - Authenticate User requires User ID as input from an external producer - Authenticate User requires Password as input from an external producer - Authenticate User requires Request to login as input from an external producer - Create Employee Account requires Employee ID as input from an external producer - Create Employee Account requires Password as input from an external producer - Create Employee Account requires Request to create account from an external producer - Authenticate Employee requires Employee ID as input from an external producer - Authenticate Employee requires Password as input from an external producer - Authenticate Employee requires Request to login as input from an external producer - Create Administrator Account requires User ID as input from an external producer - Create Administrator Account requires Password as input from an external producer - Create Administrator Account requires Request to create account from an external producer - Authenticate Administrator requires User ID as input from an external producer

	<ul style="list-style-type: none"> - Authenticate Administrator requires Password as input from an external producer - Authenticate Administrator requires Request to login as input from an external producer - Enable Two-Factor Authentication requires User Toggle as input from an external producer
FROM: External TO: Shopping	<ul style="list-style-type: none"> - Choose a Preferred Brick and Mortar Store requires Store as input from an external producer - Choose a Preferred Brick and Mortar Store requires Request to choose a Preferred Store as input from an external producer - Select Item in Store requires Request to select an item as input from an external producer - Add Item to Cart requires Request to Add Item to Cart as input from an external producer - Make a Purchase requires Confirmation as input from an external producer - Enter Payment Information requires Payment Information as input from an external producer
FROM: External TO: Inventory	<ul style="list-style-type: none"> - Verify Item Available in Inventory requires Request to Add Item to Cart as input from an external producer - Update Store Inventory requires Made Successful Purchase as input from an external producer - Update System with new inventory requires Items as input from an external producer - Update System with new inventory requires Adds Items to inventory as input from an external producer
FROM: External TO: Pick-Up and Delivery	<ul style="list-style-type: none"> - Schedule a Pick Up Time requires Time as input from an external producer - Mark Order as Picked Up requires Order as input from an external producer - Mark Order as Picked Up requires Request to Mark Order as Picked Up as input from an external producer - Confirm Delivery requires Request Delivery as input from an external producer - Pay for Delivery requires Payment Information as input from an external producer - Enter Delivery Address requires Delivery Address as input from an external producer - Get All Brick and Mortar Stores requires Request for Stores as input from an external producer
FROM: External TO: Human Resources	<ul style="list-style-type: none"> - Schedule Refund requires Reason for Return as input from an external producer - Send Notifications requires What to Send as input from an external producer

FROM: External TO: Browsing	<ul style="list-style-type: none"> - Add Item to Wishlist requires Notification Request as input from an external producer
FROM: External TO: User Data	<ul style="list-style-type: none"> - Save Payment Information requires Payment Information as input from an external producer - Save Delivery Address Information requires Delivery Address Information as input from an external producer

Table 5: I/O Dependencies within components

Components	Functions and Data
FROM: Shopping TO: Shopping	<ul style="list-style-type: none"> - Select Item in Store requires Store which is generated by Choose a Preferred Brick and Mortar Store - Add Item to Cart requires Item which is generated from Select Item in Store - Make a Purchase requires Shopping Cart which is generated by Add Item to Cart - Make a Purchase required Payment Information which is generated by Enter Payment Information - Create Subscription required Item Inventory which is allocated in Inventory - Choose a Preferred Brick and Mortar Store requires Brick and Mortar Database which is generated by Get All Brick and Mortar Store
FROM: Inventory TO: Inventory	<ul style="list-style-type: none"> - Update Store Inventory requires Inventory which is retrieved directly from Inventory where it is located
FROM: Human Resources TO: Human Resources	<ul style="list-style-type: none"> - Process Return Ticket requires Return Ticket which is generated by Schedule Refund - Issue Refund required Return Ticket which is generated by Schedule Refund
FROM: User Data TO: User Data	<ul style="list-style-type: none"> - Edit or Delete Subscriptions requires User Subscriptions which is allocated in User Data - View Subscriptions required User Subscriptions retrieved directly from User Data where it is allocated
FROM: Pick-Up and Delivery TO: Pick-Up and Delivery	<ul style="list-style-type: none"> - Update Pending orders requires New Pending Order which is generated by Schedule a Pickup Time

Table 6: Derivation Plan

1	Goal: Reliability	
1.1	BB Heuristic: Reduce Class Complexity - Weights (created a simpler system that is more robust and less likely to fail)	<ul style="list-style-type: none"> - Why: By reducing Class Complexity by weights, the resulting system will be more robust, reducing the chances of unforeseen errors occurring. - Priority justification: Improves reliability by reducing the chances of errors
2	Goal: Security	
2.1	BB Heuristic: Isolate risk - Data (Collect data associated with technology implementations that tend to change often to pinpoint security focus)	<ul style="list-style-type: none"> - Why: The intent is to improve the security of the systems data elements by taking into account technology implementations - Priority justification: The protection of data is a very high priority of a successful system.
3	Goal: Recoverability	
3.1	BB Heuristic: Group based on Architectural Style - Client/Server based on Data (allows for more essential functions to be saved on secure servers for quick recovery)	<ul style="list-style-type: none"> - Why: With the use of secure servers, essential data can be quickly restored in the case of local failure - Priority justification: In the case of system failure, a swift recovery is key in preserving the reputation of the system.
4	Goal: Usability	
4.1	BB Heuristic: Group based on Implementation Reality (allow existing off-the-shelf solutions to influence component functionality, allowing users to more easily learn the use of the system)	<ul style="list-style-type: none"> - Why: By aligning components and their respective functions to the capabilities of known off-the-shelf solutions, our system will feel more familiar to use by the user. - Priority justification: User experience is highly correlated to the ease of use of the system

5	Goal: Availability	
5.1	BB Heuristic: Group based on Architectural Style - Client/Server based on Data (allows for more essential functions to be saved on secure servers for quick recovery, improving uptime)	<ul style="list-style-type: none"> - Why: With the use of secure servers, essential data can be quickly restored in the case of local failure - Priority justification: In the case of system failure, a swift recovery is key in preserving the reputation of the system.
6	Goal: Response Time	
6.1	BB Heuristic: Reduce Data/Event Dependency (created a simpler system that is more robust and have a quicker response time)	<ul style="list-style-type: none"> - Why: By reducing data and event dependencies, the resulting system will be more robust, reducing the latency of actions. - Priority justification: Improves response time by reducing the complexity of the system
7	Goal: Maintainability	
7.1	BB Heuristic: Group based on Task Similarity (use similar combinations of data and function inheritance/decomposition to improve maintainability of the system)	<ul style="list-style-type: none"> - Why: Improve the cohesion of the system from having the same parent function allows for easier debugging and maintenance of the system. - Priority justification:
8	Goal: Evaluability	
8.1	BB Heuristic: Group based on Resource Demand collect functions based on similar resource dependencies to determine which sections of the system requires improvement)	<ul style="list-style-type: none"> - Why: By isolating components based on resource demand, the system can be more easily evaluated by administrators. - Priority justification: Evaluability is key for determining points of improvement for future maintenance of the system
9	Goal: Scalability	
9.1	BB Heuristic: Group based on Implementation Reality (allow existing off-the-shelf solutions to influence component functionality to allow easier transition of mediums)	<ul style="list-style-type: none"> - Why: By aligning components and their respective functions to the capabilities of known off-the-shelf solutions, our system will feel be easier to scale into other platforms.

		<ul style="list-style-type: none"> - Priority justification: Scalability is highly important if the system finds success and seeks to expand.
10	Goal: Adaptability	
10.1	BB Heuristic: Group based on Implementation Reality (allow existing off-the-shelf solutions to influence component functionality)	<ul style="list-style-type: none"> - Why: By aligning components and their respective functions to the capabilities of known off-the-shelf solutions, our system will feel be easier to scale into other platforms. - Priority justification: Scalability is highly important if the system finds success and seeks to adapt.

Discussion of Stakeholder needs

All stakeholder needs and qualities are references in the above plan.

Table 7: Potential conflicts and impact on derivation plan

Potential Conflict	Possible Resolution
Reduce Data/Event Dependency may conflict with other heuristics as it may limit the implementation structure of components.	Given its usage in Goal #1, apply Reduce Data/Event Dependency when bootstrapping the blueprint, reducing data and event dependencies when possible and applying other heuristics for refinement iterations.
Isolate risk - Data may conflict with other heuristics as it may limit the implementation of data structure and storage of components.	Given its usage in Goal #2, apply Isolate risk - Data as second priority behind Reduce Data/Event Dependency when bootstrapping the blueprint, isolating and classifying data allocation and storage.
Group based on Architectural Style may conflict with Isolate risk - Data as the former attempts to classify information based on functional style while the latter attempts to classify based on data types	Emphasize Isolate risk - Data given its usage in Goal #2

Group based on Implementation Reality may conflict with a number of other heuristics as it would force components boundaries base on existing solutions	Given its usage in Goal #9 and Goal #10, apply Group based on Implementation Reality when refining the system through later iterations
---	--

Description of the bootstrap given derivation plan

Based on the derivation plan, the bootstrap presented in this business blueprint satisfies all qualities and constraints presented by the stakeholders. The bootstrap adhered to the desires and requirements set by the stakeholders, while identifying clear guidelines for improvement and topics to revisit in later iterations of the system. Additionally, the bootstrap represented a cohesive design of the system, making it a reasonable starting point based on the derivation plan.

Part 3: Evaluate Business Blueprint Structure

Table 8: Coupling and Cohesion Metrics

Component	Number of I/O between components	Number of dependencies between components	Degree of Cohesion
Administration	3	3	0%
Shopping	17	5	0%
Inventory	14	2	50%
Human Resources	4	3	0%
User Data	4	3	50%
Pick-Up and Delivery	4	1	16.7%
Browsing	9	3	75%

Table 9: Size and Complexity Metrics

Number of components in the blueprint: 7

Component	Number of functions in component	Number of data elements in component	Component Complexity
Administration	7	10	20
Shopping	8	7	32
Inventory	4	4	22
Human Resources	4	4	12
User Data	6	8	18
Pick-Up and Delivery	6	5	15
Browsing	4	4	17

Support for Applied Heuristic

Reduce Class Complexity - Weights has been applied to this design as component complexity has been kept relatively similar. The biggest discrepancy between component complexities is between shopping and human resources with a difference of 20. However, this number is still relatively small compared to the complexity of the system, satisfying the heuristic of Reduces Class Complexity - Weights.