System and Software Architecture Description (SSAD)

COSMIC – SYSTEM

Team 02

Sam Lehardi

Project Manager/ Life Cycle Planner/ Trainer

Mishaal Aleem

Prototyper/ Trainer/ Implementer

Rachel Inouye

Operational Concept Engineer/ Trainer/ Implementer

Terence Williams

Software Architect/ Implementer

Sitan Gao

Feasibility Analyst/ Tester

Sam Villarreal

Requirements Engineer/ IIV & V/ Tester

Nick Kwong

Quality Focal Point/ IIV&V/ Tester

11/30/17

Version History

Date	Author	Version	Changes made	Rationale
10/05/17	Terence	1.0	• Added Introduction and System Analysis	• Initial draft for DC Package
10/11/17	Terence	1.1	• Updated Use Case diagrams • Updated Artifacts and Information diagram • Completed All document sections	• Based on ARB feedback from the professor • Final document for DC Package
11/30/17	Terence	1.2	• Updated Process Realization Diagrams	• Based on TA feedback from the DC Package

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the SSAD

The purpose of the SSAD is to provide a detailed description of the system architecture and its components. The document aims to identify the key users and how they interact with the system, as well as the functions they can perform while using the system based on their given role.

1.2 Status of the SSAD

This is the final version of the System and Software Architecture Description document. It includes updates to the System Analysis based on the feedback from the Architecture Review Board presentation. The Technology-Specific System Design and the Architectural Styles, Patterns and Frameworks sections have been completed.

2. System Analysis

2.1 System Analysis Overview

The purpose of COSMIC-System, is to create an efficient and organized inventory tracking system for USC Viterbi's STEM Education Outreach Programs. The system will be able to track the inventory of the STEM-EOP supplies, materials and equipment that support STEM activities for over 3000 students and teachers. By doing so the USC STEM-EOP will be better able to serve the schools, communities, and student organizations that it impacts. This will also reduce cost, save time and provide better lesson plans to the students that benefit from the program.

2.1.1 System Context

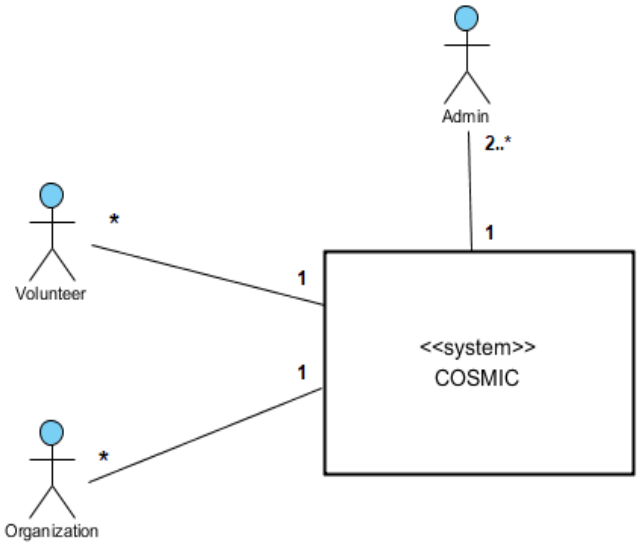

Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Admin	System Administrators who will ensure that the system is being used as intended	• Keep inventory up to date • Add/Remove items • Add/Delete users • Make/Cancel reservations • Update Item details
Student Volunteer	Users who are volunteers for the USC STEM-EOP	• View/Search inventory • Check in/out items • Reservation requests
Student Organization	Users who are members of student organizations or schools who are in partnership with the USC STEM-EOP	• View/Search inventory • Reservation requests

2.1.2 Artifacts & Information

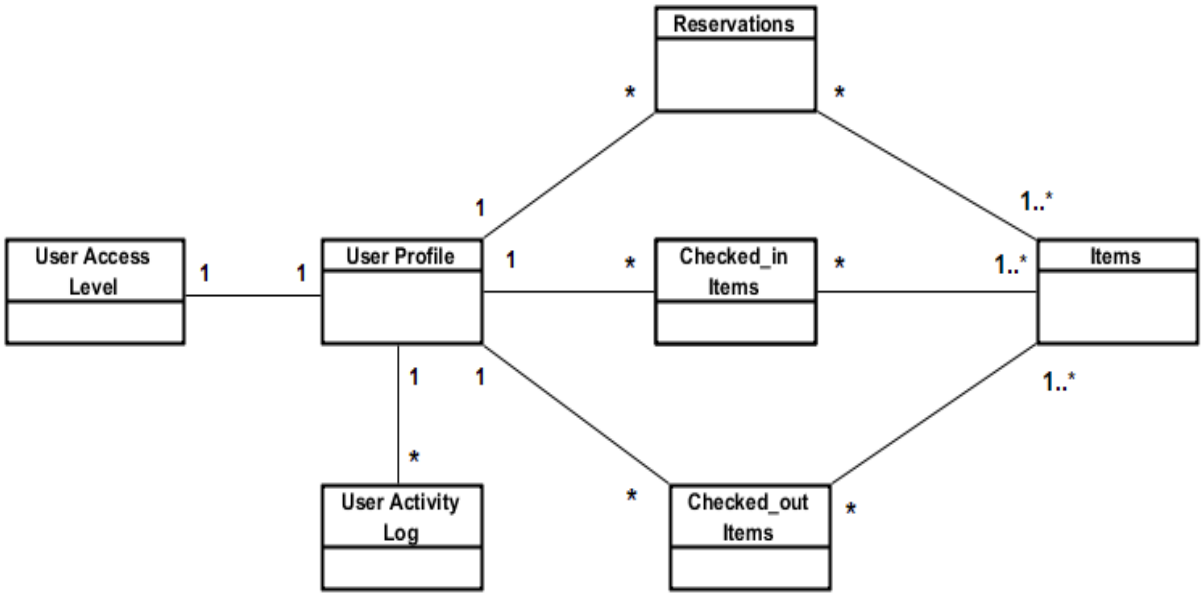

Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
ATF-1: Items	Items found in the inventory along with information about each item
ATF-2: Reservations	Contains items reserved for a specific user for a given date range. These reservations can only be made by the admin.
ATF-3: Checked Out Items	Items which have been checked out by a specific user
ATF-4: Checked In Items	Items which have been returned by the user who checked them out
ATF-5: User Profile	Contains information about a user, including: name, email address, password.
ATF-6: User Activity Log	Contains activity log for a given user
ATF-7: User Access Level	Access level privileges of a given user. This could be one of the following access levels: Admin, Volunteer, Organization

2.1.3 Behavior

Figure 3: Admin Use Case

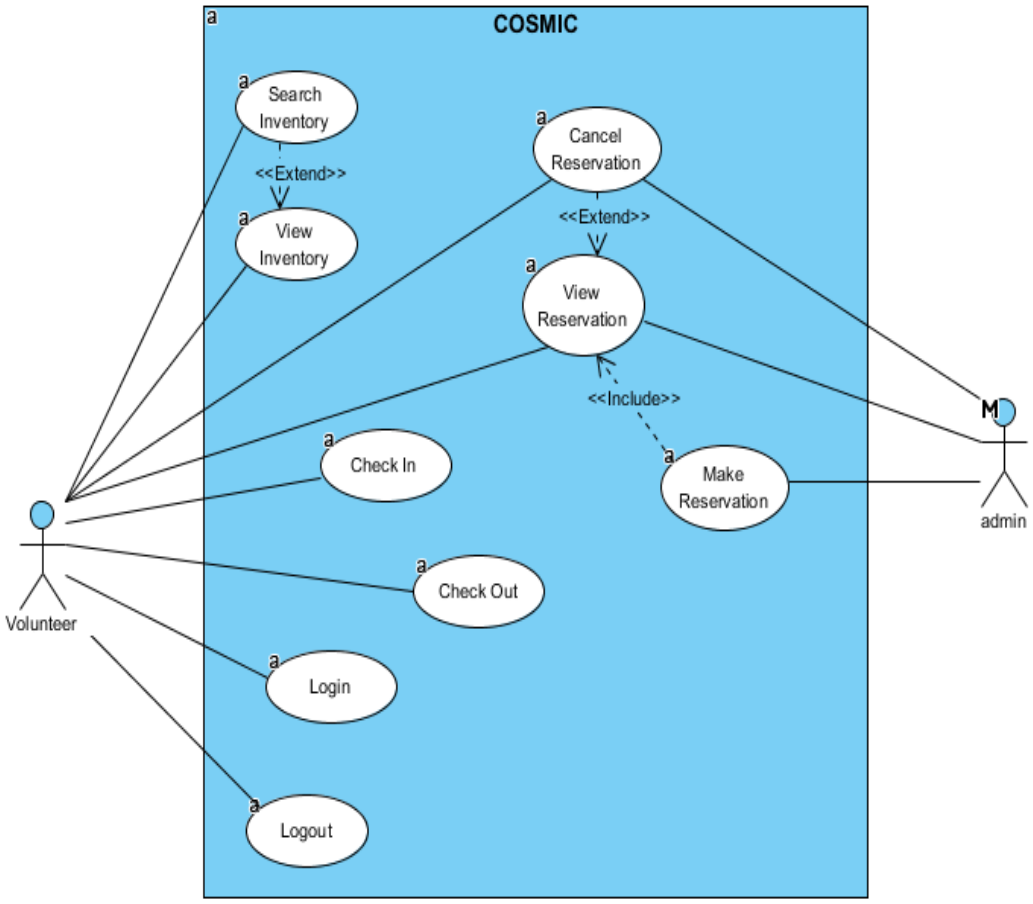

Figure 4: Volunteer Use Case

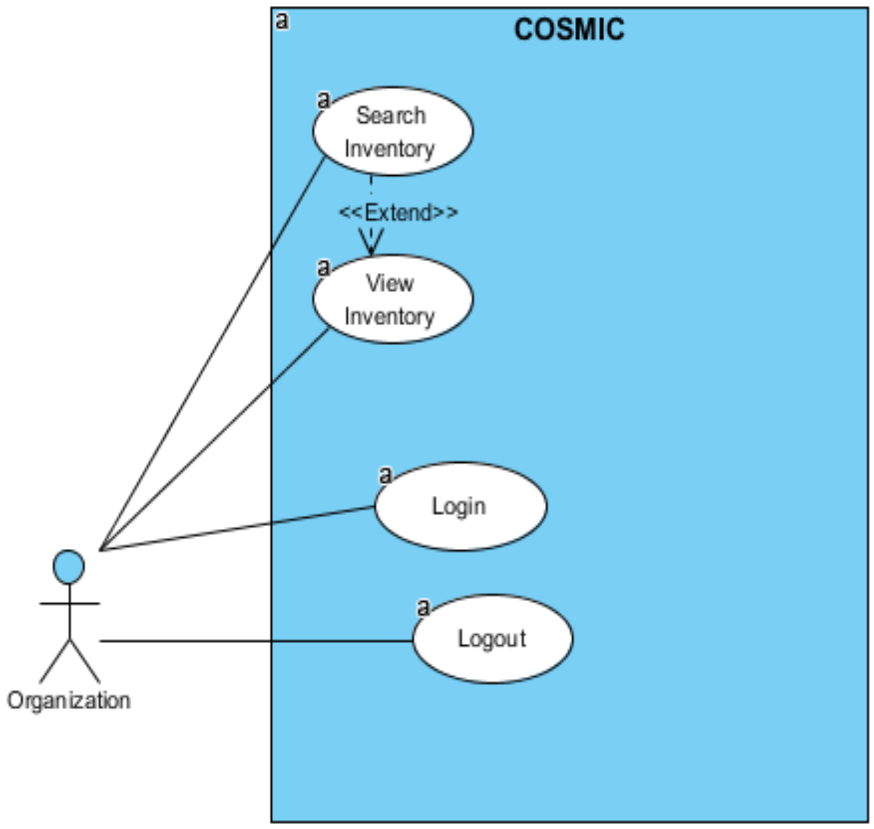

Figure 5: Organization Use Case

2.1.3.1 Authentication

2.1.3.1.1 Login

Table 3: Process Description: Login

Identifier	UC-1: Login
Purpose	Determines if the user logging in can be authenticated and if so their access level privilege.
Requirements	**WC_4446**
Development Risks	None
Pre-conditions	System database is properly initialized User is on the login page of the COSMIC-System website
Post-conditions	If user is authorized, he/she is login in the system and given

	access based on his/her assigned access privileges

Table 4: Typical Course of Action – Login Successful

Seq#	Actor's Action	System's Response
1	Enter username and password	
2	Click Login button	
3		Authenticate user credentials
4	Receive verification and session data	
5		Redirect to inventory page
6		Show success message "Logged in successfully"

Table 5: Alternate Course of Action – Login failure

Seq#	Actor's Action	System's Response
1 - 4	Refer to Typical Course of Action	
5		Display error message "No such user is registered"

2.1.3.1.2 Logout

Table 6: Process Description: Logout

Identifier	UC-2: Logout
Purpose	Logs user out of the website
Requirements	**WC_4446**
Development Risks	None
Pre-conditions	User is logged into the COSMIC-System website
Post-conditions	User is logged out of the COSMIC-System website

Table 7: Typical Course of Action – Logout Successful

Seq#	Actor's Action	System's Response
1	Go to user dashboard	
2	Click Logout button	
3		Log user out of system

4		Redirect to login page
5		Show success message “Logged out successfully”

2.1.3.2 Inventory Functions

2.1.3.2.1 View Inventory

Table 8: Process Description: View Inventory

Identifier	UC-3: View Inventory
Purpose	Allows user to browse through the items in the inventory
Requirements	**WC_4476, WC_4552, WC_4482, WC_4432,**
Development Risks	None
Pre-conditions	Database is initialized properly User is on the Inventory page
Post-conditions	User can view the inventory

Table 9: Typical Course of Action – View Inventory Successful

Seq#	Actor’s Action	System’s Response
1	Go to inventory page	
2		Populate items list
3	View items	

2.1.3.2.2 Search Inventory

Table 10: Process Description: Search Inventory

Identifier	UC-4: Search Inventory
Purpose	Allows user to search the inventory by item name or tag
Requirements	**WC_4551, WC_4552, WC_4482, WC_4432**
Development Risks	None
Pre-conditions	Database is initialized properly User is on the inventory page
Post-conditions	User can search for items in the inventory

Table 11: Typical Course of Action – Item Found

Seq#	Actor's Action	System's Response
1	Enter name/tag in the search bar	
2		Search database for item
3		Return search results
4	Click on desired item	

Table 12: Alternative Course of Action – Item Not Found

Seq#	Actor's Action	System's Response
1-3	Refer to Typical Course of action	
4	Change search criteria	

2.1.3.3 Item Functions

2.1.3.3.1 Check Out

Table 13: Process Description: Check Out

Identifier	UC-5: Check Out
Purpose	Allows Admins and Volunteers to check out items from the inventory.
Requirements	**WC_4447, WC_4483**
Development Risks	None
Pre-conditions	Database is initialized properly User has Admin/Volunteer access privilege
Post-conditions	User can checkout inventory items

Table 14: Typical Course of Action – Check Out Successful

Seq#	Actor's Action	System's Response
1	Click on Item	
2	Click Check Out button	
3	Enter item quantity	
4		Verify quantity requested is available
5		Update item total in database
6		Update item available total in inventory
7		Add item to "Items Checked Out" section of the given user's dashboard
8		Display success message "Item checked

		out”

Table 15: Alternative Course of Action – Check Out Failure

Seq#	Actor’s Action	System’s Response
1 - 4	Refer to Typical Course of Action	
5		Display error message “Please enter a valid quantity”

2.1.3.3.2 Check In

Table 16: Process Description: Check In

Identifier	UC-6: Check In
Purpose	Allows Admins and Volunteers to check in items they have checked out
Requirements	**WC_4475, WC_4483**
Development Risks	None
Pre-conditions	User has Admin/Volunteer access privilege User has an item checked out
Post-conditions	User can check in items that he/she has checked out

Table 17: Typical Course of Action – Check In Successful

Seq#	Actor’s Action	System’s Response
1	Go to user dashboard	
2	Select item from checked out items	
3	Click check in	
4		Provide check in form
5	Fill out check in form	
6		Update item total in database
7		Update item available total in inventory
8		Display success message “Item checked in”

Table 18: Alternative Course of Action – Check In Failure

Seq#	Actor's Action	System's Response
1 - 5	Refer to Typical Course of Action	
6		Display error message "Quantities cannot be empty"

2.1.3.3.3 View Reservation

Table 19: Process Description: View Reservation

Identifier	UC-7:
Purpose	Allows Admins and other users that reservations have been made on behalf of, to view their reservations
Requirements	**WC_4613**
Development Risks	Calendar Integration
Pre-conditions	Database is initialized properly User has Admin privilege, or a reservation has been made on their behalf
Post-conditions	User can view reservation details

Table 20: Typical Course of Action – View Reservation Successful

Seq#	Actor's Action	System's Response
1	Go to user dashboard	
2	Select item from "Items Reserved"	
3	Click view	
4		Get reservation details from database
5		Return reservation details

2.1.3.3.4 Cancel Reservation

Table 21: Process Description: Cancel Reservation

Identifier	UC-8:
Purpose	Allows Admins and other users that reservations have been made on behalf of, to cancel their reservations
Requirements	**WC_4613**

Development Risks	None
Pre-conditions	Database is initialized properly User has Admin privilege, or a reservation has been made on their behalf
Post-conditions	Reservation is cancelled and removed from database

Table 22: Typical Course of Action – User Cancelled Reservation

Seq#	Actor's Action	System's Response
1	Go to user dashboard	
2	Select item from "Items Reserved"	
3	Click cancel	
4		Remove reservation details from database
5		Remove reservation from item's "Upcoming Reservations" section

Table 23: Alternative Course of Action – Admin Cancelled Reservation

Seq#	Actor's Action	System's Response
1	Click on item	
2	Click on reservations	
3	Click cancel on desired reservation	
4		Remove reservation details from database
5		Remove reservation from item's "Upcoming Reservations" section

2.1.3.4 Admin Functions

2.1.3.4.1 Make Reservation

Table 24: Process Description: Make Reservation

Identifier	UC-9: Make Reservation
Purpose	Allows Admins to make reservations for themselves or other users
Requirements	**WC_4480**
Development	Calendar Integration, Restricting reservation time periods

Risks	
Pre-conditions	User has Admin access privilege Items needed for the reservation are available
Post-conditions	Items are reserved for the specified user

Table 25: Typical Course of Action – Make Reservation Successful

Seq#	Actor's Action	System's Response
1	Click on item	
2	Click on reservations	
3	Enter reservation details	
4	Click Add Reservation button	
5		Reserve item for specified user
6		Add reservation to upcoming reservations
7		Add item to specified user dashboard's reserved items section
8		Update item total in database
9		Update item available total in inventory

2.1.3.4.2 Add User

Table 26: Process Description: Add User

Identifier	UC-10: Add User
Purpose	Add new users to the system
Requirements	**WC_4446**
Development Risks	None
Pre-conditions	Database is initialized properly User has Admin access privilege
Post-conditions	New user added to the system

Table 27: Typical Course of Action – User Added Successfully

Seq#	Actor's Action	System's Response
1	Go to user dashboard	
2	Click on register a new user	
3	Enter user details	
5		Add user to database
6		Create user account

7		Display success message “User account created successfully”

2.1.3.4.3 Delete User

Table 28: Process Description: Delete User

Identifier	UC-11: Delete User
Purpose	Delete users from the system
Requirements	**WC_4446**
Development Risks	None
Pre-conditions	Database is initialized properly User has Admin access privilege
Post-conditions	User deleted from system

Table 29: Typical Course of Action – User Deleted Successfully

Seq#	Actor’s Action	System’s Response
1	Go to user dashboard	
2	Click on delete user	
3	Choose user to delete	
4		Remove user from database
5		Display success message “User account deleted successfully”

2.1.3.4.4 Add Item

Table 30: Process Description: Add Item

Identifier	UC-12: Add Item
Purpose	Add item to the inventory
Requirements	**WC_4430**
Development Risks	None
Pre-conditions	Database is initialized properly User has Admin access privilege
Post-conditions	Item added into the inventory

Table 31: Typical Course of Action – Item Added Successfully

Seq#	Actor's Action	System's Response
1	Go to user dashboard	
2	Click on add a new item	
3	Enter item information	
4	Upload item photo	
5		Add item to the database
6		Add item to the inventory
7		Display success message "Item added successfully"

2.1.3.4.5 Delete Item

Table 32: Process Description: Delete Item

Identifier	UC-13: Remove Item
Purpose	Remove item from the inventory
Requirements	**WC_4430**
Development Risks	None
Pre-conditions	Database is initialized properly User has Admin access privilege
Post-conditions	Item removed from the inventory

Table 33: Typical Course of Action – Item Deleted Successfully

Seq#	Actor's Action	System's Response
1	Go to user dashboard	
2	Click on delete item	
3	Choose item	
4		Remove item from database
5		Remove item from the inventory
6		Display success message "Item removed successfully"

2.1.3.4.6 Update Item Details

Table 34: Process Description: Update Item Details

Identifier	UC-14: Update Item Details
Purpose	Update item details in the inventory
Requirements	**WC_4479**
Development Risks	None
Pre-conditions	Database is initialized properly User has Admin access privilege
Post-conditions	Item information updated in the inventory

Table 35: Typical Course of Action – Item Details Updated Successfully

Seq#	Actor's Action	System's Response
1	Go to inventory page	
2	Click on item	
3	Click on update details	
4	Enter new information	
5		Update item information in the database
6		Update item information in the inventory
7		Display success message "Item updated successfully"

2.1.3.4.7 View Inventory Log

Table 36: Process Description: View Inventory Log

Identifier	UC-15: View Inventory Log
Purpose	View user and item activity log
Requirements	**WC_4431**
Development Risks	Lack of domain knowledge, scalability issues
Pre-conditions	Database is initialized properly User has Admin access privilege
Post-conditions	User can view inventory log activity

Table 37: Typical Course of Action – View Inventory Log Successfully

Seq#	Actor's Action	System's Response

1	Go to user dashboard	
2	Click on view inventory log	
3		Return database log data
4	View log data	

2.1.4 Modes of Operation

The COSMIC-System will only operate in one mode; therefore, no further information is required.

2.2 System Analysis Rationale

The COSMIC-System is an efficient and organized inventory tracking system for USC Viterbi's STEM Education Outreach Programs. There are three type of access levels present in the system. These roles are:

1. Admin – this user access level has all functions that the system can perform. They are responsible for keeping the system inventory up to date. They are responsible for creating user accounts for all other user types and they are also the only individuals who can make reservations.

2. Volunteer– these users are volunteers who work within the USC STEM-EOP. They have no admin privileges, but they can check in/out items.

3. Organization – this user type includes: student organizations, teachers and schools. They are only able to view the inventory.

All three types of these users can view and search the inventory, but what they are able to see and do within the inventory depends on their level of access.

3. Technology-Independent Model

This section has been intentionally omitted as our client has specified what technologies we should use for the proposed system.

4. Technology-Specific System Design

4.1 Design Overview

4.1.1 System Structure

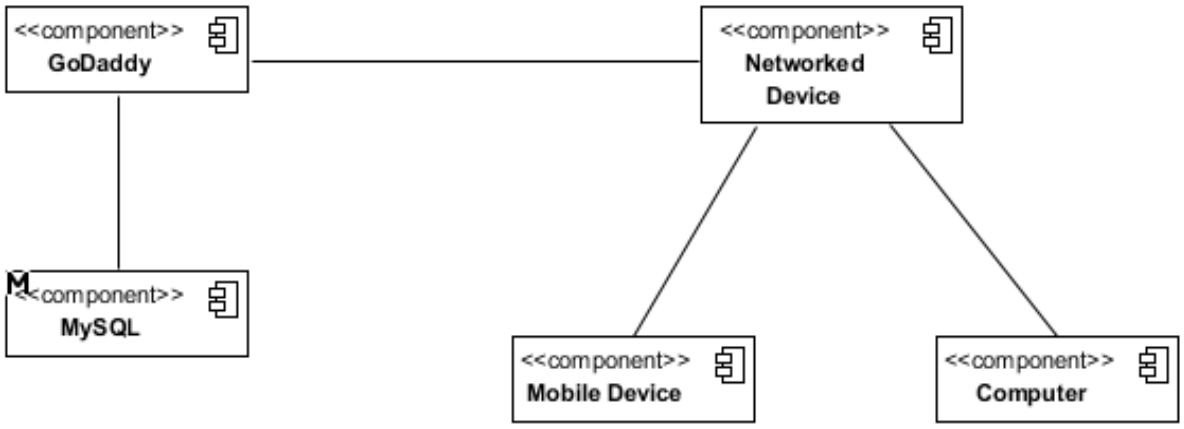

Figure 6: Hardware Component Class Diagram

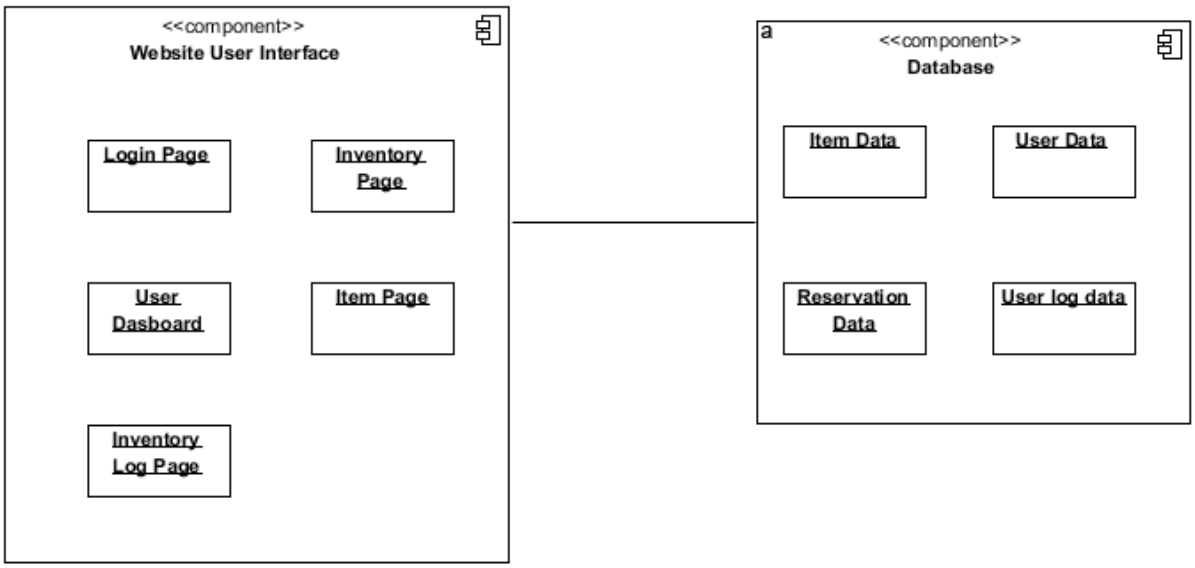

Figure 7: Software Component Class Diagram

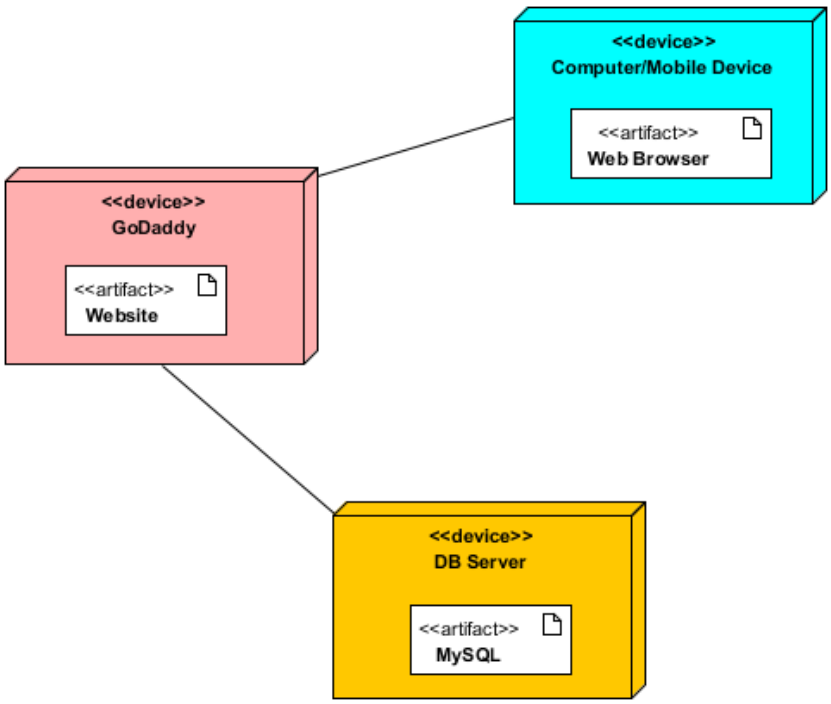

Figure 8: Deployment Diagram

Table 38: Hardware Component Description

Hardware Component	Description
Networked Device	Any device that can connect to the website
Computer	Desktop computer or portable laptops
Mobile Device	Android or IOS device
GoDaddy	Server that hosts the website
MySQL	Database which feeds the website

Table 39: Software Component Description

Software Component	Description
Login Page	User login page
Inventory Page	Default homepage that each user is directed to once they log in. It contains a list of all the items in the inventory
Item Page	Contains all the details for each specific item
Inventory Log page	Page where Admins can view the inventory system log
User Dashboard	Contains user specific information as well as items they have checked out or reserved
Item Data	Contains all the items in the inventory as well as their data
User Data	Contains all the users in the system as well as their data
Reservation Data	Contains the reservations in the system as well as the reservation information
User Log Data	Contains user activity system log

4.1.2 Design Classes

4.1.2.1 System Diagram

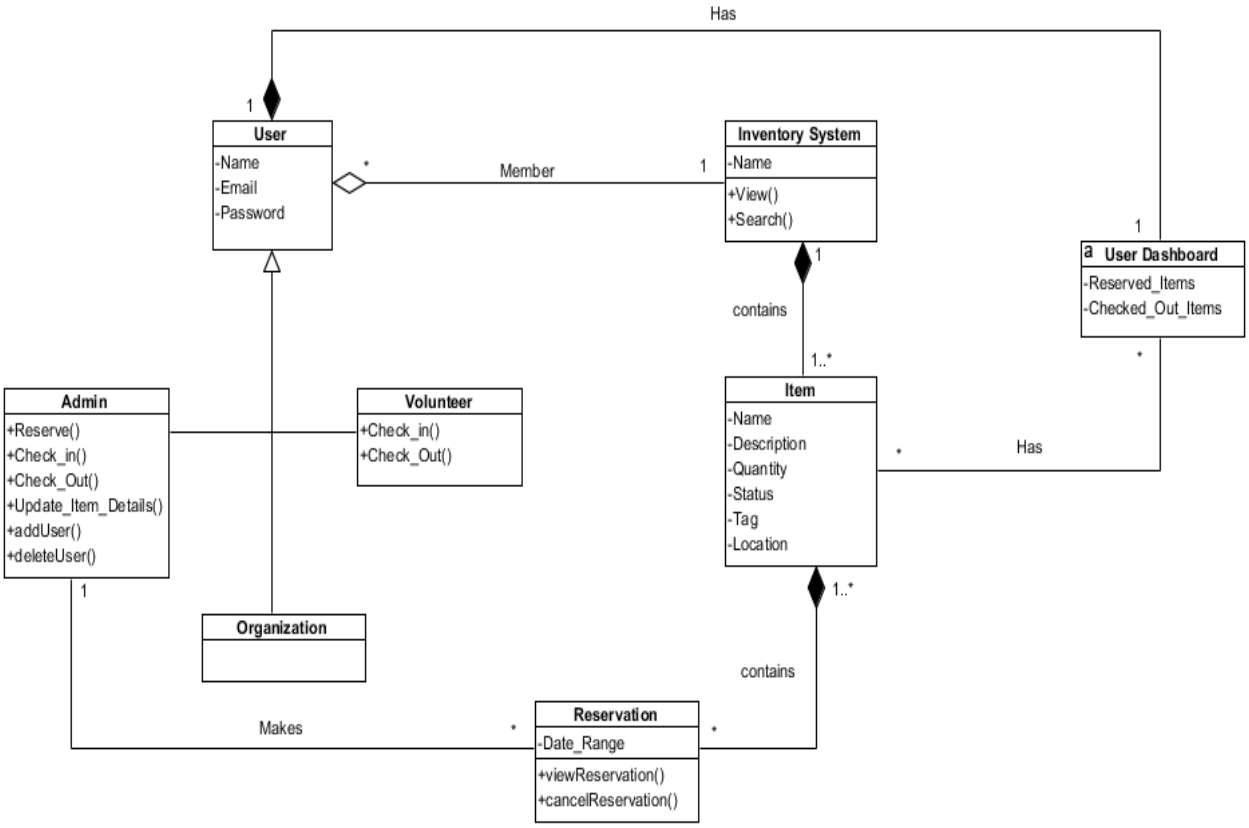

Figure 9: Design Class Diagram

Table 40: Design Class Description

Class	Type	Description
User	Entity	Users who have access to the system
Admin	Generalization	Users who have Admin privileges
Volunteer	Generalization	Users who have Volunteer privileges
Organization	Generalization	Users who have Organization privileges
Inventory System	Entity	System containing all the items which are found in the USC STEM-EOP
Item	Entity	Items which are found in the inventory system
Reservation	Component	Reservations made by Admins
User Dashboard	Component	Specific section where user can view their

		info as well as items they have checked out or reserved

4.1.3 Process Realization

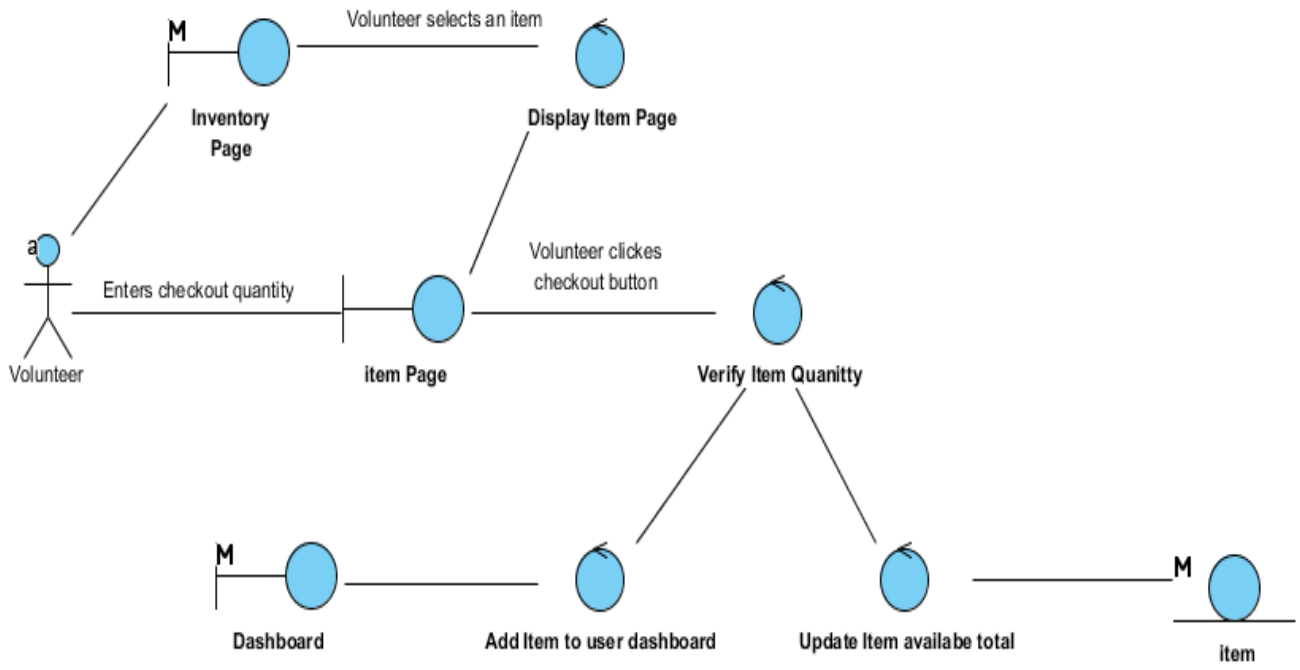

Figure 10: Robustness Diagram – Check Out

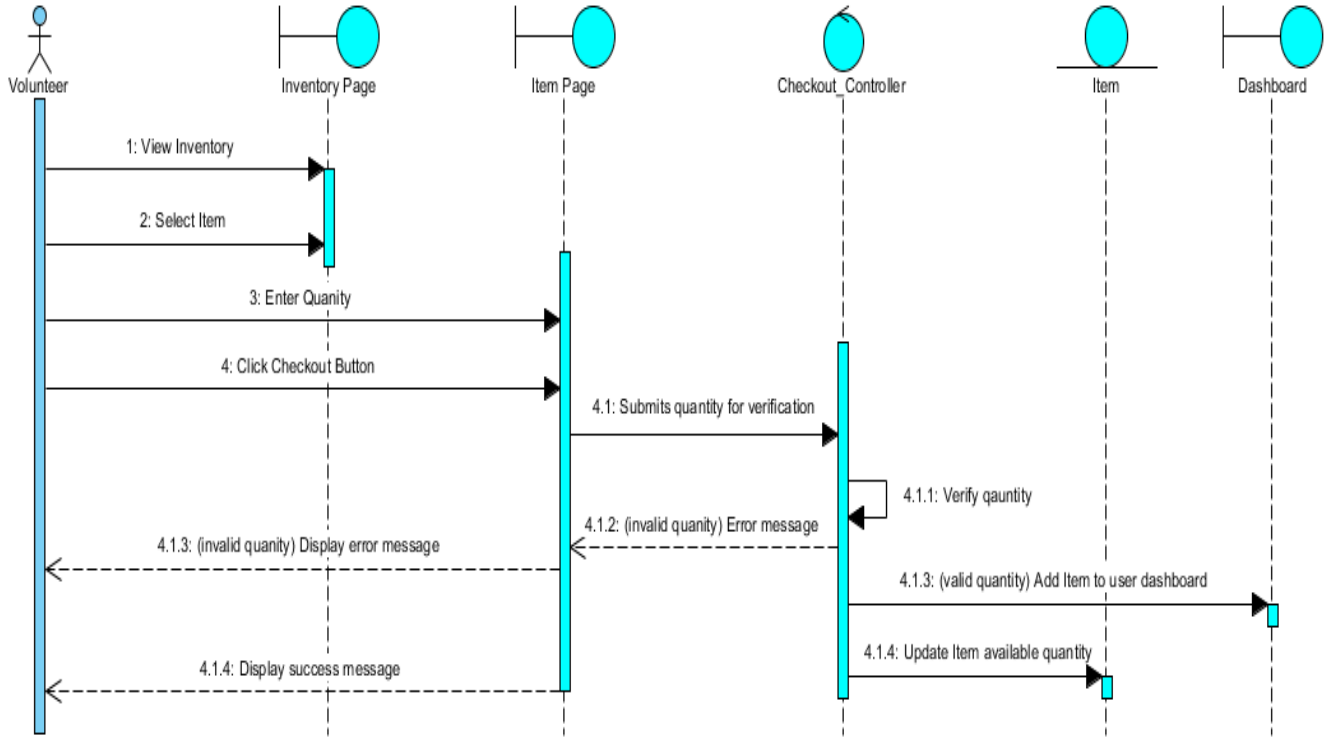

Figure 11: Sequence Diagram – Check Out

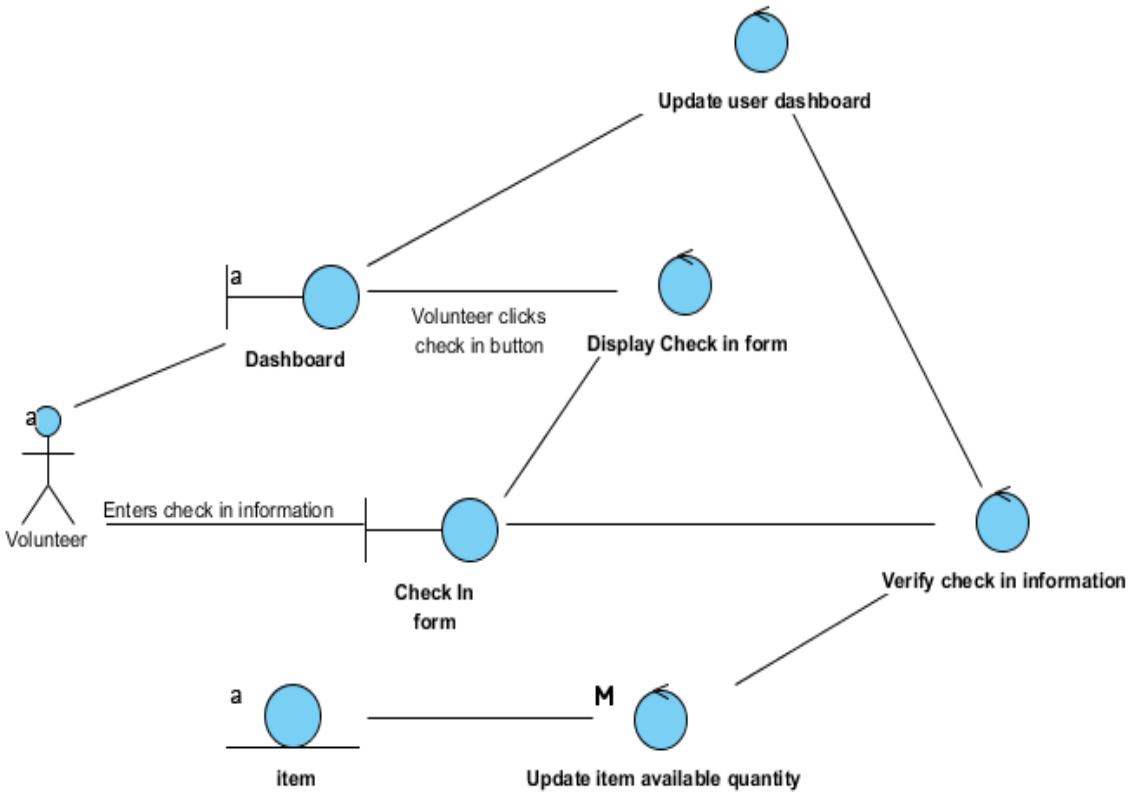

Figure 12: Robustness Diagram – Check In

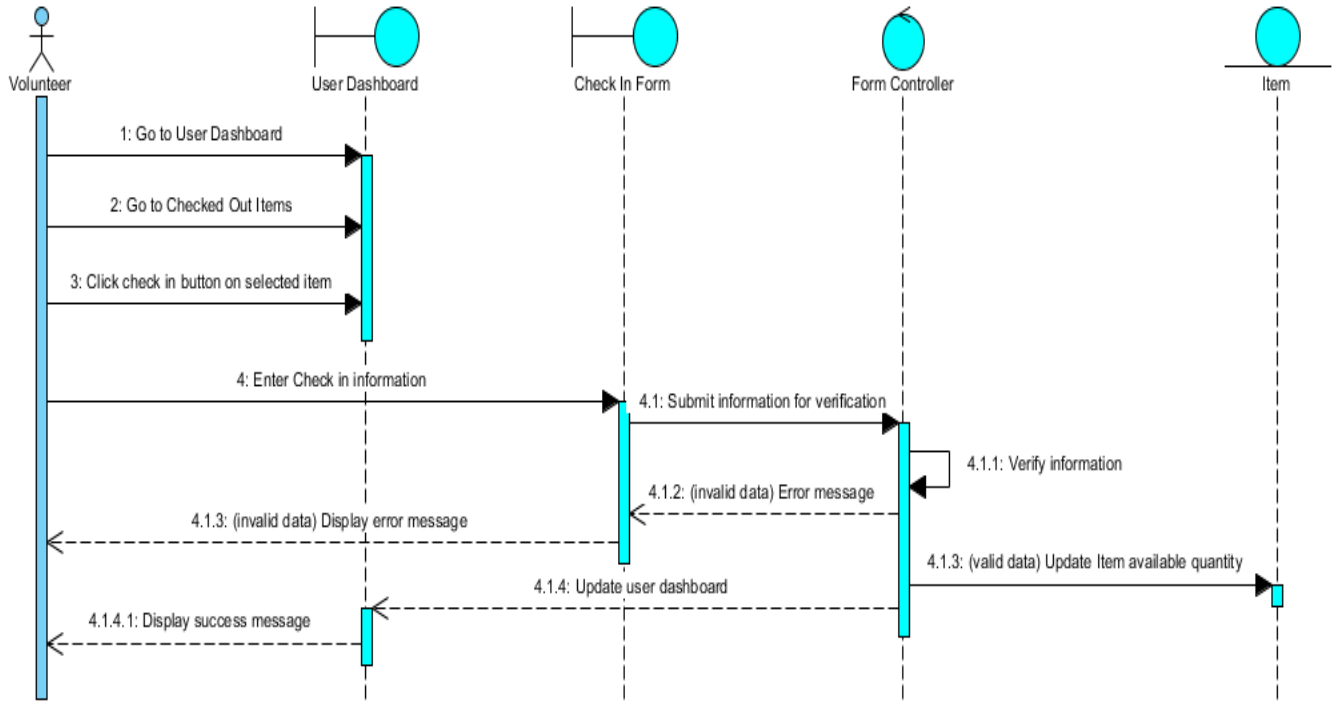

Figure 13: Sequence Diagram – Check In

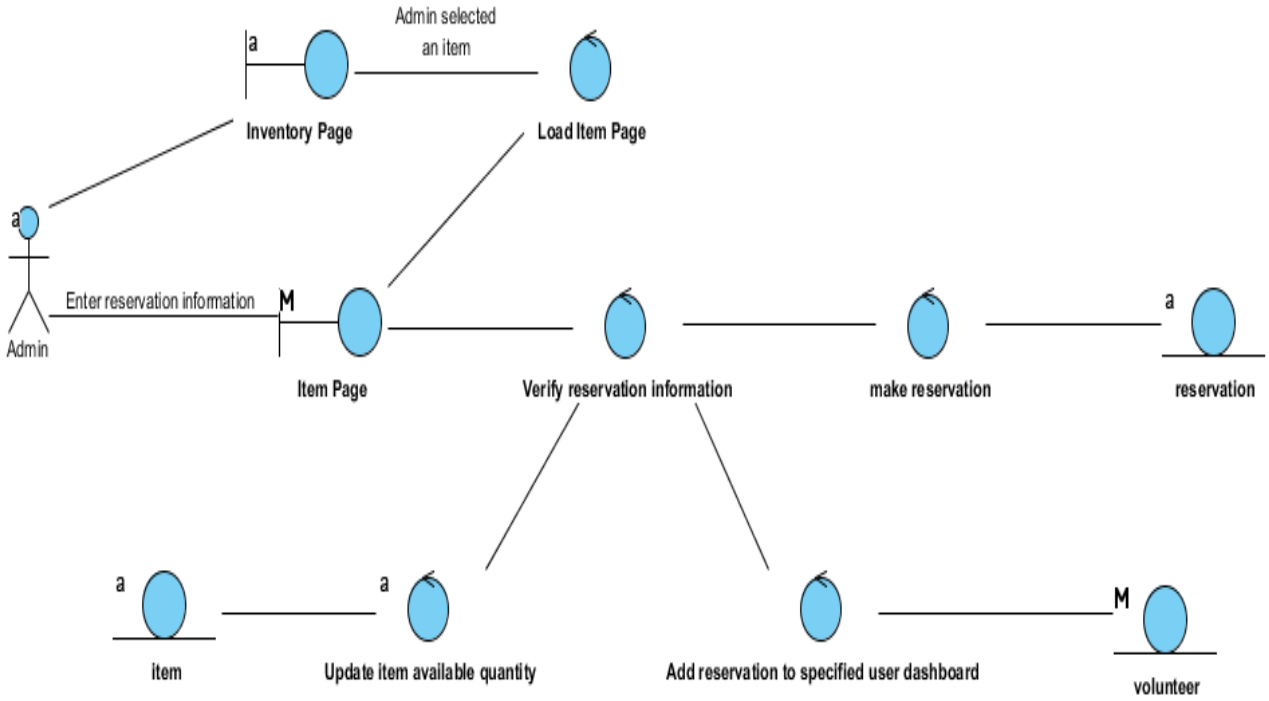

Figure 14: Robustness Diagram – Make Reservation

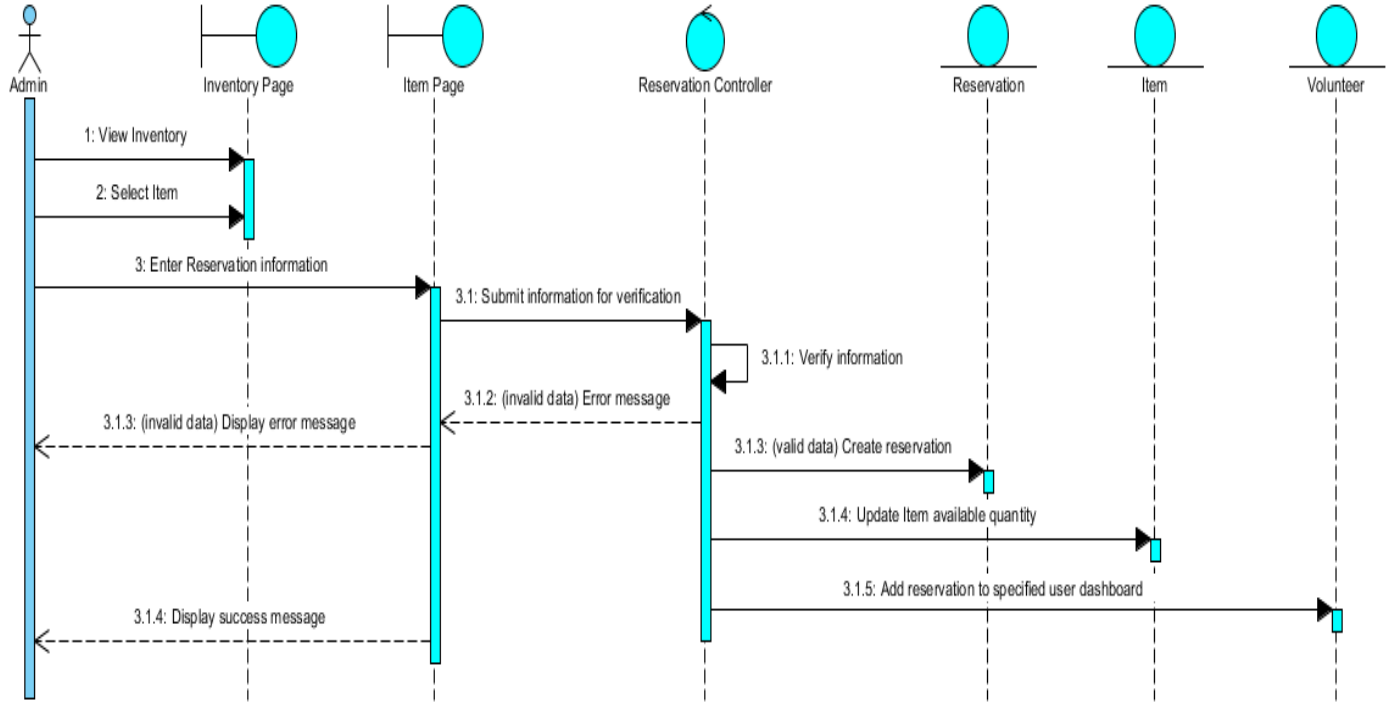

Figure 15: Sequence Diagram – Make Reservation

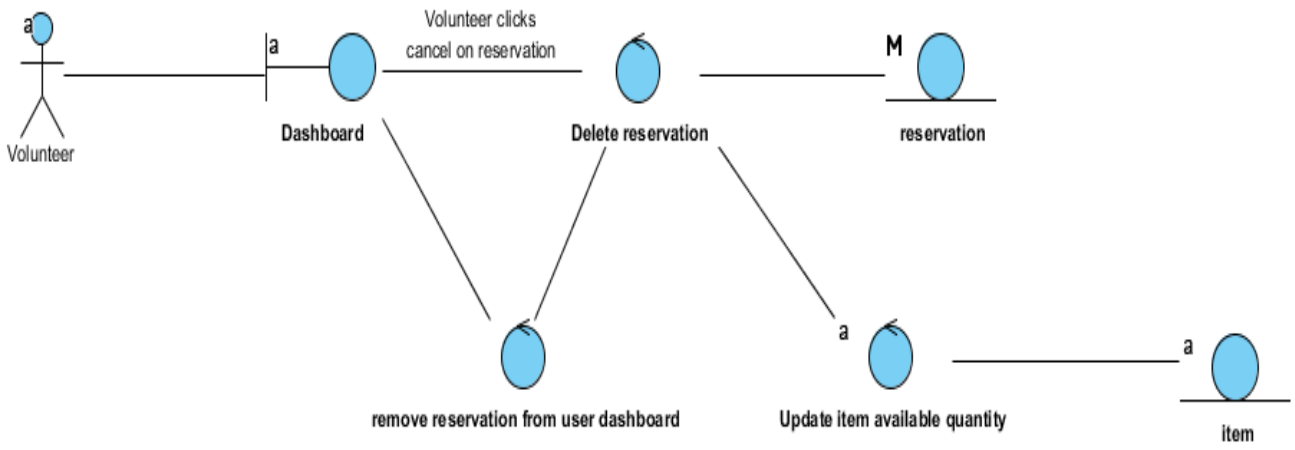

Figure 16: Robustness Diagram – User Reservation Cancellation

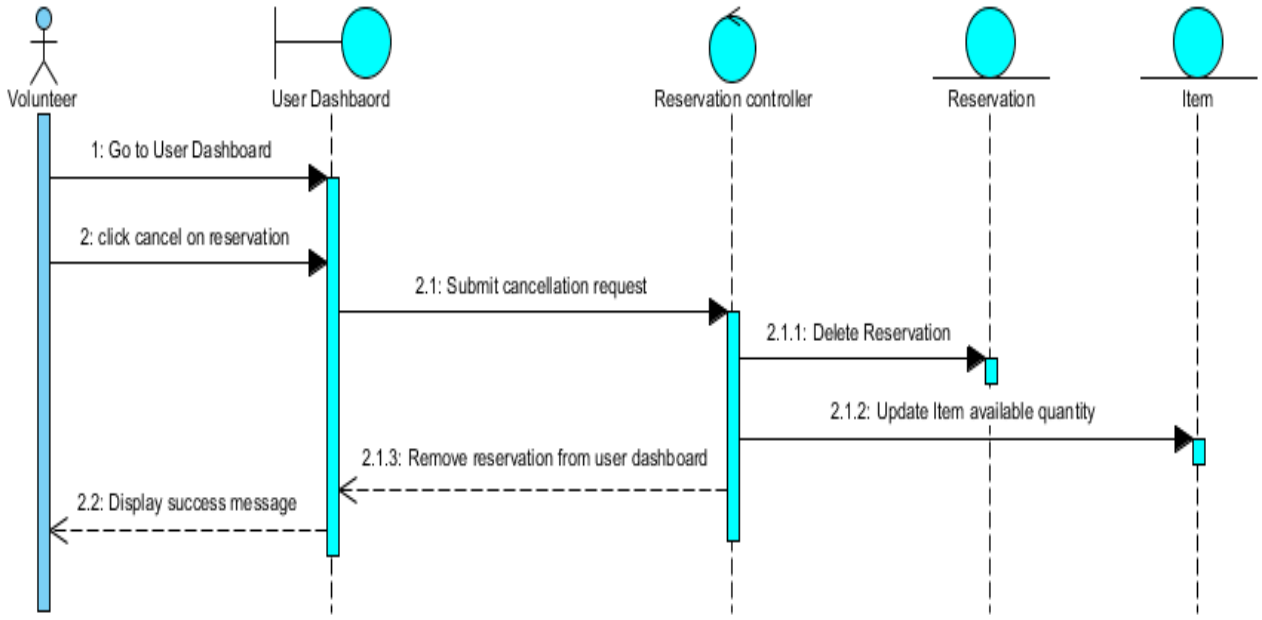

Figure 17: Sequence Diagram – User Reservation Cancellation

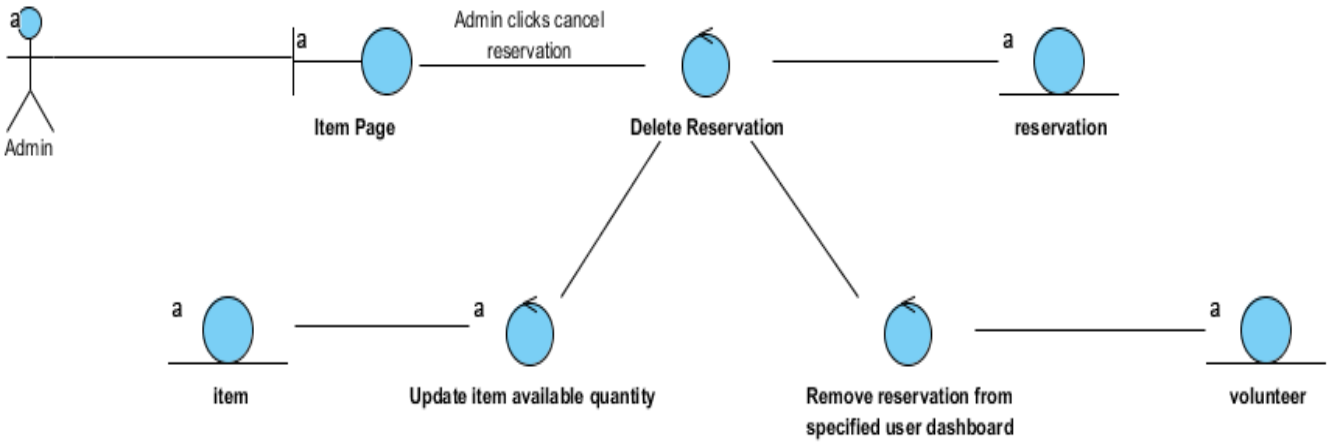

Figure 18: Robustness Diagram – Admin Reservation Cancellation

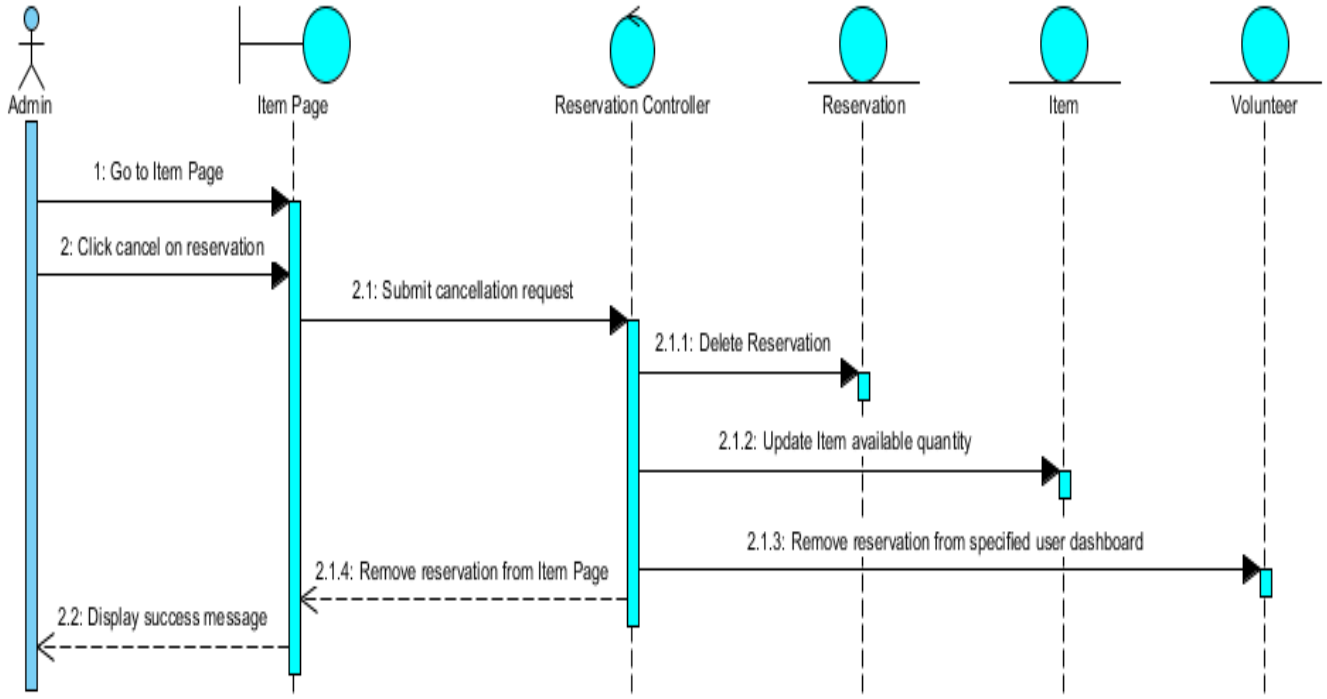

Figure 19: Sequence Diagram – Admin Reservation Cancellation

4.2 Design Rationale

The system is designed according to the client specification to be an efficient and organized inventory tracking system. The look and feel of the system is based on similar websites such as Amazon.

For the frontend portion of our system AngularJS and Bootstrap were used in to give the website a user friendly, intuitive look and feel. Since the system will be hosted on our client's GoDaddy account, MySQL and PHP are the two languages used in the backend to support the system database because these are the two languages that come with the client's hosting package. Both languages are more than suitable for the project being developed and the phpMyAdmin UI being used with the MySQL database makes it easier for non-technical Admin users to understand what is happening in the system.

5. Architectural Styles, Patterns and Frameworks

Table 41: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations
AngularJS	JavaScript open source framework	Benefits: • Mature frontend framework • Form Validation • Open source Limitations: • Steep learning curve
Bootstrap	Open source frontend web framework. Comprised of HTML and CSS elements for creating and styling webpage elements.	Benefits: • User friendly look and feel • Open source • Works well with other frameworks Limitations: • Hard to change web design outside of framework
3-Tier Architecture	Segments an application's components into three tiers of services. These tiers are: presentation tier, logical tier and data tier	Benefits: • Modular software • Ease of code maintenance • Better data management and security Limitations: • Increased complexity has system size increase