

AD Project Report

GDipSA47 Team 1

Contents

1.1 Introduction.....	2
1.2 Product Deliverables.....	2
1.3 Recommendations.....	3
1.4 Lessons Learned	4
1.5 Problems and Solutions.....	4
1.6 Looking back.....	4
2.1 Roles and responsibilities of each member.....	5
2.2 Deliverables and interaction within the team	6
2.3 Major Challenges and Resolution.....	10
2.4 Amount of time spent on the project by each member.....	13
2.5 Reflection on the experience gained from the project by each member.....	15
3.1 User Volume	17
3.2 Data Volume	18
3.3 Business transaction.....	20
3.4 Security Requirements	23
3.5 Reliability Requirements.....	24
8.1 Web Application for Department Sequence Diagram	40
8.2 Mobile Application for Department Sequence Diagram	49
8.3 Web Application for Store Sequence Diagram	52
8.4 Mobile Application for Store Sequence Diagram	71
10.1 Department web User interface.....	80
10.2 Store web User interface	92
10.3 Department Mobile User Interface	115
10.4 Store Mobile User Interface	121

1. Project Report

1.1 Introduction

The aim of the project is to develop a fully functional software system which would automate several work processes or main activities for the stationery inventory store as well as the internal customers who order with the said store. User Requirements gathering were conducted in the early phase of the project and from there, our team is able to break down the tasks into various phase with completion date for UAT signoff on the 11th of February.

1.2 Product Deliverables

Phase/Workflow	Deliverables	Elements in the Deliverables
----------------	--------------	------------------------------

Requirement Capture Workflow	Requirement Specification (Functional)	<ul style="list-style-type: none"> • Department Use Case Diagram for mobile application • Department Use Case Diagram for web application • Store Use Case Diagram for mobile application • Store Use Case Diagram for web application
	Requirement Specification (Non Functional)	<ul style="list-style-type: none"> • User Volume • Entity Volume • Business (use case) Transaction Volume • Security Requirement and Access Matrix
	UI Specification	Wireframe for each use case
Design Workflow	Architectural Platform	
	Final Design Sequence Diagram	Sequence diagram for each use case for both web and mobile application
	Consolidated Class Diagram	Class diagram
	Relational DB Design	Entity Relations Diagram
Implementation Workflow	Source Code	
Test Workflow	Functional testing for each Use Case	Test Scripts

1.3 Recommendations

We would like to recommend installation of electronic display tags for inventory and scanners, leading to increased efficiency and accuracy of the stock check and goods received process. We would also recommend the use of tablets for Store Clerks so they have a better interface when performing their tasks.

Future phases of development

Currently, only certain features are accessible in the mobile application. In the future we will continue to incrementally implement more features in the mobile application using AGILE approach to be as comprehensive as the web, as well as making the web application more compatible to be viewed on a mobile web browser; should the user chooses.

In the future, the application would also be able to have a fair allocation system across departments for disbursement, should the request quantity exceed the current stock level in the inventory.

The application will use Google Map API to allow the Department Representatives to track where the Store Clerk is and be alerted when they are nearby the collection point.

We propose the implementation of EDI (electronic data interchange) with suppliers so that purchases can be sent electronically instead of faxing or emailing as attachment. There is also potential for return of purchase confirmation date and quantity so that the clerks do not need to update purchases in the system should there be any changes from suppliers.

1.4 Lessons Learned

After breaking down a task, we learned that it was important for the entire team to go through all the completed subtasks to gain a better understanding of the whole project and to vet each other's work. This brings the whole team to a common understanding of the project and reduces the effort needed to address repeated questions or changes to codes that conflicts with the overall goal of the project.

We also learned that a change control is needed to properly process any alteration in the business requirements and design models. Before we implement the change control, a huge amount of errors occurred during programming, due to changes in the database which was not properly communicated to the rest of the team. After this incident, we start to establish change control to ensure that any changes should be made known to the team, followed by assessing the changes' impact, and implemented only with everyone's approval.

Finally we learnt that pair programming can increase the efficiency of programming. When two people are coding and or debugging, it is much faster to spot the mistakes and understand the logic of the code better, cutting down time used to program.

1.5 Problems and Solutions

Understanding the user requirements was quite challenging for the team since the perception and level of understanding for each team member was different. This situation was handled by conducting brainstorming sessions to arrive at a common consensus.

During the implementation of the actual mobile and web application, we could not adhere to the initial design model due to complexity and technological limitations of the development software. To overcome, this problem, the team did a lot of research and try out new codes to implement certain complex functionalities of the applications. We also modify the design model to simplify the implementation, while ensuring that the required functionalities of the system were met.

We also face problems during the integration of the source code from each member. As different members were using different versions of the development software to produce the codes, there were some incompatibility issues. To overcome this problem, we created a standardise library of codes of common functionalities and UI proven to work across different platforms that the team used to implement their respective use case. This greatly reduces the errors faced during integration.

1.6 Looking back

Currently, we have implemented a basic user-friendly UI for the application due to time constraint and lack of experience. Given time, we would explore and implement advanced front-end technologies for a better user-friendly interface and also come up with creative ideas of implementing other features apart from the user requirements which would benefit them.

2. Contribution report

2.1 Roles and responsibilities of each member

Name	Roles and Responsibilities
Adam Png Yee Soon	Programmer and Business Analyst
Aisha Ashiq	Programmer and Business Analyst
Bessa Nicoletta	Programmer and Business Analyst
Cen Xudong	Programmer and Business Analyst

Chen Wu-Chi	Technical Lead, System Integrator, Programmer and Business Analyst
Dicky Lie	Database Administrator, System Integrator, Programmer and Business Analyst
Kip Hoe Poon	Project Manager, Documentation Lead, Programmer and Business Analyst
Lily Ho Mei Lee	Programmer and Business Analyst
Madhumitha Murali	Database Administrator, Programmer and Business Analyst

2.2 Deliverables and interaction within the team

Name	Deliverables and interaction
Adam Png Yee Soon	<ul style="list-style-type: none"> ● Research and development on ASP.NET MVC ● Explained MVC concepts to other team members ● MVC starter project configuration and set up ● Web Purchase Order use case ● Web Adjustment Voucher use case ● Android Disbursement use case ● Android Authentication and Login ● Android Store Landing Pages ● Android Store Routing of Landing Pages to User Role ● Integration of Store Android ● Creation of parts of Project Work Plan ● Creation of Platform-specific Architecture ● Debugging of Web and Android ● Editing of non-technical work ● Populating database ● Wireframe for initial UI design and annotation for store and web ● Checking, updating and submission of the user manual ● Rational database table discussion ● User Requirement gathering ● Crystal Reports testing ● Popularized use of Postman for debugging

Aisha Ashiq	<ul style="list-style-type: none"> ● Android View Adjustment Voucher for Store Clerk ● Android View and Issue Adjustment Voucher for Store Supervisor and Manager ● Android Change DR/Collection Point for Department Head and Department Representative ● Android Approve/Reject Request for Department Head ● Testing of Web API with the above Android Use cases ● Performing System, Functional and Regression Testing for whole project (both web and android) ● Creating error report to be sent to team for fixing errors ● Generation of final Test Scripts ● UI Prototype designing ● Sequence Diagram for Store and Department ● Annotated Wireframes for Store and Department ● Created, checked and updated User manual for Store and Department ● Project Report ● Rational database table discussion ● User Requirement gathering ● Use case discussion with the whole team
Bessa Nicoletta	<ul style="list-style-type: none"> ● Populating database ● Created Power BI dashboard (Store and Department Trend Analysis) ● Android View Stock Card ● Android Purchase Order ● User Requirement gathering/ interview ● Rational database table discussion ● Use case diagrams ● Sequence diagrams ● Class diagrams ● Wireframe for initial UI design and annotation for store and web ● Perform Testing of Android and Web ● Inform Team for troubleshooting and bug fixing ● Checking and updating the user manual ● User Requirement gathering ● Researched data reporting methods
Cen Xudong	<ul style="list-style-type: none"> ● MVC project Configuration and set up ● Adjustment Voucher for Web ● Stock Card for Web ● View history request for DH in Web ● Inventory use case for Web ● Create Adjustment Voucher for Android ● Integration of Store Android ● Class Diagram for store and department ● Sequence Diagram for store and department ● Wireframe for initial UI design and annotation for store and web ● Project Work Plan ● Progress report week 1 ● Populating Database ● Email notifications for store ● Checking and updating the user manual ● Rational database table discussion ● User Requirement gathering

Chen Wu-Chi	<ul style="list-style-type: none"> ● Database and transaction method design ● Integration of asp.net MVC ● Discuss the business transaction with team and troubleshoot the problems ● Wireframe design ● Structure the code of business logic ● Raise and edit request use case (web) ● Raise and edit purchase order use case (web) ● Retrieve goods use case (both web and android) ● Disbursement use case (both web and android) ● Manage adjustment voucher use case (web) ● Manage stock card use case (web) ● Role delegation and change DR use case (web) ● Restful Web API ● Generate the roles in startup.cs for Asp.net identity ● Email notifications for store ● Bug fix for both android and web ● User Requirement gathering
Dicky Lie	<ul style="list-style-type: none"> ● Integration as well as code checks in android ● Debugging of android ● Role delegation for department head ● Change collection point and department representative for department head ● Change collection point for department representative ● Explained MVC concepts to other team members ● Help on adding validations for web and androids for many use cases ● Responsible on routing the url based on the user role ● Populating database ● Wireframe for initial UI design and annotation for store and web ● Use case discussion with the whole team ● Bug fix and testing for both android and web ● Checking and updating the user manual ● Email notifications for store ● User Requirement gathering

Kip Hoe Poon	<ul style="list-style-type: none"> ● Research the creation of spinner and how to set onclicklistener for the spinner to populate the listview ● Create a way to pass an intent across 3 activities back to the first activity in the Android studio ● Implement proper UI design for Android to be shared with the team. ● I familiarise myself in applying the WEB API to the Android, allowing me to mentor others in the team who were still unfamiliar in this area. ● Proactive in consulting and seek clarification on who to draw the sequence diagrams which allowed me to shape up the structure for others to follow. ● Store and develop wireframe and Use case discussion with the whole team ● Help in debugging codes ● Class Diagrams for store and department ● Write up the progress report, project report, project work plan, Non Functional requirement. ● Design the Entity Relational Diagram ● Project scheduling (GANTT) ● Checking and updating the user manual ● Rational database table discussion ● User Requirement gathering/Interview
Lily Ho Mei Lee	<ul style="list-style-type: none"> ● Created Power BI dashboard (purchasing and chargeback) and Web reports ● Android View Stock Card ● Android Purchase Order ● Crystal Reports form creation and MVC codes for adjustment vouchers and purchase orders ● User Requirement gathering/Interview ● Rational database table discussion ● Use cases for store and department ● Sequence Diagram for store and department ● Class Diagrams for store and department ● UI design ● Project scheduling (GANTT) ● Project work plan ● Project charter ● Project report ● Populating database for purchasing and adjustment ● Wireframe for initial UI design and annotation for store and web ● Perform Testing of Android and Web ● Inform Team for troubleshooting and bug fixing ● Checking and updating the user manual ● Researched data reporting methods

Madhumitha Murali	<ul style="list-style-type: none"> ● Android View Adjustment Voucher for Store Clerk ● Android View and Issue Adjustment Voucher for Store Supervisor and Manager ● Android - Search Catalogue for Department ● Android Department Landing Pages ● Database administrator ● Help and also taught teammates how to debug error codes ● Testing and Integration of Web API with the above Android Use cases ● Performing System and Regression Testing ● Creating error report to be sent to team for fixing existing errors ● Generation of final Test Scripts ● UI Prototype designing ● Sequence Diagram for Store and Department ● Annotated Wireframes for Store and Department ● Created, check and updated User manual for Store and Department ● Project Report ● Relational database table discussion ● User Requirement gathering
-------------------	--

2.3 Major Challenges and Resolution

Name	Major Challenges and Resolution
Adam Png Yee Soon	<p>Limited user interactions led to a heavier reliance on case study notes instead of actual interview insights, but this situation was mitigated by having team consensus on product development. The resultant product was able to meet user requirements.</p> <p>There was a need to apply and integrate novel technologies (JavaScript, jQuery and Power BI) in the web app development and while they were not difficult to learn, integration was not straight-forward so a lot of experimentation and testing were needed to get things working.</p> <p>User authentication was an off-syllabus topic and although Derek provided an Android workshop, it was abstruse to follow through. Wuchi and I researched and experimented on the user authentication concept in Visual Studio and then got it to work in Android Studio too.</p> <p>Integrating Store-Android components and general debugging were difficult as there were a lot of areas to analyse and quite often, a problem might even be caused by server-side code.</p>

Aisha Ashiq	Limited interaction with users, hence difficult to understand requirements completely. Finalizing the UI design was challenging because the team had different understandings of the user requirements. Somehow, through team discussions we came to a decision and worked on the project.
Bessa Nicoletta	<p>Use of an unfamiliar business analytics service to generate reports i.e. Power BI took time to learn</p> <p>Limited interaction with the users made it difficult to ascertain the requirements which impacted every component (use cases, sequence, class diagrams on top of coding) but was resolved by having the team agreed upon the design, and ultimately to achieve a workable solution</p>
Cen Xudong	Limited time to interact with user and sometimes may confuse with some unfamiliar use case and cause some problem in my code. Luckily ,my teammates found it and help me.Also it took some time to learn how to use the mvc, however finally we made it and learn a lot from this project.
Chen Wu-Chi	Not enough time to interact with user and gather requirements
Dicky Lie	<p>Limited time with a little interaction with the user. Learning how to use and familiarize with MVC took some time, however after getting used to MVC the project went more smoothly.</p> <p>Integration and debugging were hard since the code tend to have some problems when it was merged. All the team members were kind enough to help me understand their code and help me found the problem to fix.</p>
Kip Hoe Poon	<p>The use of WEB API is very specific on how it is used to interact with the mobile application using JSON. I found a problem where the WEB API requires an object so we had to convert a JSON array into a JSON object before we get it to interact with the WEB API.</p> <p>The use of StartActivityForResult is limited to passing intent across 2 activities. So in order for me to pass intent across 3 activities, I used an innovative way to use setResult and startActivityForResult across 3 activities.</p> <p>The other technical difficulties would be using spinner and its onclick listener to populate listview. As this was a new functionality I was exposed to, I resolve it by using spinner with AsyncTask to populate it, and use activity intent to pass the spinner selection to populate the listview with another AsyncTask</p>
Lily Ho Mei Lee	Limited time to do the project so we divided the group up with CNY in the last week. It helps to do pair programming(Bessa wasn't good at debugging especially that of the API communicating between android and web luckily Dicky taught me so I can understand and learn. Somethings you cannot learn on the internet and which you can learn from your teammates and share with others!

Madhumitha Murali	<p>Implementation of sending model as JSON from web to android. We create a view model and convert the model from database to view model before sending the data to android. It makes things easier as we can collect all the information that we need in the view model before we send it to the android.</p> <p>Limited interaction with users, hence difficult to understand requirements completely. Team discussion and brainstorming sessions helped to overcome the above issue.</p>
-------------------	---

2.4 Amount of time spent on the project by each member

Name	Amount of time spent
Adam Png Yee Soon	<p><u>213.5 hours total</u></p> <p>Research & Development – 10 hours User Requirement – 5 hours Group Discussions – 45 hours Wireframes – 40 hours Use case Diagrams – 8 hours Sequence Diagrams – 3 hours Coding & Testing – 95.5 hours User Manual Editing and Compilation – 3 hours Other Documentation – 4 hours</p>
Aisha Ashiq	<p><u>193.5 hours total</u></p> <p>User Requirement- 5 hours Group Discussions -15 hours Use case Diagrams- 5 hours Sequence Diagrams- 10 hours Android Coding & Testing- 60 hours Wireframes- 33 hours User Manual - 30 hours Testing & Test Scripts- 20 hours Documentation – 15.5 hours</p>
Bessa Nicoletta	<p><u>198 hours total</u></p> <p>User Requirement – 5 hours Wireframe – 40 hours Group Discussions – 45 hours Use Case Diagram – 7 hours Sequence Diagram – 30 hours Class Diagram – 21 hours Populating Database – 15 hours Power BI Reports – 20 hours Android Coding – 7 hours (pair programming with Lily Lee) Testing android and web for all processes– 7 hours Documentation – 16 hours</p>
Cen Xudong	<p><u>199.5 hours total</u></p> <p>User Requirement – 5 hours Wireframe –40 hours Group Discussions – 45 hours Sequence and Class Diagram – 18 hours Coding and testing– 91.5 hours</p>
Chen Wu-Chi	<p><u>240 hours total</u></p> <p>User Requirement – 5 hours Group Discussions – 45 hours Wireframe – 30 hours Database Design – 20 hours Coding and testing – 100 hours Integrating and Debugging – 40 hours</p>
Dicky Lie	<p><u>233.5 hours total</u></p> <p>User Requirement – 5 hours Wireframe – 37.5 hours Group Discussions – 45 hours Database – 10 hours Coding and testing – 96 hours Integrating and Debugging – 40 hours</p>

Kip Hoe Poon	<u>243.5 hours total</u> User Requirement – 5 hours Wireframe – 37.5 hours Group Discussions – 45 hours Database – 2 hours Coding and testing – 114 hours Sequence Diagram – 20 hours Creating Classes and Class Diagram – 20 hours
Lily Ho Mei Lee	<u>227 hours total</u> User Requirement – 5 hours Wireframe – 41 hours Group Discussions – 45 hours Use Case Diagram – 7 hours Sequence Diagram – 20 hours Creating Classes and Class Diagram – 30 hours Populating Database – 6 hours Power BI Reports – 15 hours Android Coding – 7 hours (pair programming with Lily Lee) Test run of the android codes to make sure it works – 5 Crystal reports creation - 21 Testing android and web for all processes – 10 hour Documentation – 15 hours
Madhumitha Murali	<u>203.5 hours total</u> User Requirement- 5 hours Group Discussions -15 hours Use case Diagrams- 5 hours Sequence Diagrams- 10 hours Database – 10 hours Android Coding & Testing- 60 hours Wireframes- 33 hours User Manual - 30 hours Testing & Test Scripts- 20 hours Documentation – 15.5 hours

2.5 Reflection on the experience gained from the project by each member

Name	Reflection
Adam Png Yee Soon	<p>Learning and applying new technologies is not as difficult as imagined and should always be considered when taking on a new project, because sometimes an unfamiliar technology might be even more suitable in implementation.</p> <p>The user gathering process must be thorough to ensure the final product can fulfil requirements. If there were a chance to do an interim check with the users, the resultant product will be even closer to their ideal.</p> <p>Work breakdown that is aligned to both project deadlines and each member's forte is essential for achieving project success. Active and committed participation by all team members is integral to high-performing teamwork.</p>

Aisha Ashiq	<p>Clear understanding of the user requirements is a must as we are going to build a system for them, hence it is very important to understand their needs and expectations and try to deliver them to the fullest</p> <p>Team discussion and considering other's opinions and ideas gives a lot more perception in a different way and hence can avoid last minute changes and problems</p>
Bessa Nicoletta	<p>User experience design is very important; however, at our level of competency it might be better to work on the coding first then do it in reverse</p> <p>Interaction with user cannot be on a one-time basis, there needs to be an on-going timely communication</p> <p>Deliverables and deadlines need to be clearly stated to make sure no impediments push back the final delivery date</p> <p>Teamwork is key, agree upon a workable solution first, a satisfying design, then build from there. Most importantly, trust each other.</p>
Cen Xudong	<p>The aim of design is to meet the required need and unmet need for user, the interaction with user is most important for a project, it decide the final result and evaluation to the whole project.</p> <p>Good Team distribution and communication are essential to finish the project, the former will influence the efficiency, the latter will influence the quality and effectiveness.</p>
Chen Wu-Chi	<p>Focus on not only UI but UX. Planning is important, and we need to spend more time to know the current business transaction approach because we can design the system to solve the user's pain points. Working with big group is not easy, we need to keep sharing the progress and the problems we faced, so we can release the stress when we are integrating the system.</p>
Dicky Lie	<p>Understand the user needs better to design a better application. Should focus on more user experience rather than focusing only the functionality of the application</p> <p>Good teamwork and communication are essential on finishing the project. Team members have their own skill set and by helping each other will help increase the project pace</p>
Kip Hoe Poon	<p>Clear understanding on how the Design pattern of MVC can promote much more usability and cleaner codes.</p> <p>It is also important to explore how user interact with their phones and their environment to come up with better user experience.</p> <p>The importance to tie down design with the implementation of the system. Previously, it was purely design without seeing how the design can impact the system implementation.</p>

Lily Ho Mei Lee	<p>Challenging to meet the expectations of all user requirement within the short frame of a month. Would like to implement all in the web and android but it was tough. However, I guess I have learnt that it is not possible even in real life and it is practical to implement the important use cases leaving the “nice to have” as the last to implement if there is enough time.</p>
Madhumitha Murali	<p>Team work, proper communication and understanding the strengths of each team member is vital for a successful project. Most importantly listening to all team members opinions and suggestions and arriving at a consensus is a must in every team.</p> <p>Learnt the importance of team discussions as it arrives at a clear understanding of the requirements and concepts.</p>

3. Non Functional Requirements

3.1 User Volume

User Volume table

User Group (Actor)	Country/ Location/ Count	Remark
Department Head	Singapore/various Logic University departments/10	Department Head is stationed most of the time in the department, however he is often on the go to other locations for his work commitments. He needs to use the system to grant approval to pending stationery requisition form and maybe delegate an acting department head when is out of the office.
Department Employee	Singapore/various Logic University departments/476	Department Employee are stationed in the department. They need to access the system when they want to raise or edit stationery requisition form.
Department representative	Singapore/various Logic University departments/9	Department Employee are stationed in the department. They need to access the system to raise or edit stationery requisition form and be notified when the stationery can be collected at his selected collection point. Department representatives are given the flexibility to change collection point.

Store Clerk	Singapore/Logic University Store/3	Store Clerk are stationed in the warehouse and will be more reliant on a mobile device than a desktop to access the system to carry out his duties. For example when he gets a notification on low stock level, stock retrieval or stock check.
Store Supervisor	Singapore/Logic University Store/1	Store Supervisor are stationed at the store and mainly use the system to check on the adjustment vouchers and issue adjustment vouchers to the Store Manager for adjustment voucher with greater amount than \$250.
Store Manager	Singapore/Logic University Store/1	Store Manager are stationed at the store and mainly use the system to check on the adjustment vouchers with values above \$250.

3.2 Data Volume

Data Volume table

Business Entity Class	Source Document	Retention Period	Target Volume
Request	Stationery Requisition Form	3 years	50 raise per week: 7800 request for 3 years
Staff	Staff profile in the system	5 year	500 records
Department	Department list	5 years	10 records
Inventory	Stationery catalogue	3 years	90 records
Collection point	Stationery Requisition Procedure for Employees	5 years	6 records
Supplier	Supplier List	3 years	10 records
Pricing	Stationery Supply Tender Quotation	2 year	180 records
AdjustmentVoucher	Inventory Adjustment Voucher	3 years	1 adjustment voucher per month; 36 adjustment voucher for 3 years.
PurchaseOrder	Stationery Purchase Order Form	3 years	10 purchase orders (1 per supplier) raised bi-weekly: 780 purchase orders
PurchaseOrderDetails	Stationery Purchase Order Form	3 years	10 items per purchase order (for max 10 suppliers) raised bi-weekly: 7800 purchase order details
Stockcard	Stock card	3 years	90 records

Documented Calculation for Data Volume table

User

User entity list out the username and password for each staff in the department. The record is kept relatively fixed unless there is new hire and exit of work personnel from the department which is difficult to predict the frequency.

Request

The request entity stores the list of stationery requisition form details. On average, 10% of the employees will request for stationery every week which makes up to 50 stationery requisition request per week. If the record is kept for 3 years, the total volume of record is 7800.

Staff

Staff entity list out the role and details for each staff in the department. The record is kept relatively fixed unless there is a promotion, new hire, exit of work personnel from the department which is difficult to predict the frequency.

Department

Department entity list out the details of the department. The record is fixed.

Inventory

The Inventory entity contains the details of the stationery including category, reorder level, reorder quantity and unit of measure for example. Currently Logic University Stationery Store has 90 different stationery. If the data of each product is retained for 3 years, the target volume to store the data of all products in the Inventory is remains relatively constant at 90 assuming that the record continuously deleted and replaced with a new stationery that is more price competitive.

Collection point

Collection point entity list out the details of the collection point. The record is fixed.

Supplier

The Supplier entity contains the details of the supplier including phone number, address, fax number, contact name for example. Currently Logic University Stationery Store has 10 suppliers. If the data of each supplier is retained for 3 years, the target volume to store the data of the suppliers remains relatively constant at 10 assuming that the record continuously deleted and replaced with a new supplier that offers a wider range of stationery with better pricing.

Pricing

The pricing entity records the price of each stationery offered by the respective supplier. Assuming each supplier provide tender price for 18 stationery, since there are 10 suppliers, there would be 180 record from the 10 suppliers.

Retrieval

The retrieval entity stores the total amount of stationery requested from the consolidated stationery requisition forms. 1 stationery retrieval form is produced every week. If the record is kept for 3 years, the total volume of record is 156.

Adjustment Voucher

The adjustment voucher entity stores the details of each adjustment details. 1 adjustment voucher is produced every month. If the record is kept for 3 years, the total volume of record is 36.

Purchase Order

The purchase order entity stores the details of each purchase order. The purchase order is raised bi-weekly on average so assuming one purchase order is raised for each supplier (there are 10 suppliers) and each purchase order is kept for 3 years, there would be 780 records.

Purchase Order Details

The purchase order details record the items in the purchase order. On average there are about 10 items per purchase order. Since, the purchase order is raised bi-weekly on average so assuming one purchase order is raised for each supplier (there are 10 suppliers) and each purchase order is kept for 3 years, there would be 7800 records.

Stockcard

The Stockcard entity contains the details for each change to the balance of the inventory. One stockcard is tied to one stationery. And since there are 90 different stationery, there will be a need for 90 records.

3.3 Business transaction

Business transaction table

Business Transaction (use case)	Persona	Business Criticality	No. of users	Transaction information	Concurrent important activity
Manage user account	All	Critical	500	500 login data per day 130500 per year (year 2019)	50 employees in each department can login on the system concurrently
Modify supplier information	Store manager/ Store supervisor	Moderate	2	Once a month 12 per year	Store manager and store supervisor modify supplier information
Issue adjustment voucher report	Store manager / Store supervisor	Moderate	2	Once a month 12 per year	Store manager and store supervisor can view and issue adjustment voucher concurrently
Process purchase order	Store clerk	Moderate	3	Non-peak period: 10 per 2 weeks Peak period (January and June): 20 per 2 weeks 2360 per year	3 store clerks can process purchase order concurrently
View stock card	Store clerk	Critical	3	90 per day 23490 per year	3 store clerks can view stock card concurrently
Modify stationery catalogue	Store clerk	Moderate	3	Once a year	3 store clerks can modify stationery catalogue concurrently
Process adjustment voucher	Store clerk	Moderate	3	Once a month 12 per year	3 store clerks can process adjustment voucher concurrently
View Report	Store clerk	Moderate	3	Once a month 12 per year	3 store clerks can generate trend analysis report concurrently
Generate charge back report	Store clerk	Moderate	3	10 per year	3 store clerks can generate charge back report concurrently
Process retrieval list	Store clerk	Moderate	3	3 per week 156 per year	3 store clerks can modify retrieval list concurrently
Process disbursement list	Store clerk	Moderate	3	10 per week 520 per year	3 store clerks can generate disbursement list concurrently
Delegate department head duties	Department head	Low	10	10 per year	10 department head can delegate their duties concurrently
Approve / reject request	Department head	Critical	10	500 per week 26000 per year	10 department head can approve / reject a request concurrently
Change department representative/ collection point	Department head	Low	10	20 per year	10 department head can change their representative / collection point concurrently
Modify department Information	Department head	Low	10	20 per year	10 department head can change their department information
View department trend analysis	Department head	Low	10	20 per year	10 department head can view trend analysis

View stationery catalog	Department head / Department representative / Staff department/ store clerk	Critical	500	Non-peak period: 50 per week Peak period (January and June): 100 per week 3000 per year	500 users can view the catalogue concurrently
Raise requests	Department representative / Staff department/ store clerk	Critical	495	Non-peak period: 50 per week Peak period (January and June): 100 per week 3000 per year	495 users can create request concurrently
Modify pending requests	Department representative/ Staff department	Critical	495	Non-peak period: 50 per week Peak period (January and June): 100 per week 3000 per year	495 users can view request concurrently
Manage cart	Department representative/ Staff department	Critical	495	Non-peak period: 50 per week Peak period (January and June): 100 per week 3000 per year	495 users can view request concurrently
Change collection point	Department representative	Low	10	20 per year	10 department representatives can change their collection point concurrently

Documented Calculation for Business Transaction Table

Documenting the calculation for transaction information

*In 2019 there will be 261 working days, assuming all the employees will be present for the whole year, there will be 500 logins per day ($500 * 261 = 130500$ logins per year)

**Assuming 10% of the employee will request for stationery every week, there will be 50 request per week for 10 months (non-peak period) and 100 request per week for 2 months (peak period).

Total requests = $100 * 2 \text{ months} * 4 \text{ weeks} + 50 * 44 \text{ weeks} = 3000$ requests per year

***The department head will receive notification for each request. Whenever the department head approve / reject a request, the respective requester will receive a notification. By using this assumption, the total notification will be ($3000 * 2 = 6000$ notifications per year). Assuming the total of the rest of notifications (change of collection details, receive reorder notification, delegation of department head duties, etc) are 150 notifications per year, the total notifications per year will be $6000 + 150 = 6150$ notifications per year.

****Assuming that per 2 weeks all the supplier will have a purchase order for non-peak period and per 2 weeks all the supplier will have 2 purchase order for peak period (January and June),

Since there are 10 suppliers, each bi-weekly will have 10 purchase order (non-peak period) and 20 purchase order (peak period).

The total purchase order in one year is = $20 * 2 \text{ months} * 4 \text{ weeks} / 2 + 10 * 10 \text{ months} * 44 \text{ weeks} / 2 = 2360$ purchase orders

3.4 Security Requirements

Security requirement (Data access)

Data (Entity class) \\ User Role (actor)	Department Head	Department representative	Employees	Store Clerk	Store Supervisor	Store Manager
Request	r	c,r,u,d				
Staff	c,r,u,d	r,u	r,u	r,u	r,u	r,u
Department	c,r,u,d					
Inventory	r	r	r	c,r,u,d	c,r,u,d	c,r,u,d
Collection point	u	u		u	u	u
Supplier				c,r,u,d	c,r,u,d	c,r,u,d
Pricing				c,r,u,d	c,r,u,d	c,r,u,d
AdjustmentVoucher				c,r,u,d	r	r
PurchaseOrder				c,r,u,d	c,r,u,d	c,r,u,d
PurchaseOrderDetail s				c,r,u,d	c,r,u,d	c,r,u,d
Stockcard				c,r,u,d	c,r,u,d	c,r,u,d

Security requirement (use case access)

Business Transaction (use case)\\ User Role (actor)	Department Head	Department representative	Employees	Store Clerk	Store Supervisor	Store Manager
Manage user account	x					
Modify adjustment voucher					x	x
Issue adjustment voucher					x	x
Process purchase order				x		
View stock card				x		
Modify stationery catalogue				x		
Process adjustment voucher				x		
View report				x		
Generate charge back report				x		
Process retrieval list				x		
Process disbursement list				x		
Delegate department head duties	x					
Approve / reject request	x					
Change department representative/ collection point	x					
View stationery catalogue	x	x	x			
Raise requests		x	x			

Modify pending requests		x	x			
Manage cart		x	x			
Change collection point		x				

3.5 Reliability Requirements

Reliability

Operation time

The operation time for the system will be the office hours of Logic University campus (9am - 5pm Mon to Fri; 9am - 12pm Sat). The down time for maintenance can occur from 2 am to 5 am in the morning, when the system is not being used or it can occur on Sunday. The downtime should not occur more than 1 times a month.

Hardware failure (server, network)

If there is a hardware failure in the server and the network, the recovery duration should not take more than an hour a day. As the system server connects to all the important functions to run the business (e.g. create requests, approve / reject requests, modify stock cards, etc), it is vital that the system recovers in not more than an hour to ensure insignificant disruptions to the business.

After the recovery has been made, the data concurrency cannot be more than an hour old. Users should be able to access data that are not more than half an hour old to ensure a smooth continuation of the business transaction.

*DRJ (<https://www.drj.com/articles/online-exclusive/5-disaster-recovery-options-balancing-the-pros-and-cons-objectives-and-cost.html>)

Disaster recovery

The maximum down time to do a disaster recovery is not more than one hour. This is a reasonable time needed to do recovery of the system and given that there is a robust system to pre-empt the disaster (refer to the Techniques to improve reliability).

After the recovery has been made, the data concurrency cannot be more than a day old. Users should be able to access data that are not more than half an hour old to ensure a smooth continuation of the business transaction.

Techniques to improve reliability

Data duplication:

Having a mirroring server to back up existing server, so that data backup is possible during disaster. Setting up a mirroring server can be useful especially when there is planned maintenance such as upgrading to new server or software updates which would require the system to be stopped or rebooted. With this mirroring strategy, the company can do a role switch in which the main server can be swapped at any time when planned maintenance is done.

Backup:

Also implementing an automatic backup system (e.g. Hewlett-Packard backup and recovery manager has a 7 days auto backup drive) that records and save active directory (email accounts, etc.) is also crucial.

Better Design:

Apart from having a mirroring server and backup system, setting up a firewall is also important for any business (e.g. Sophos firewall) which allows authentication of users before it reaches the server and database. In addition to subscribing to security software against harmful virus such as ransomware is necessary as well.

Putting in place spare resources:

The company has included a spare server in their yearly budget. When there is a main server failure, the system can temporarily connect to the spare server containing critical data required to address the essential functions of the system. This reduces the down time needed to connect to the mirroring server when the main server is down.

4. Architecture Platform Specification

Implementation Technology

Technology	Implementation
ASP.NET Web Application (.NET Framework) with Web API / MVC design pattern and EF6 Data Model	<p><u>MVC</u></p> <ul style="list-style-type: none"> • Rapid and parallel code development (3 times faster than applications developed using other development patterns) • Reduced code duplication from separation of concerns (low coupling of data, business logic and display) • High cohesion (logical grouping within Model, Controller and View) helps to reduce work error • Return of unformatted data ensures same component can be called for use with any interface • Support for asynchronous technique help application load faster • Possible integration with JavaScript Framework enables development of feature-rich web app • Modularity ensures increased unit testing possibilities • Feature-specific code modification does not need to affect entire programming development • A SEO-friendly platform can be easily developed • Greater cost and time savings can be achieved <p><u>Web API</u></p> <ul style="list-style-type: none"> • Required component for enabling RESTful services • Supports ASP.NET request/response pipeline • Built-in support for different formats of response data (JSON, XML, BSON) • Various hosting possibilities (IIS, Self-hosted or other web server that supports .NET 4.0+) allows easy simulation <p><u>EF6 Data Model</u></p> <ul style="list-style-type: none"> • Data model enables easy and seamless save/retrieve
JavaScript / jQuery	<ul style="list-style-type: none"> • Enhanced functionality of web pages with client side scripting • Increased user friendliness of website
HTML/CSS	Styling and display of data assets on web pages
Bootstrap	Application of selected HTML/CSS style theme to increase aesthetic value of website
JSON	Serialization and transmission of structured data
XML	Data setting specifications
Java	Language used in Android programming
C#	Language used in core programming to encode all required functionalities and database interactions in MVC
SQL Server	Data persistence

Tools

Application Area	Tool	Usage
Software Development	MS Visual Studio	Web application programming and implementation
	Android Studio	Android app programming and implementation
Database Connectivity	MS SQL Server Management Studio	<ul style="list-style-type: none"> • Database design and creation • Data persistence
Design Modelling	WhiteStar UML	Creation of sequence and class diagrams
Unit Testing	Postman	Simulation of GET/POST data operations to web application
	Firefox, Chrome and other web browsers	Demonstration of web application functionalities
Analytics & Reporting	SAP Crystal Reports	<ul style="list-style-type: none"> • Data visualization of business analytics • Interactive presentation on webpage • Generation of PDF reports for download
	MS Power BI	
Project Management	MS Project	Tracking and management of project activities
	MS Office Suite	

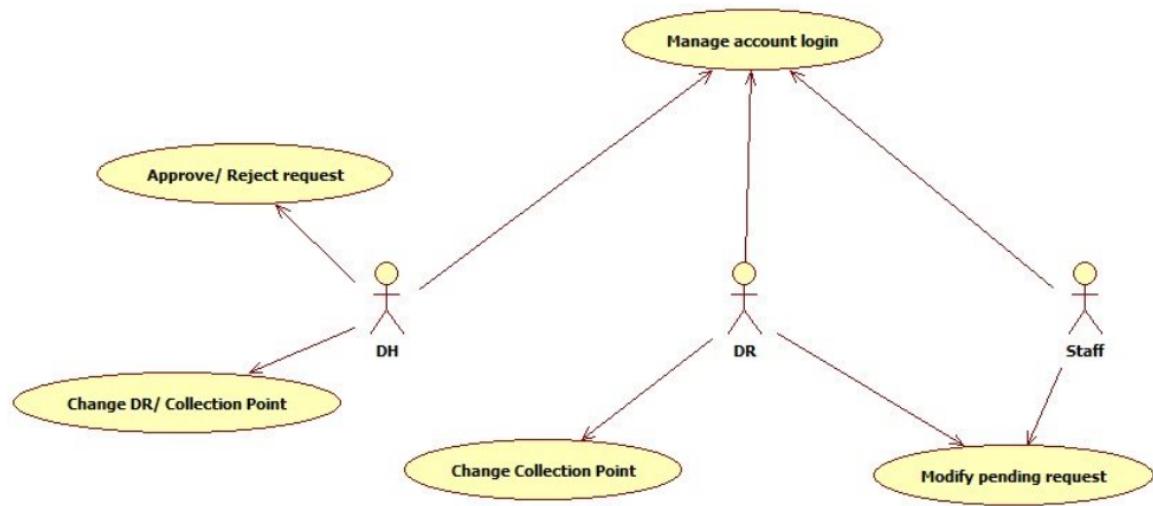
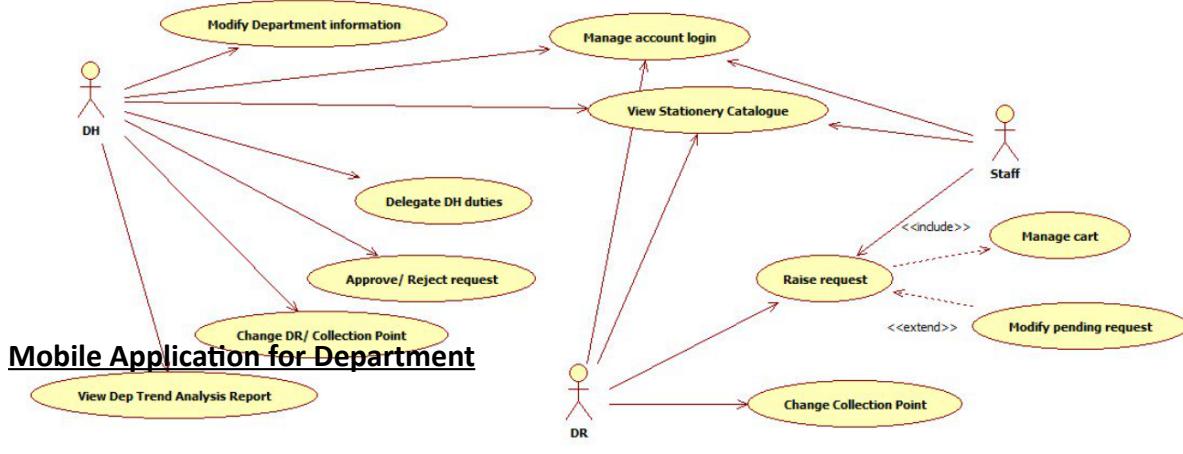
Software Architecture Design

	WEB	ANDROID
Presentation Layer		

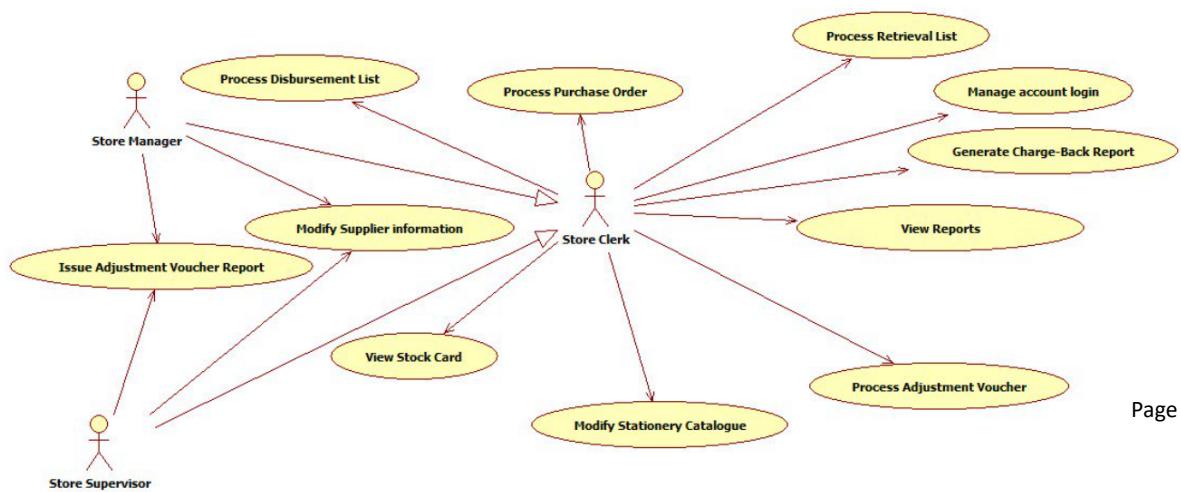
Business Logic Layer		
Persistence Layer		

5. Use Case Diagram

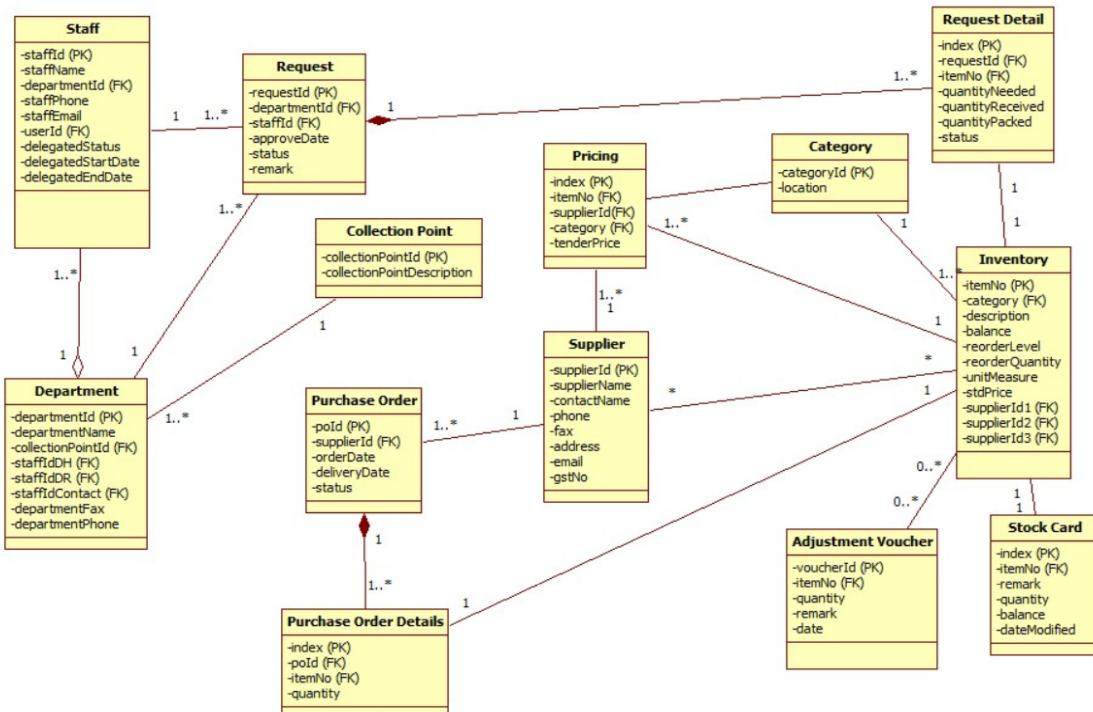
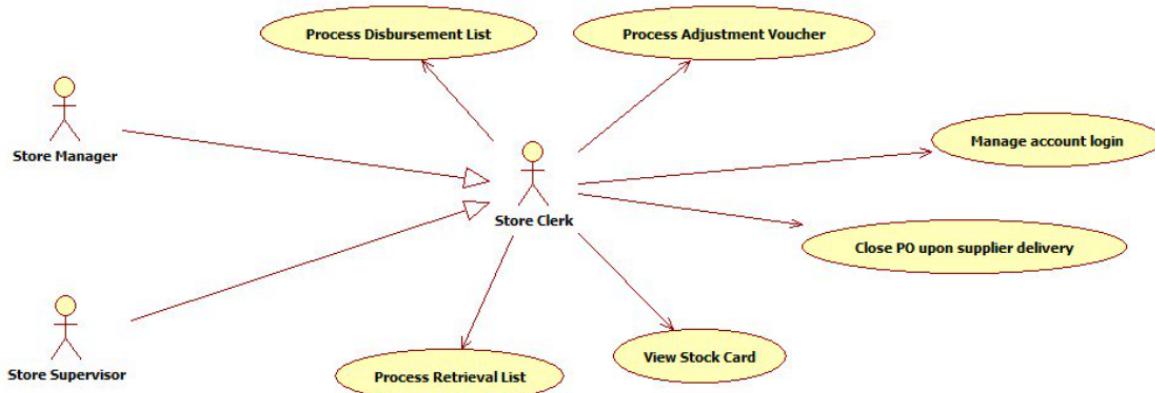
Web Application for Department



Web Application for Store



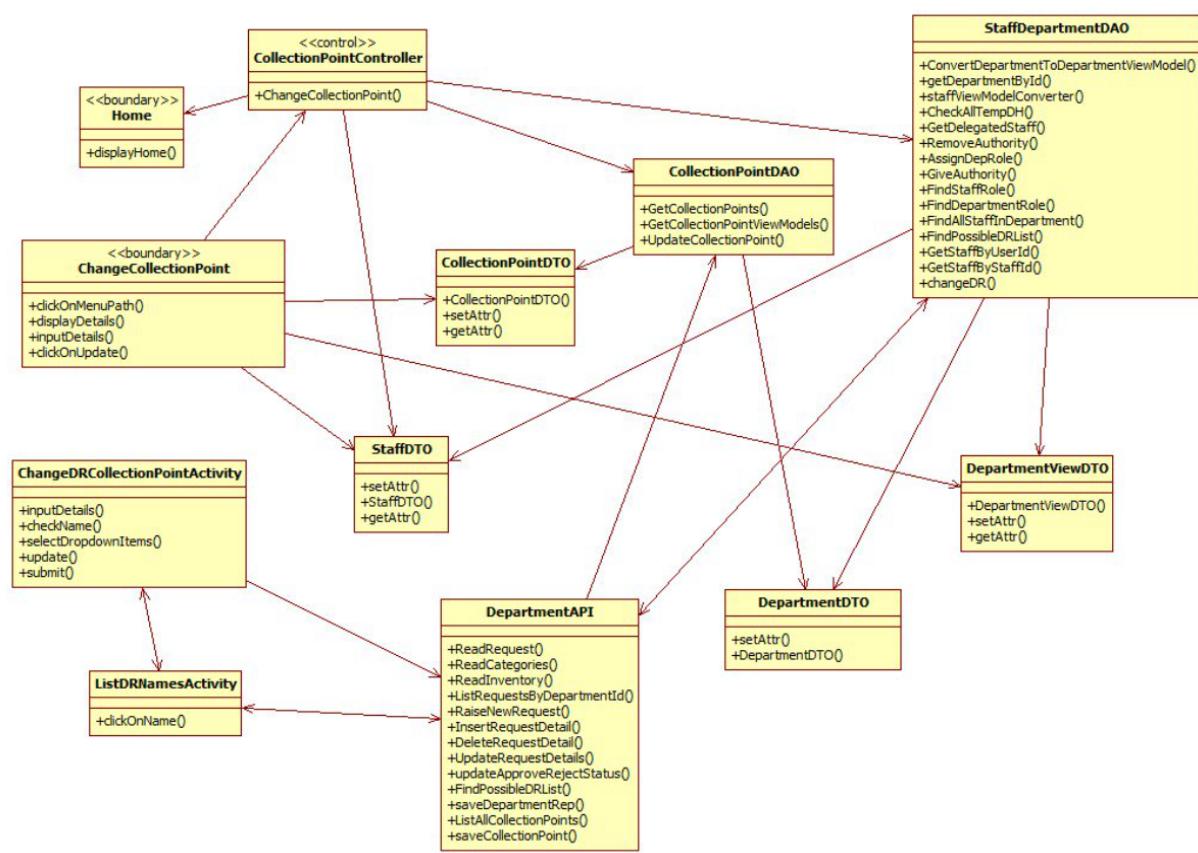
Mobile Application for Store



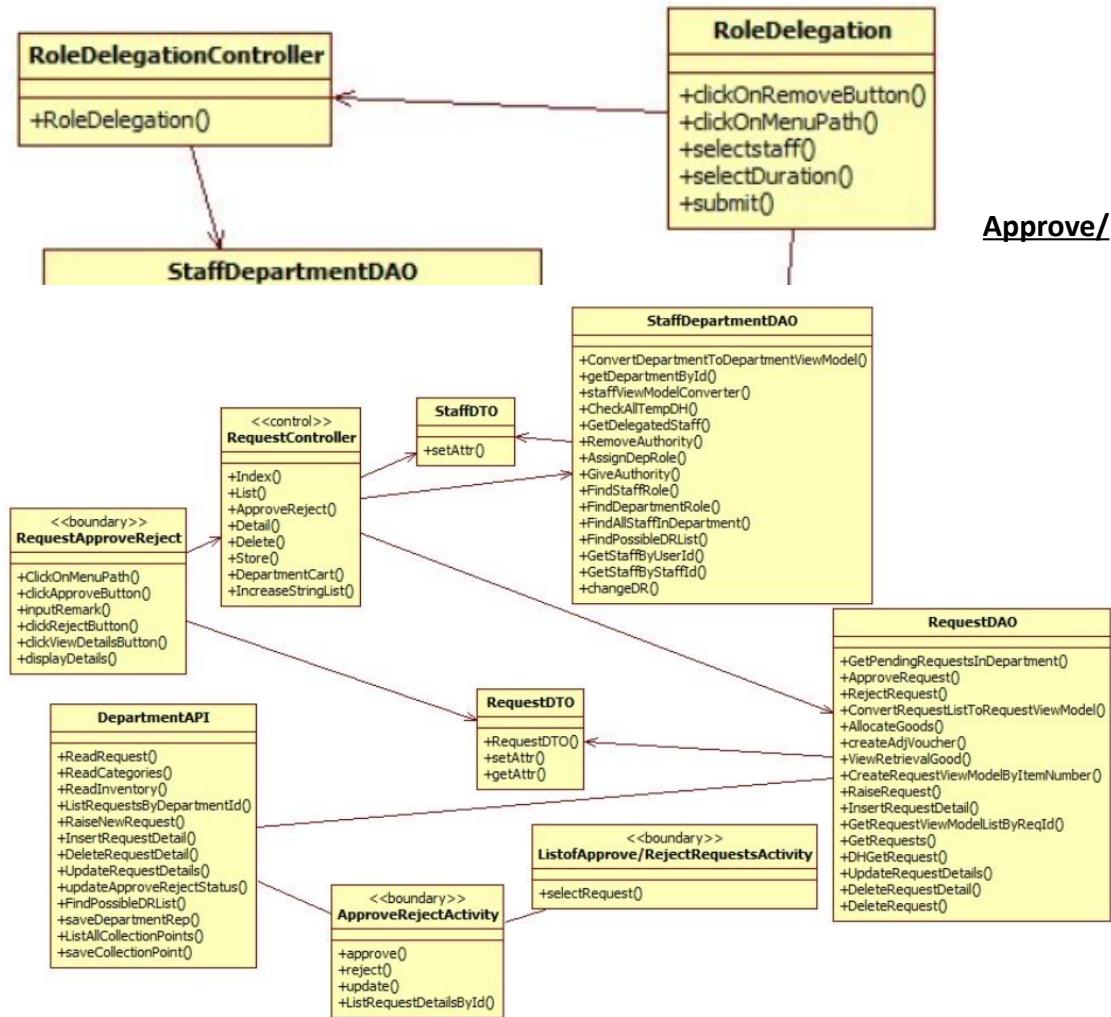
6. Entity Relationship Diagram

7. Class Diagram

Change Collection Point/Department Representative for Department

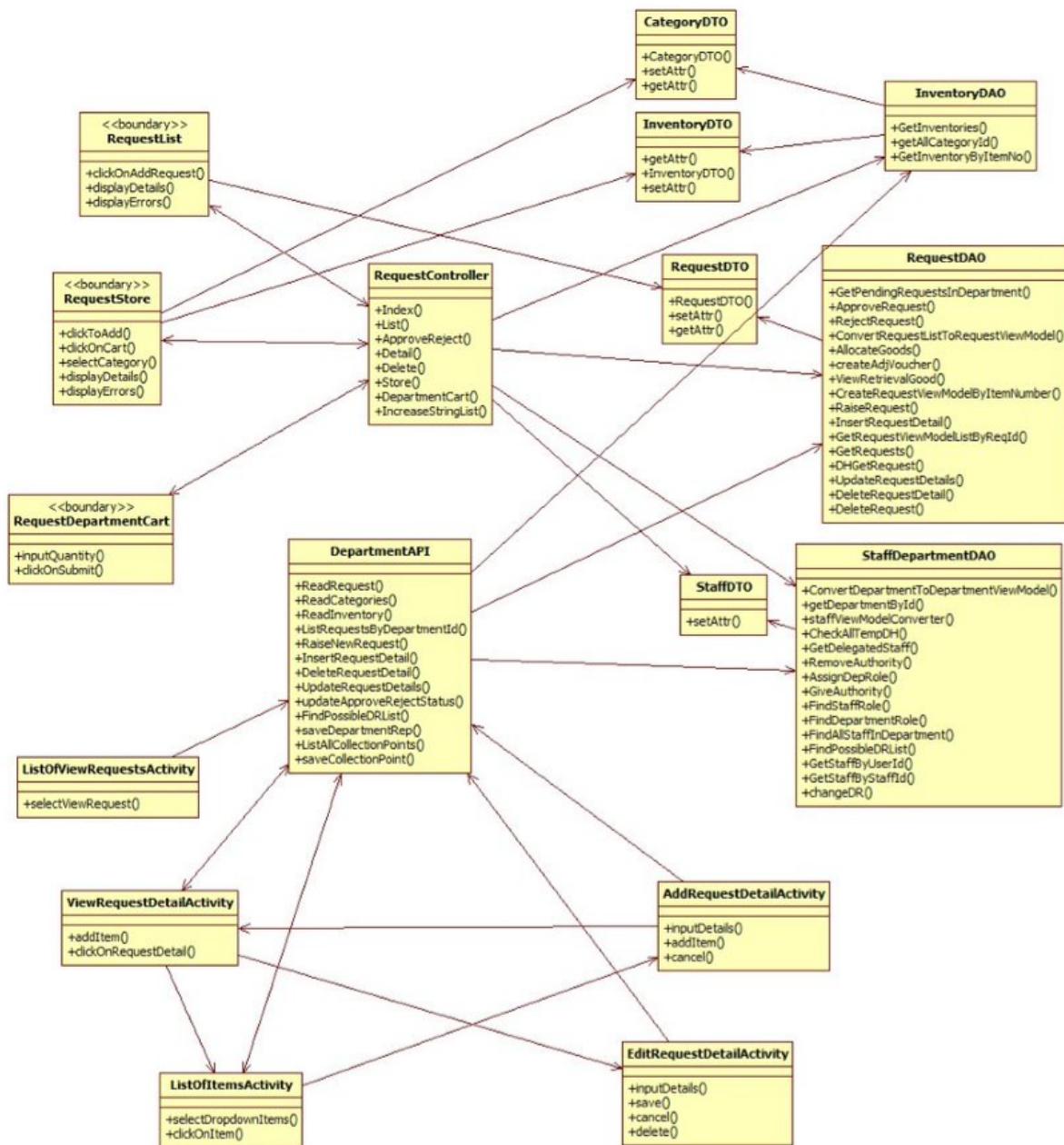


Role Delegation for Department

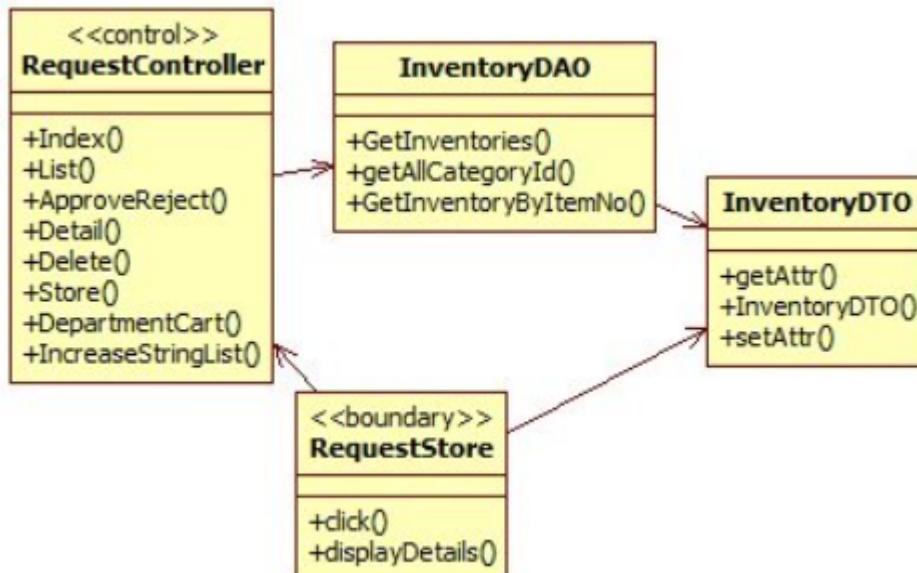


Reject for Department

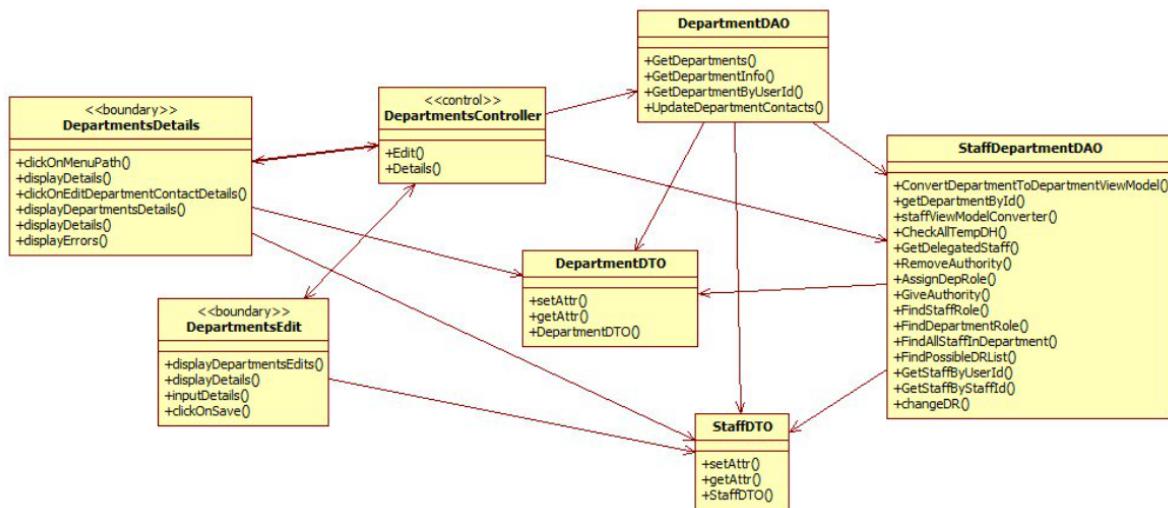
Request for Department



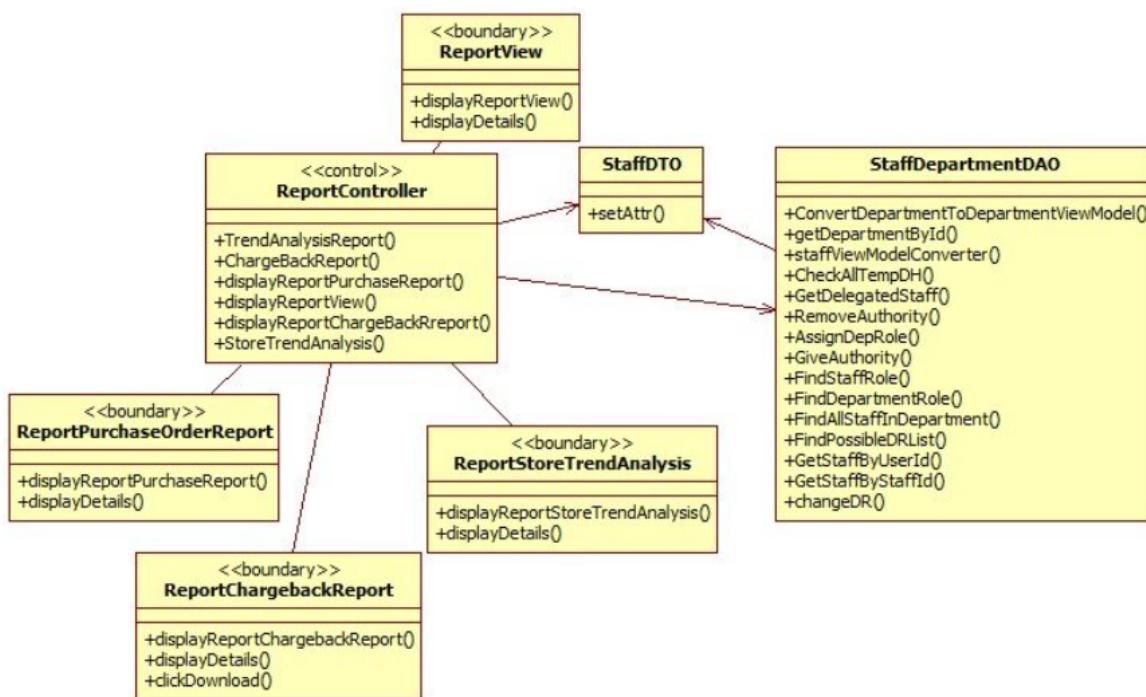
View Stationery Catalogue for Department



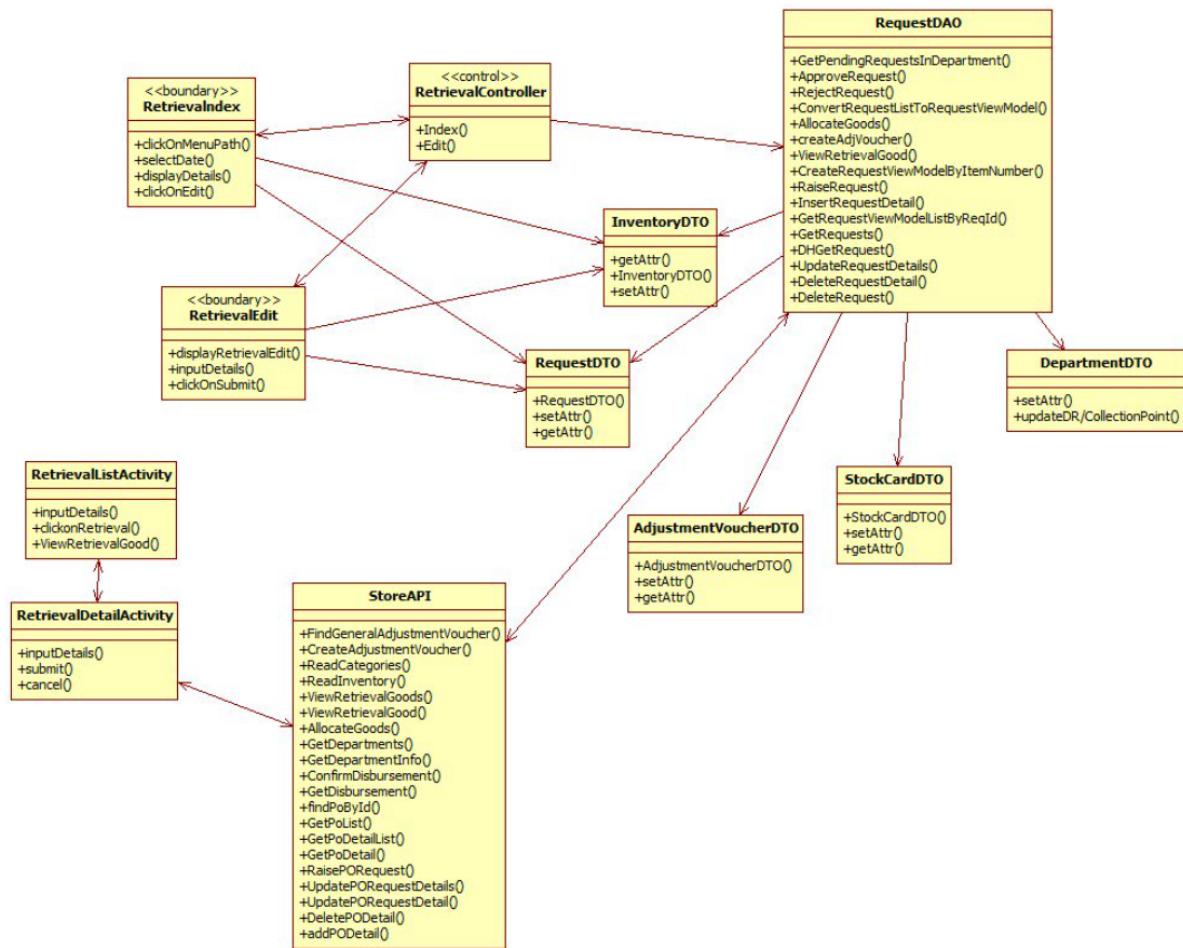
Modify Department Information for Department



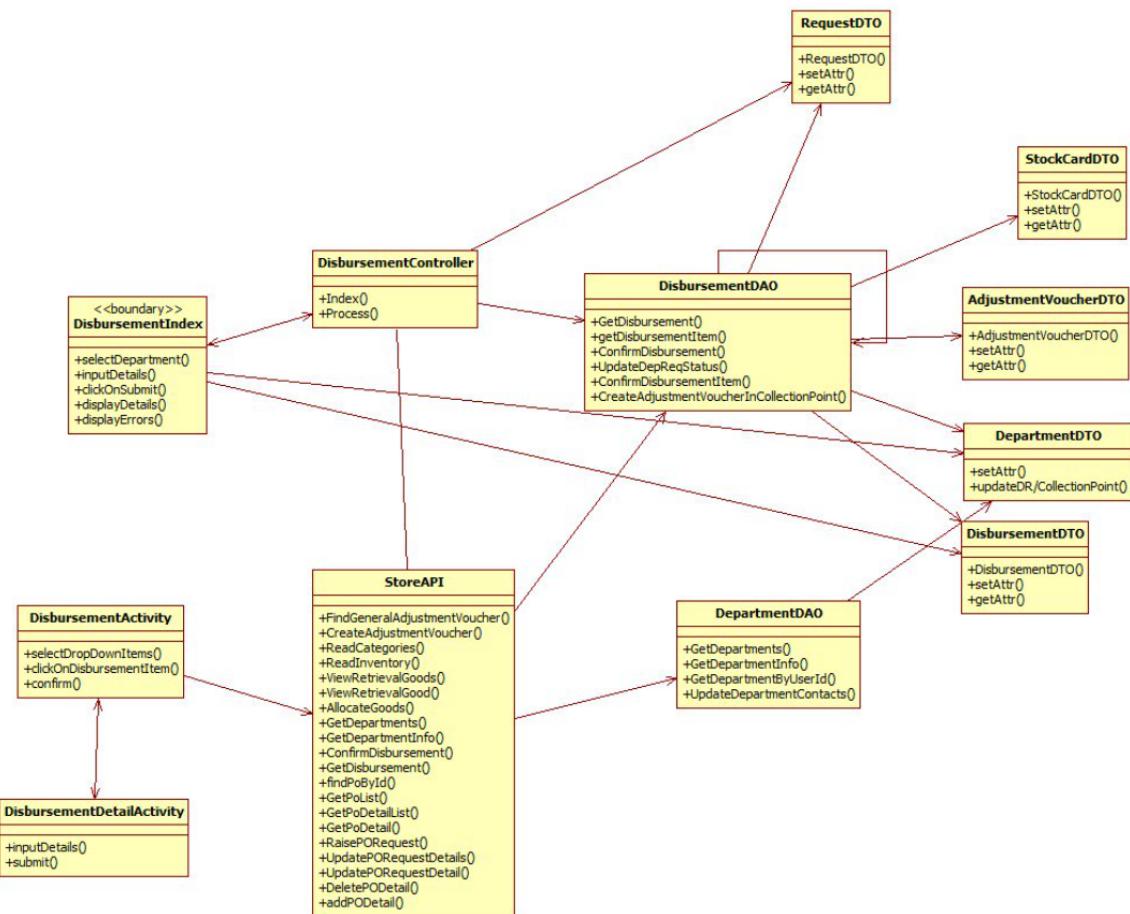
View Reports for Department and Store



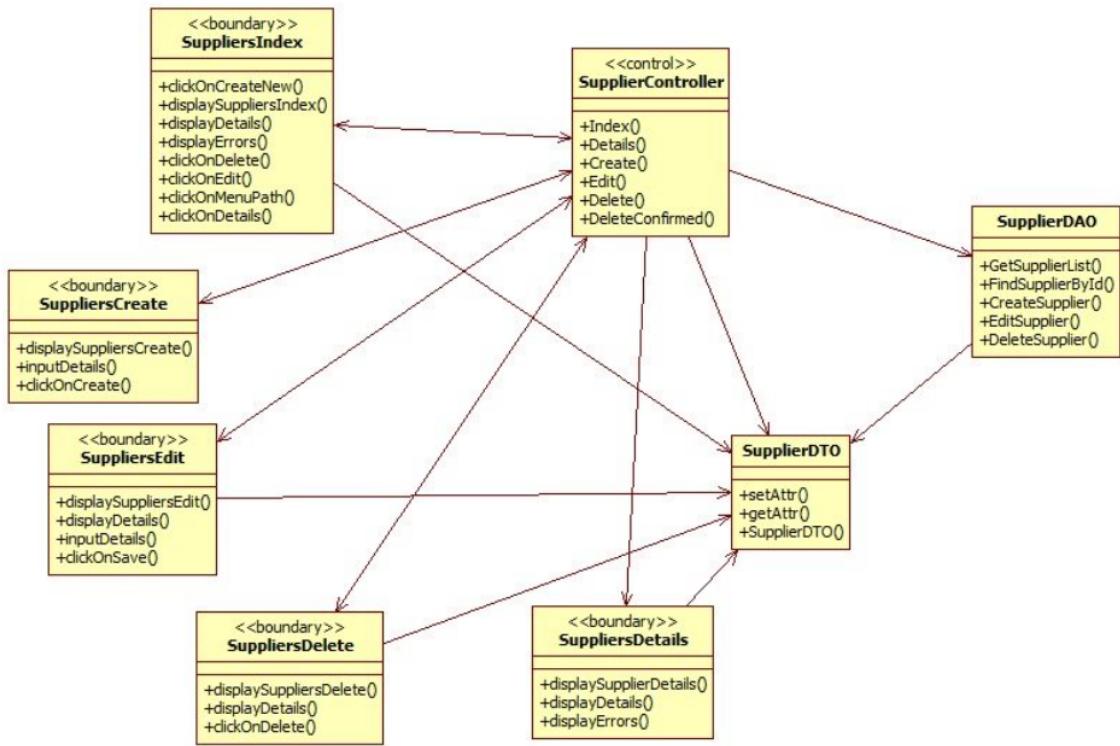
Retrieval List for Store



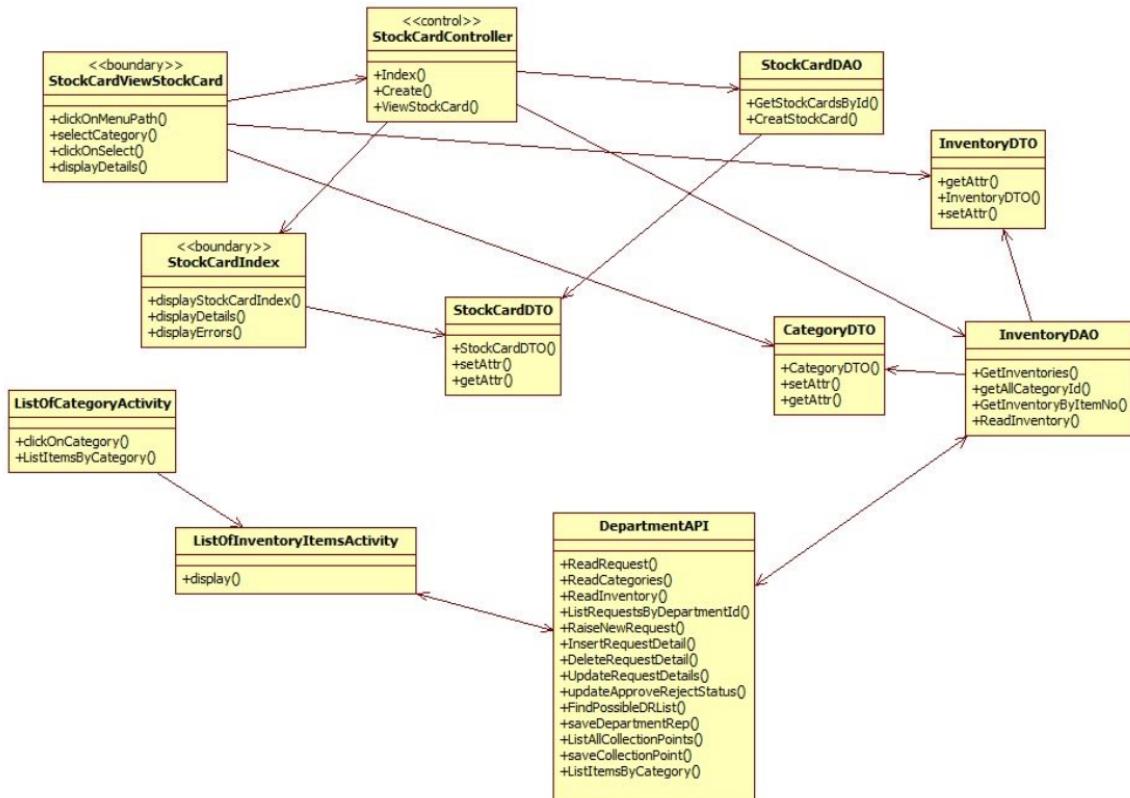
Disbursement List for Store



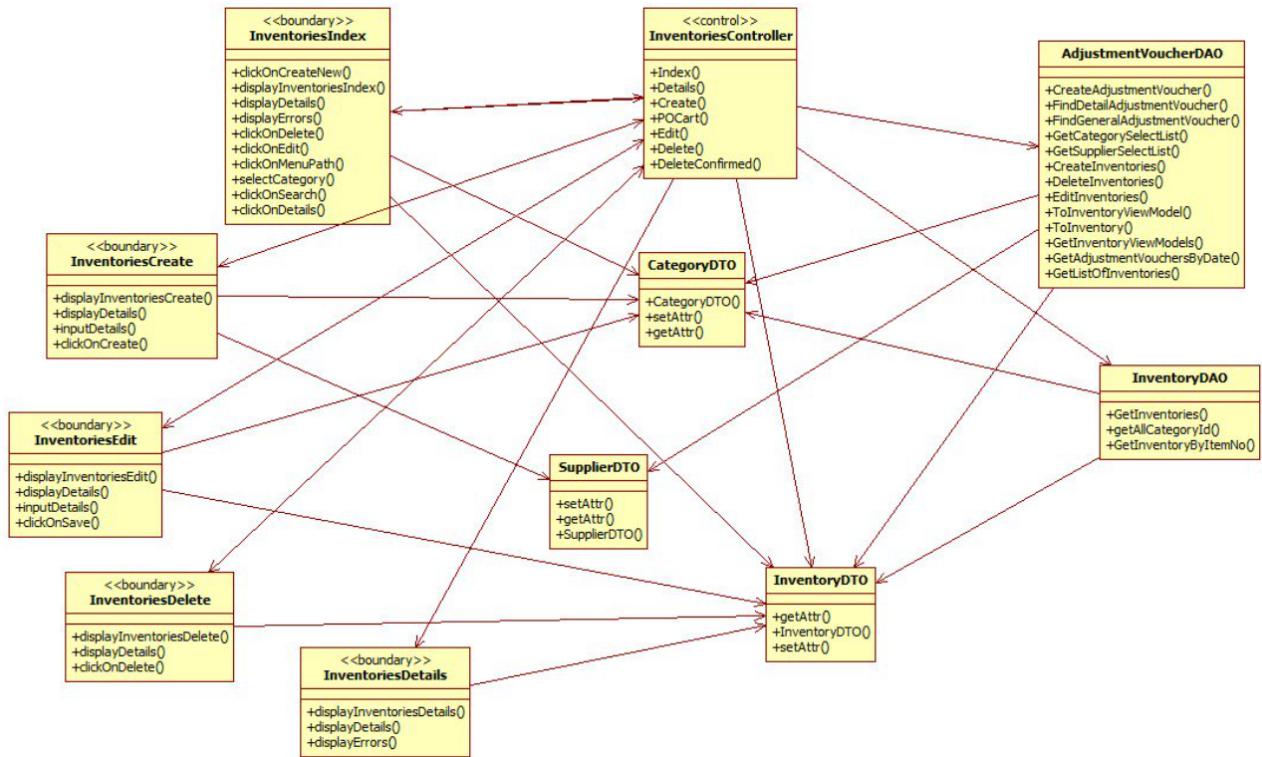
Modify Supplier Information for Store



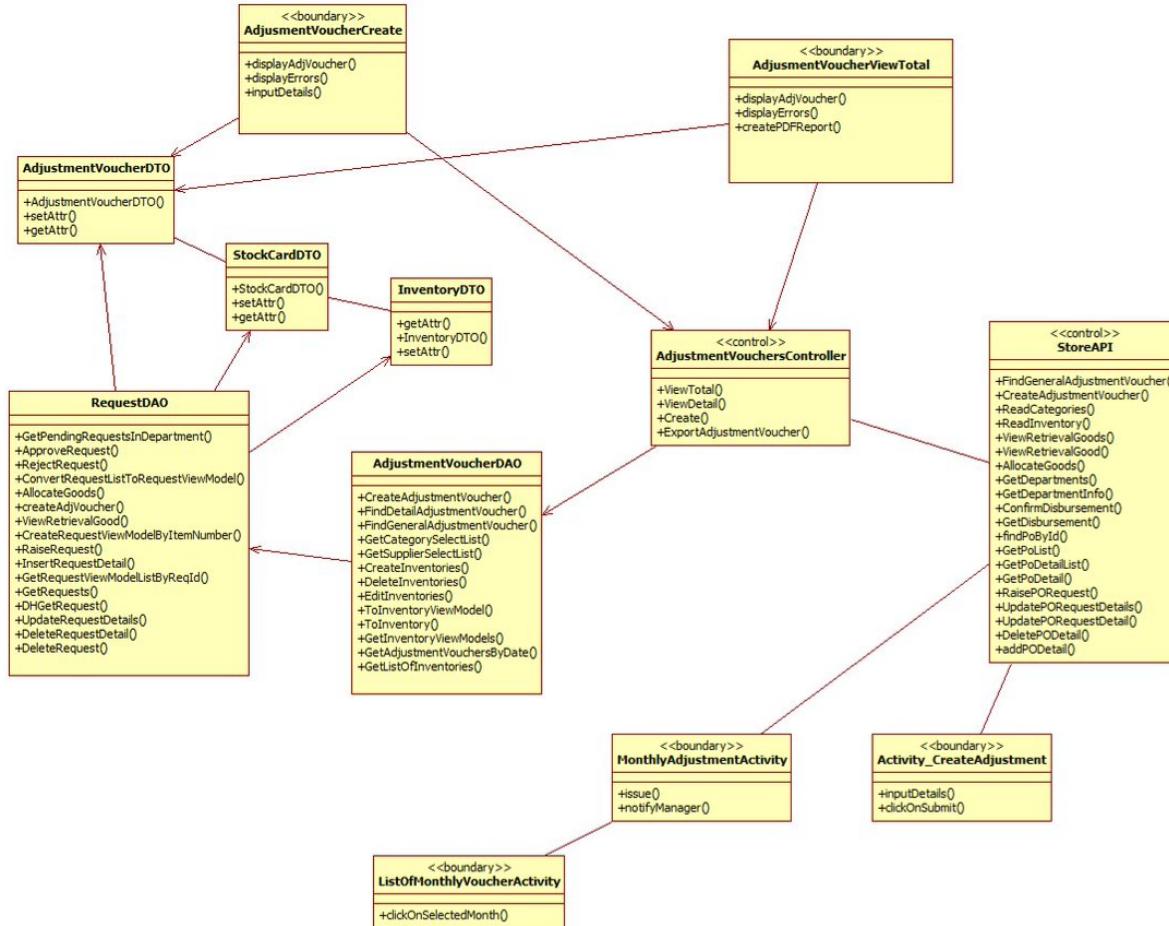
View Stock Card for Store



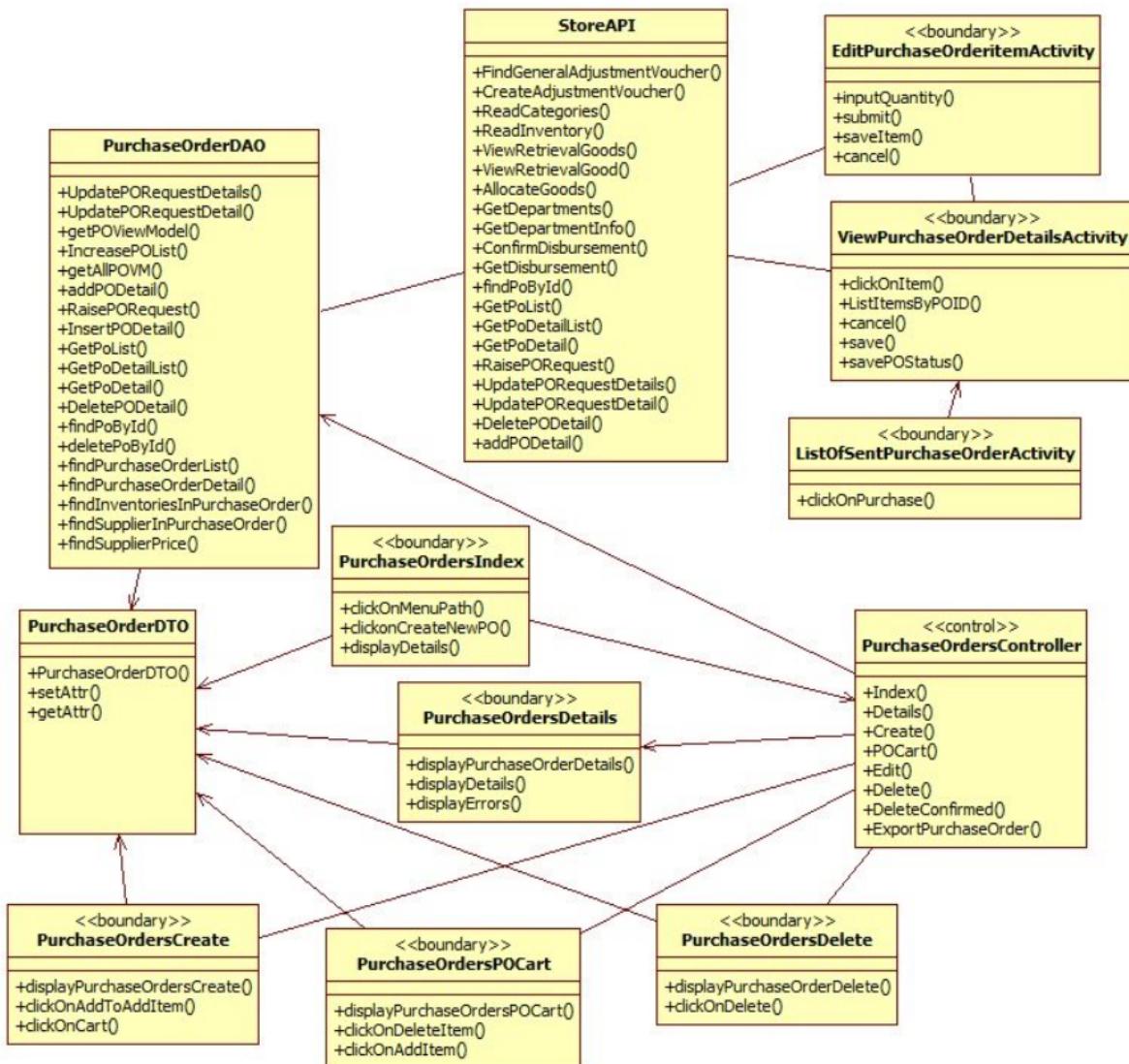
Modify Stationery Catalogue for Store



Process Adjustment Voucher for Store



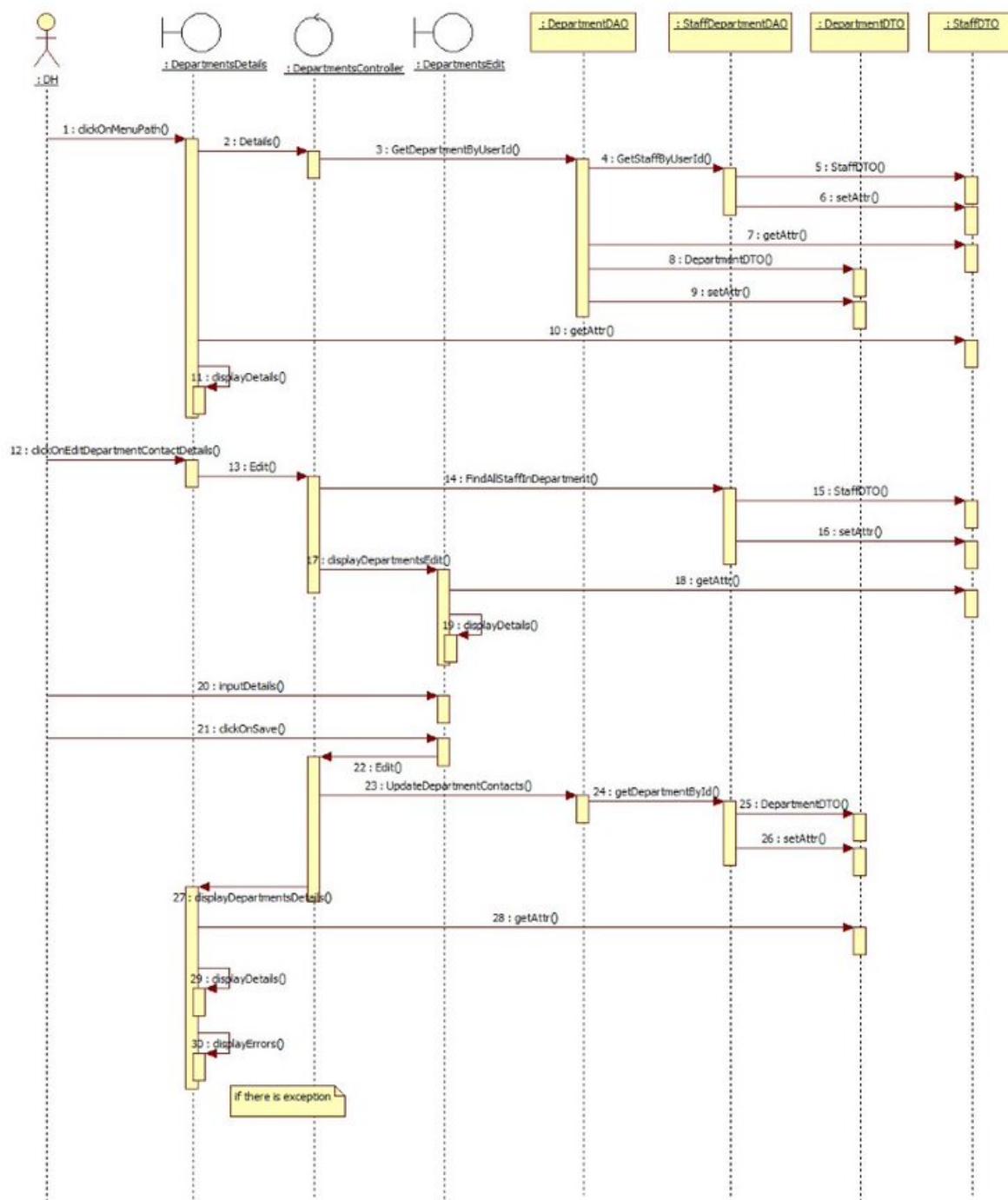
Process Purchase Order for Store



8. Sequence Diagram

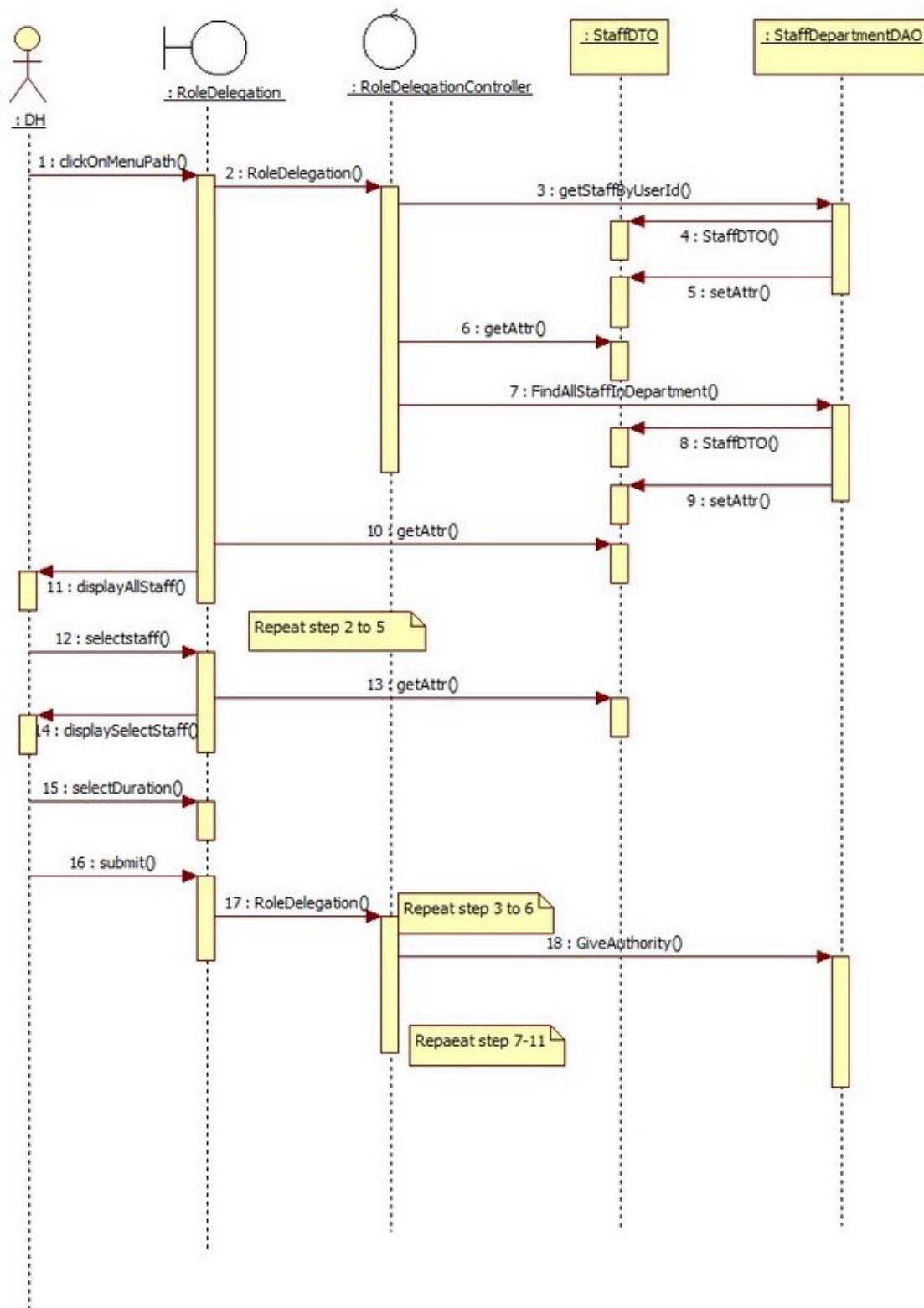
8.1 Web Application for Department Sequence Diagram

Modify Department Information

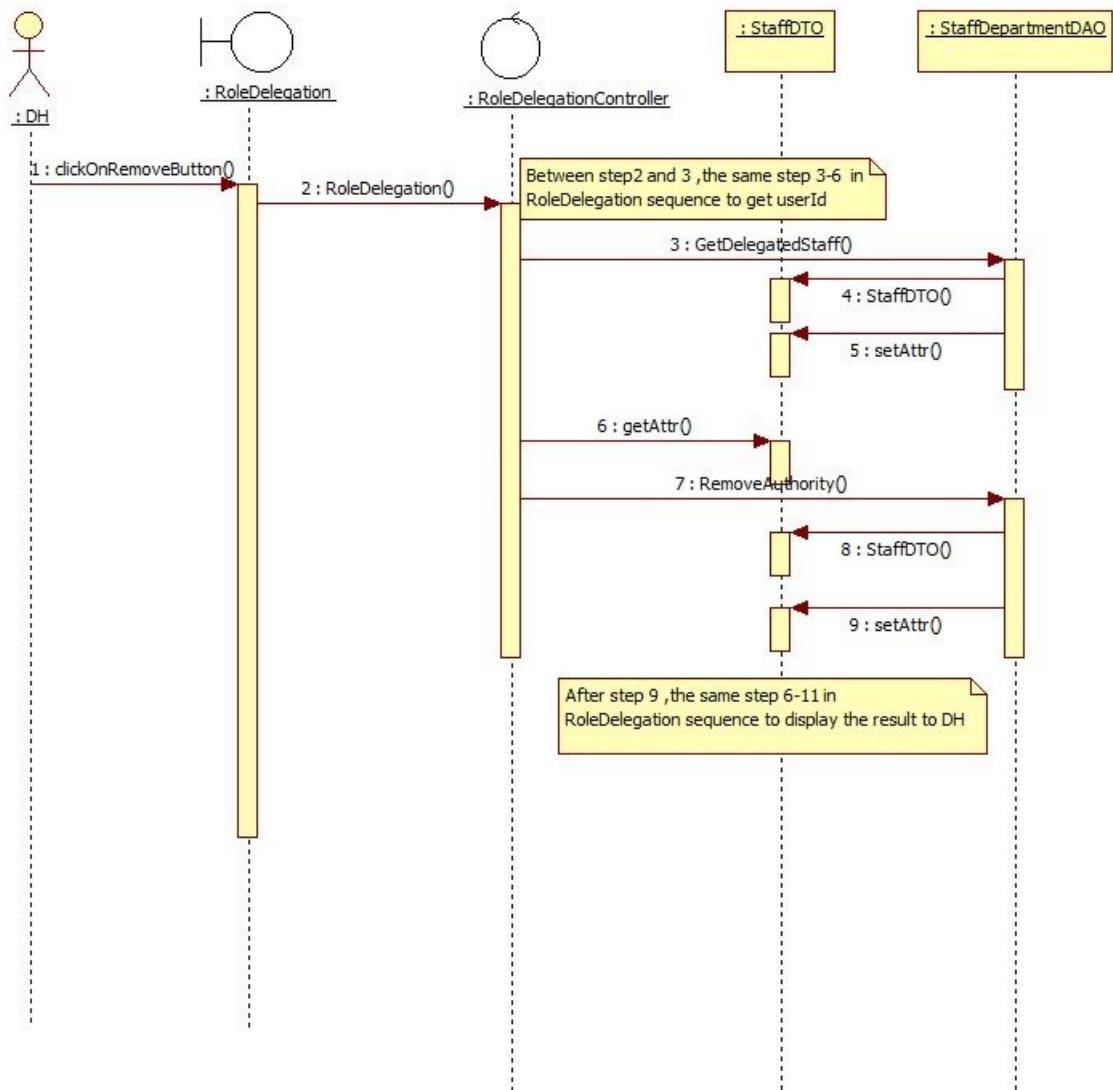


Delegate DH Duties

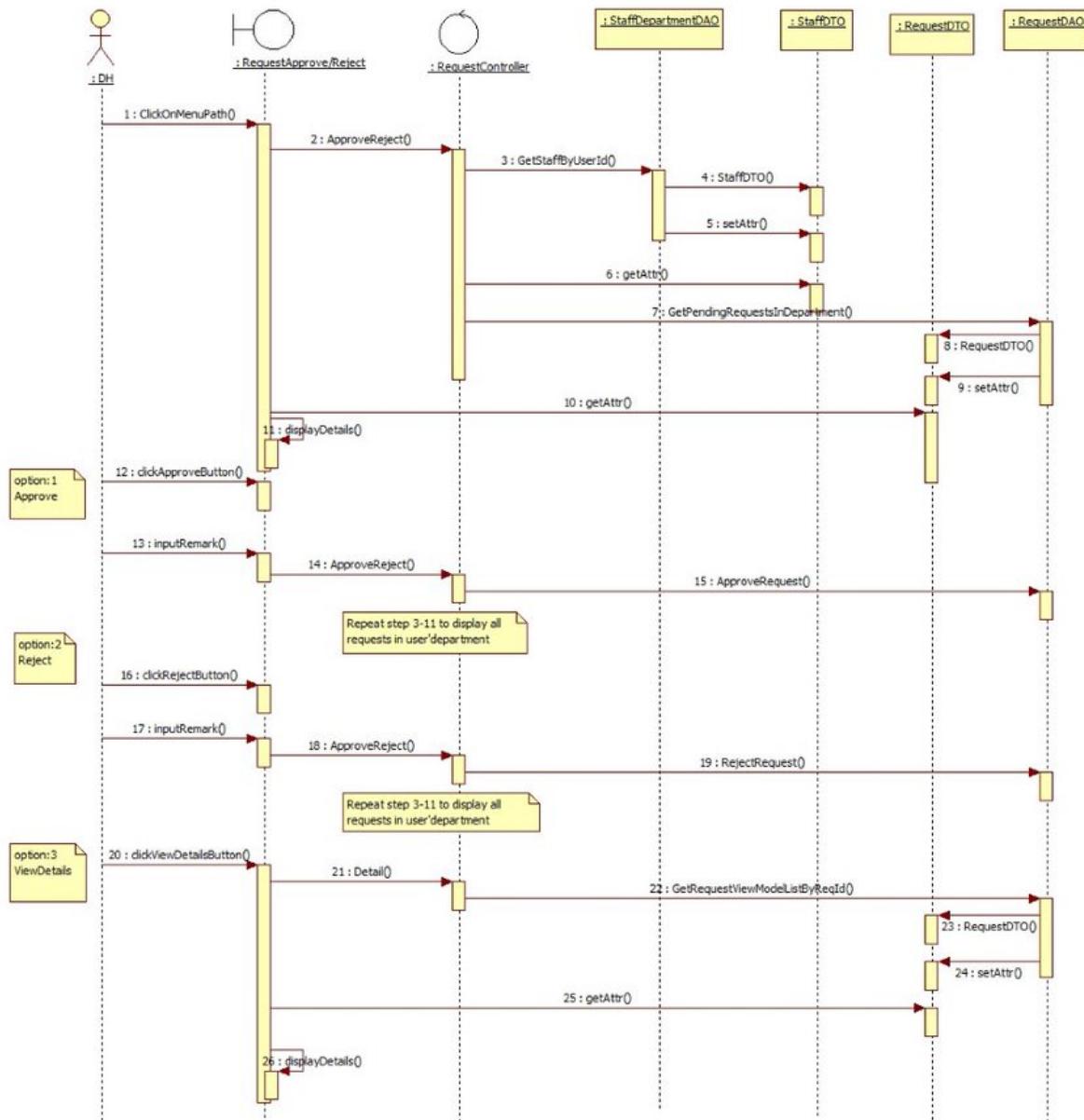
1. Role Delegation



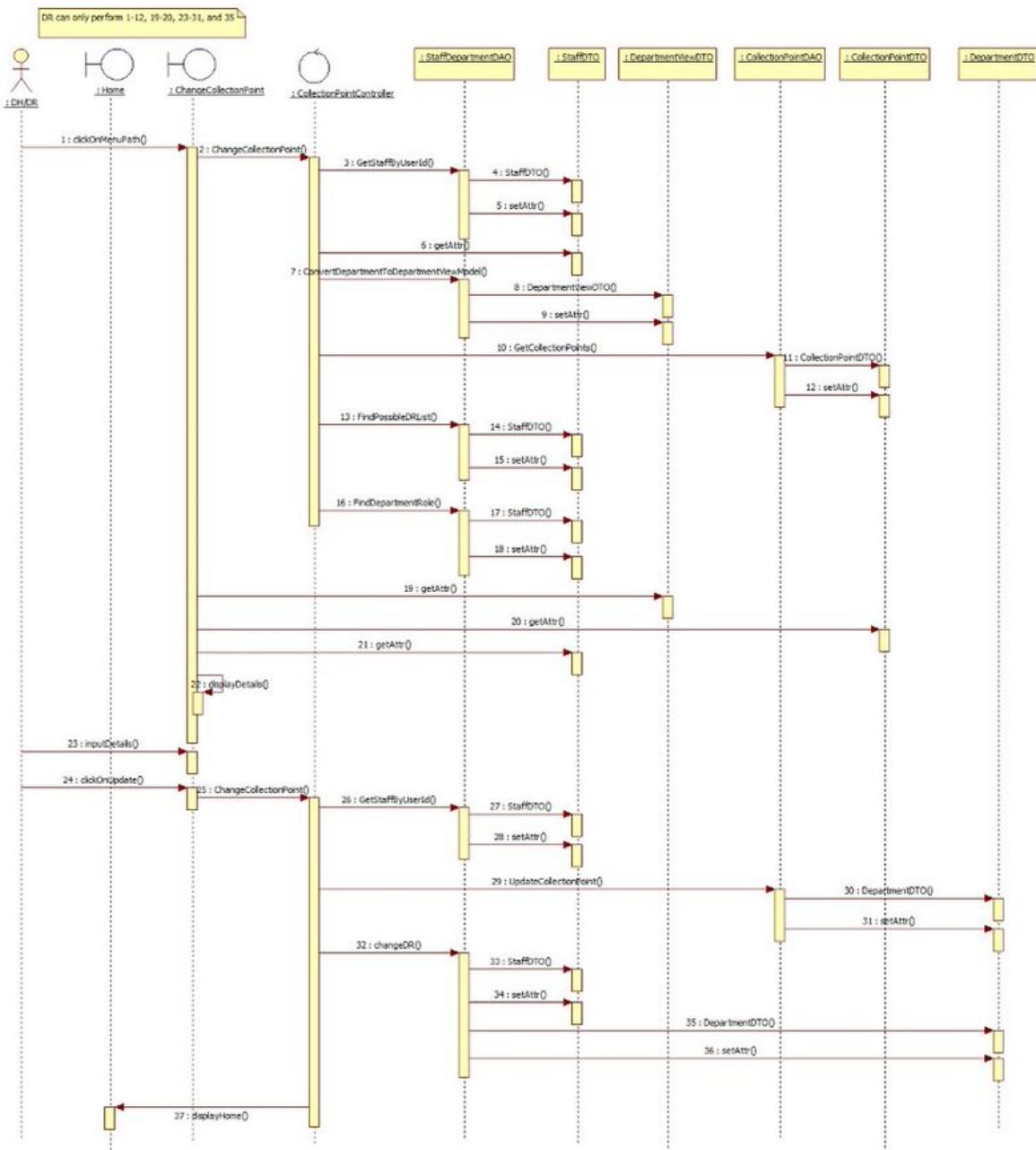
2.Remove Delegation



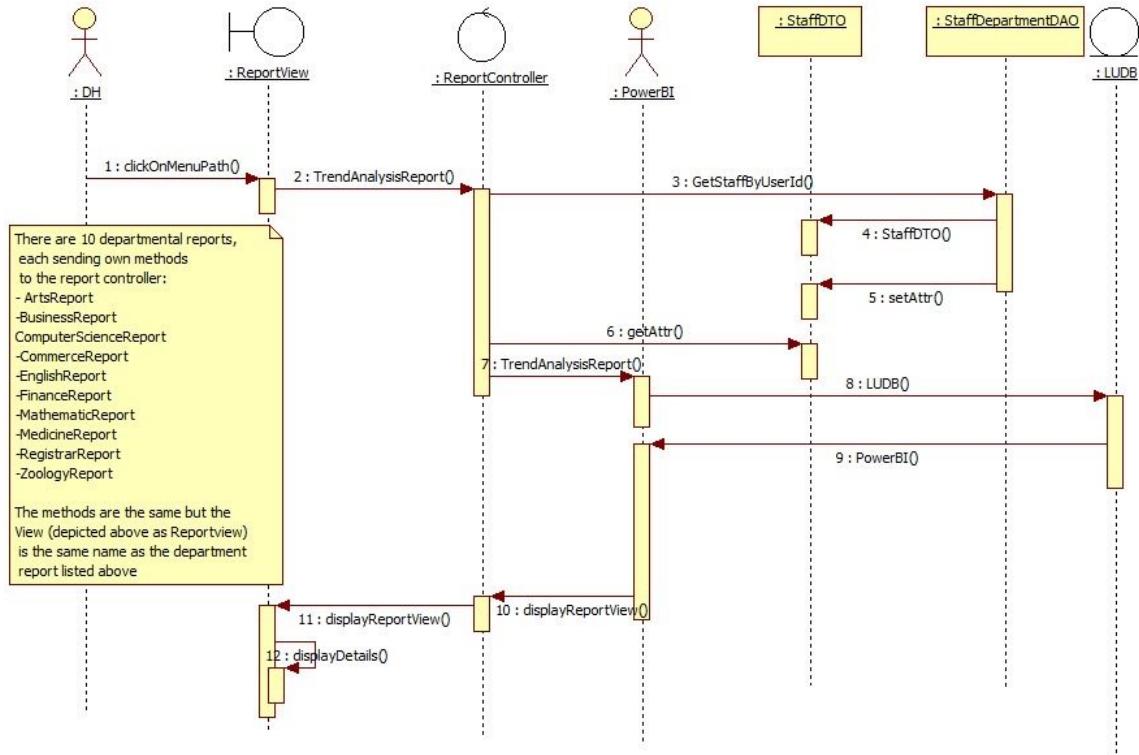
Approve/Reject Request



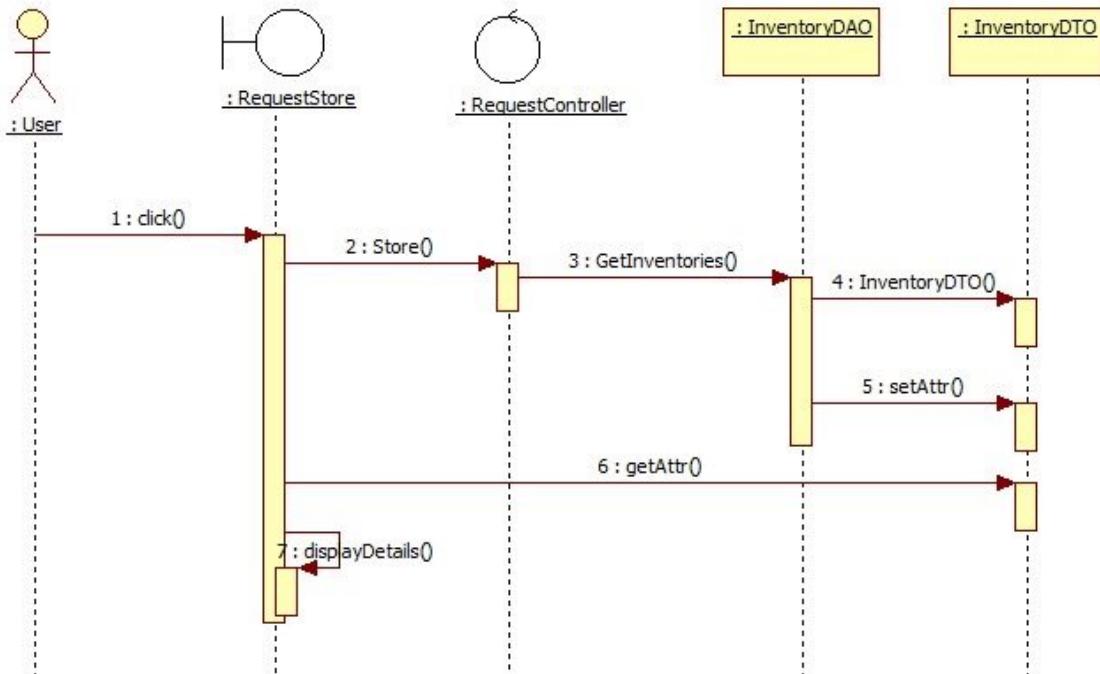
Change DR/Collection Point



View Dep Trend Analysis Report

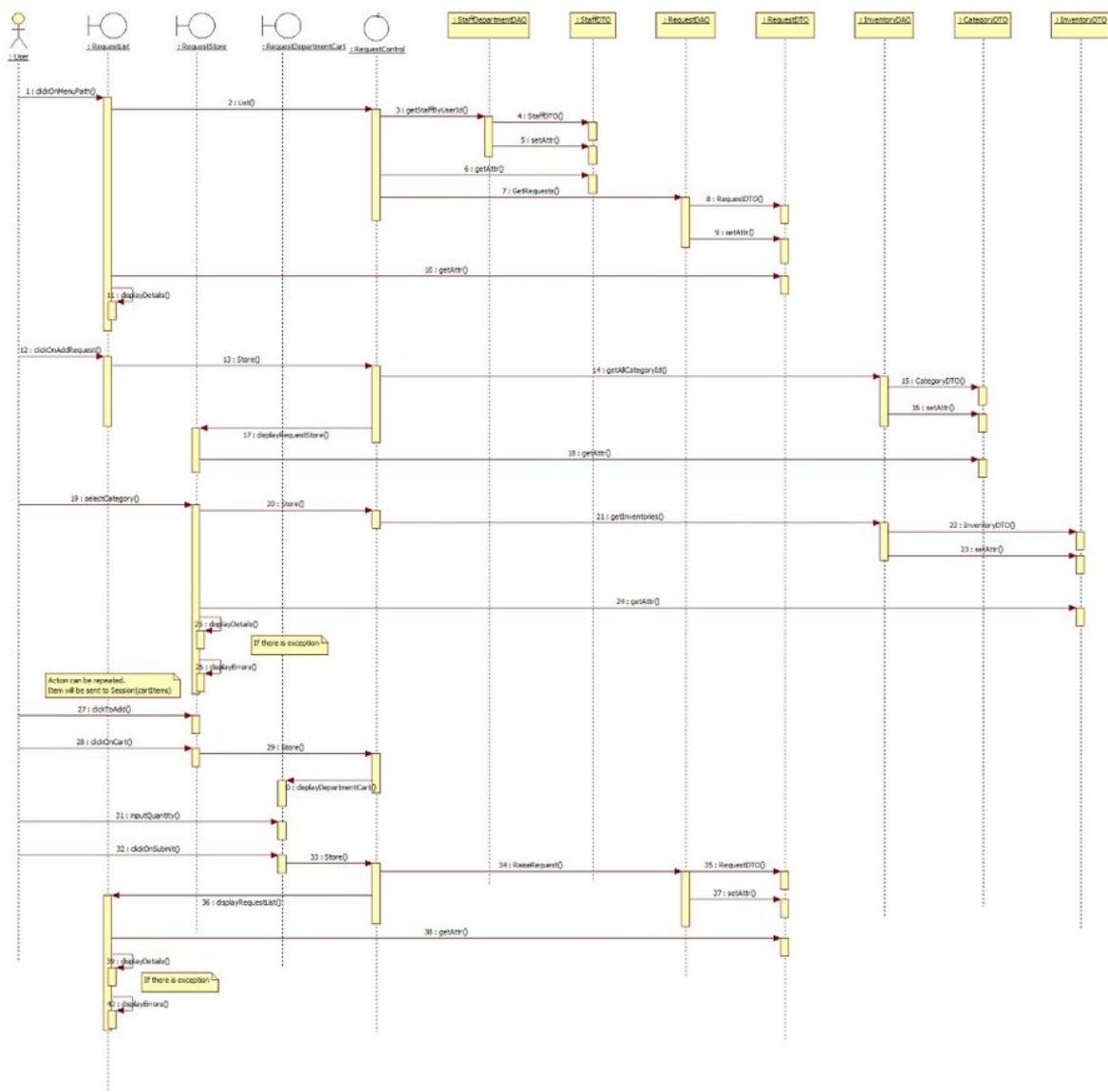


View Stationery Catalogue

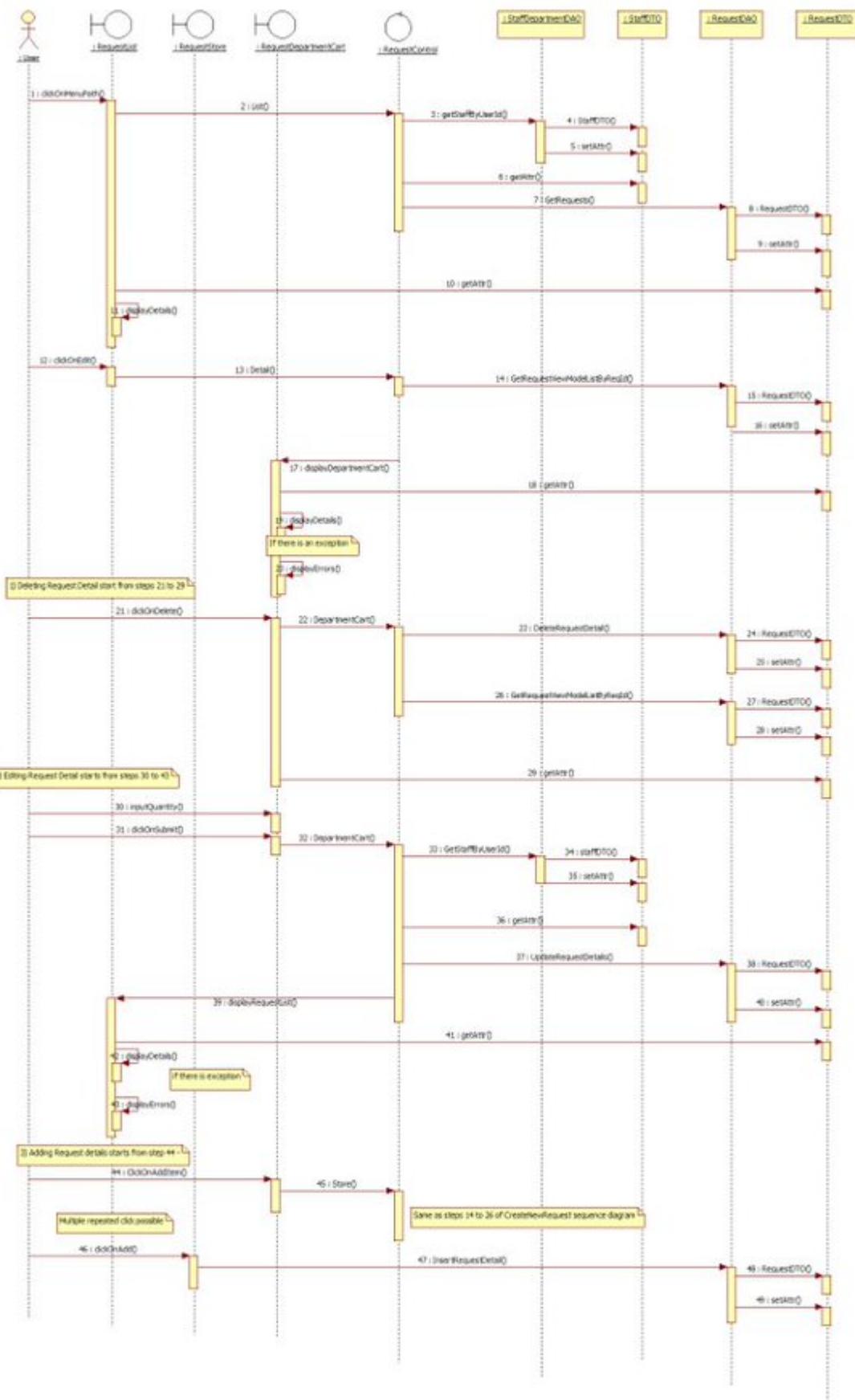


Raise Request

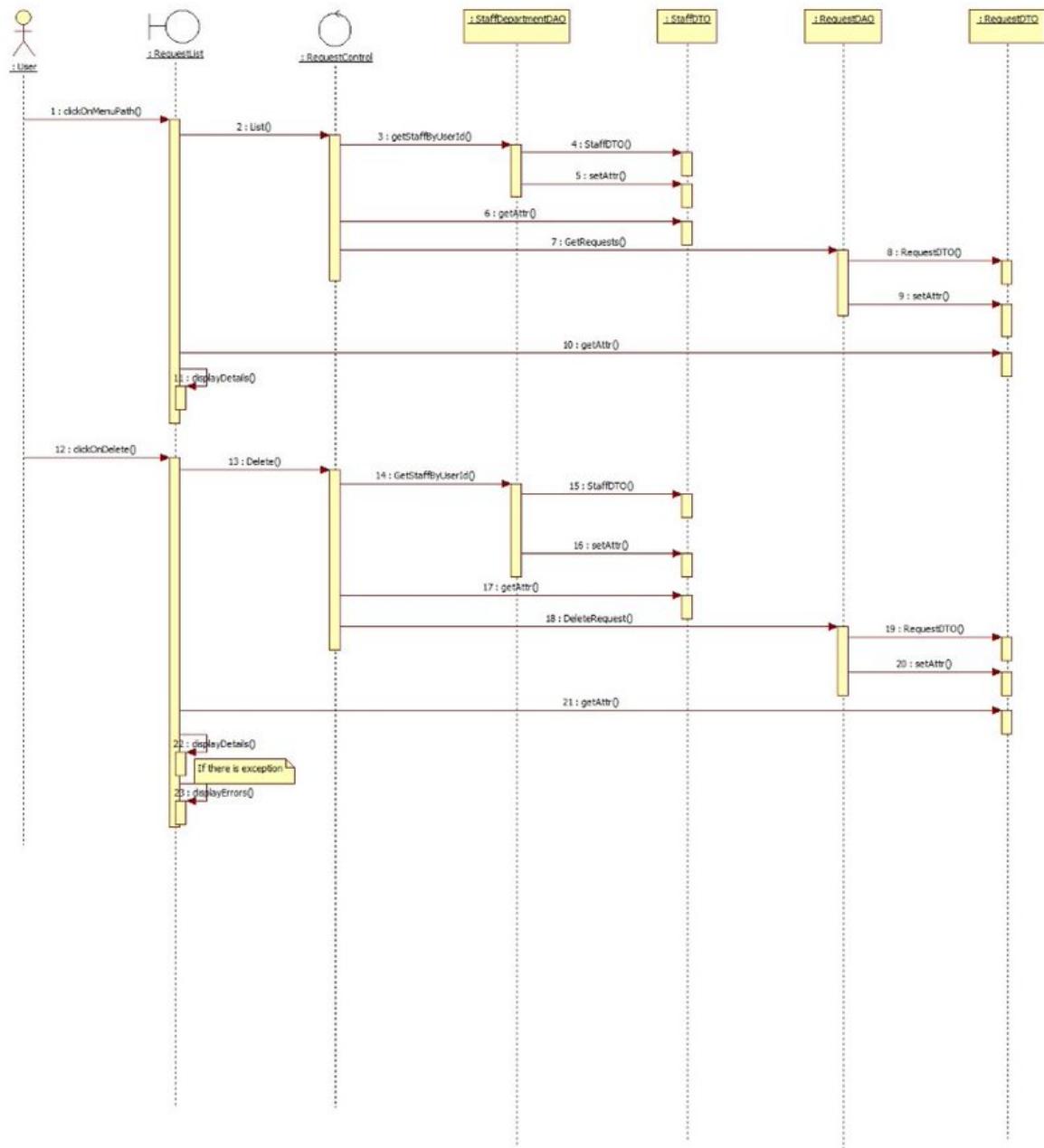
1.Raise Request



2.Modify Pending Request

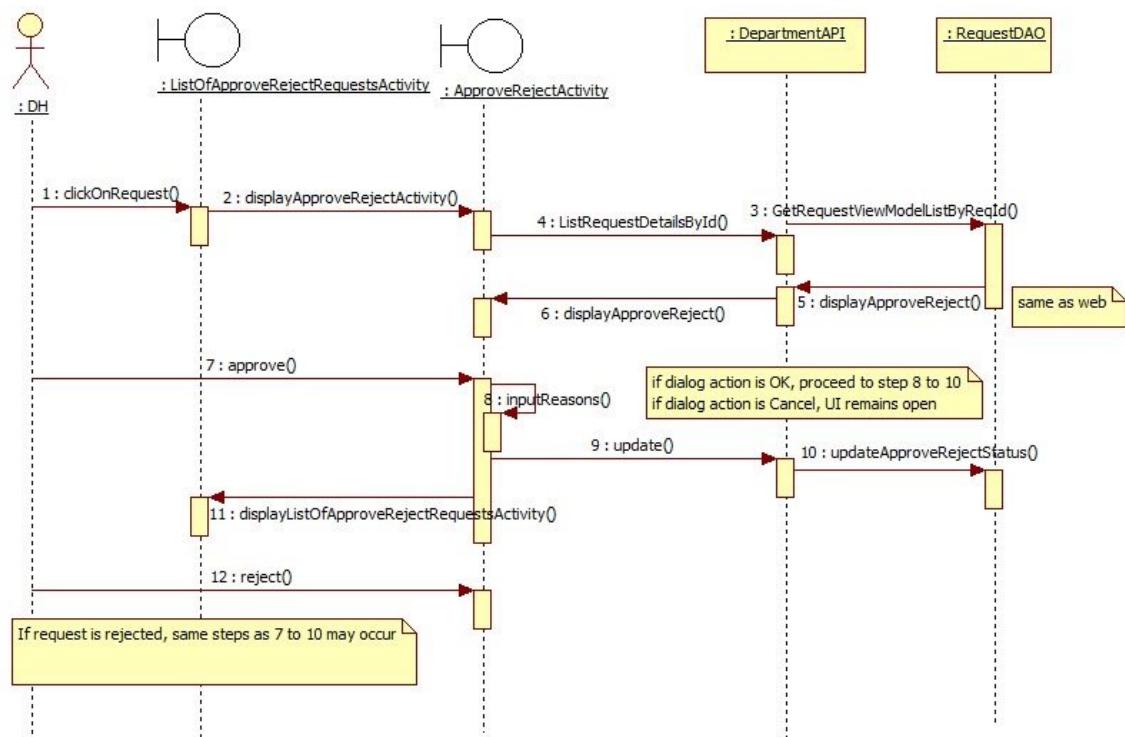


3.Delete Request

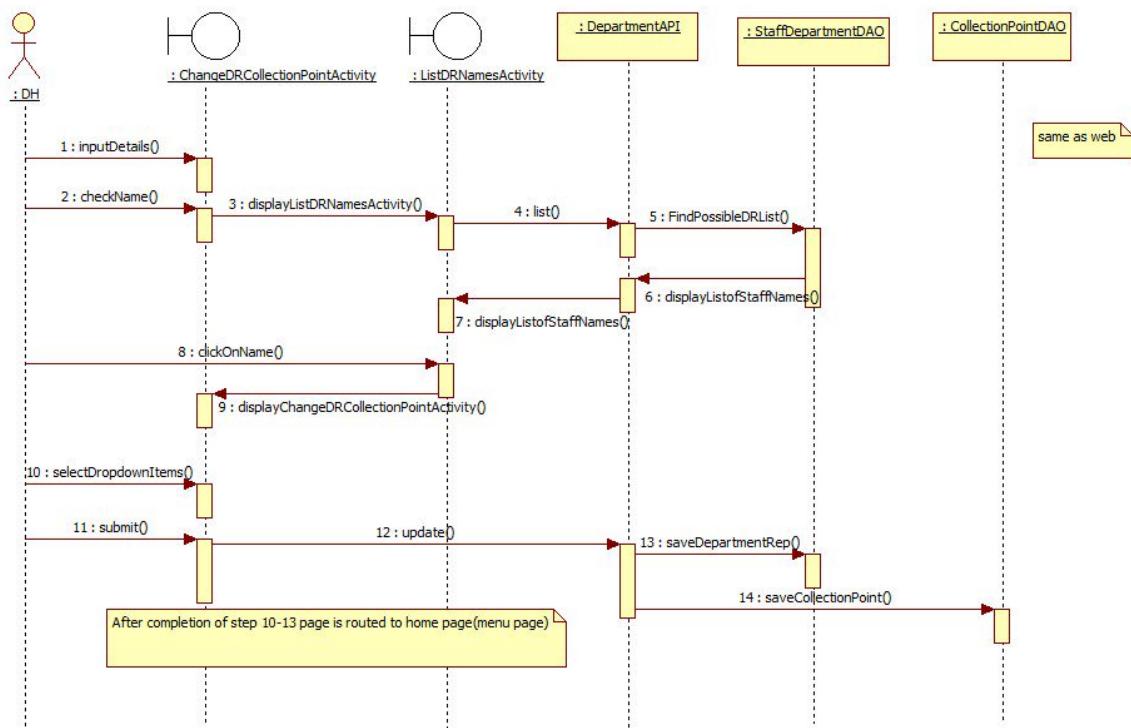


8.2 Mobile Application for Department Sequence Diagram

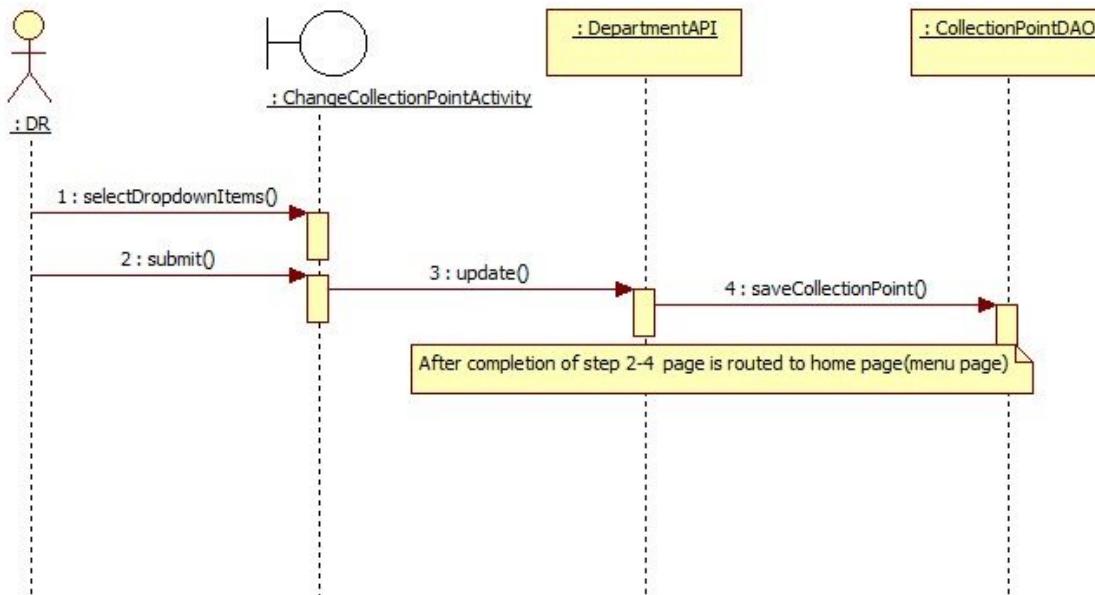
Approve/Reject Request



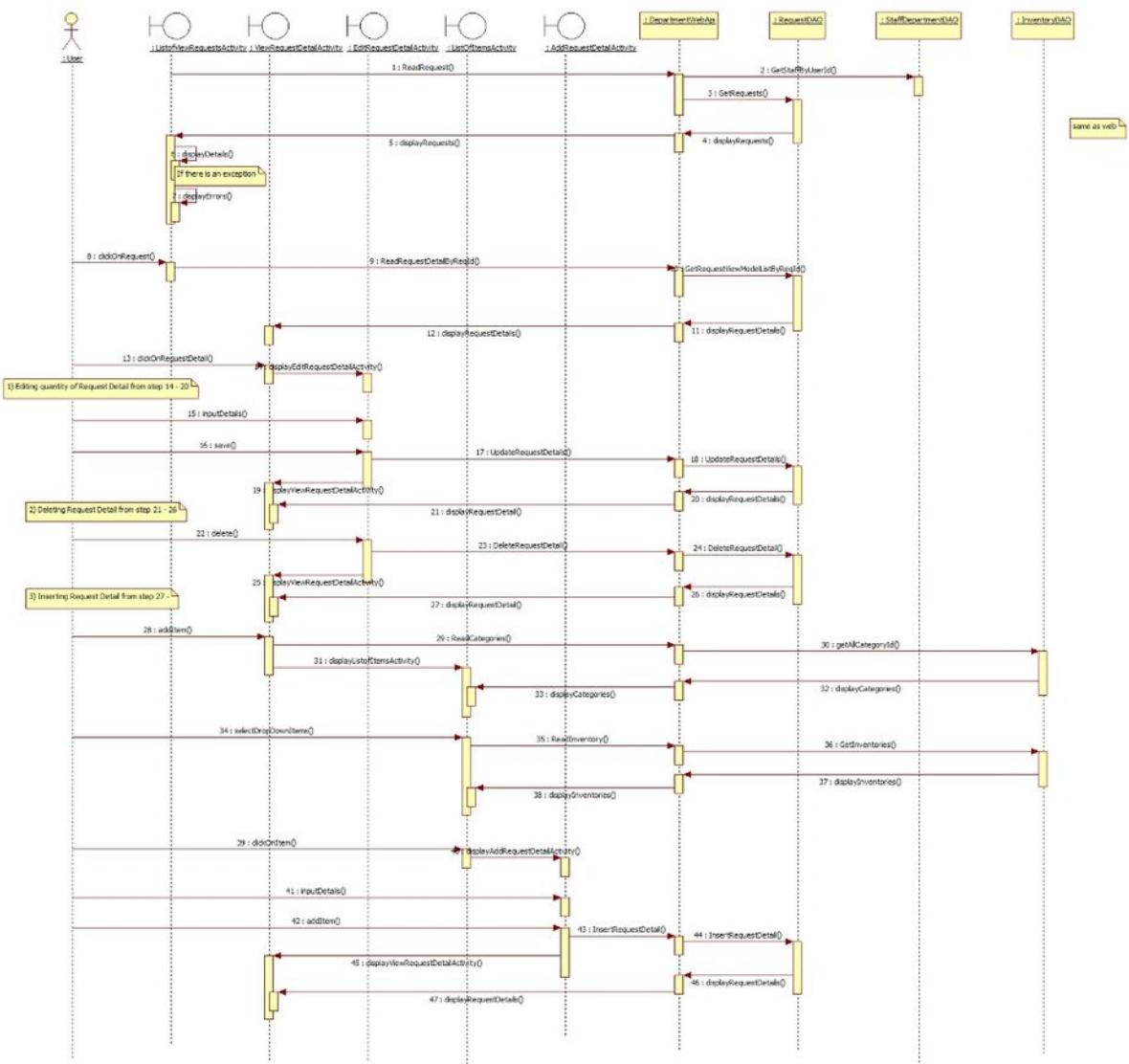
Change DR/Collection Point



Change Collection Point

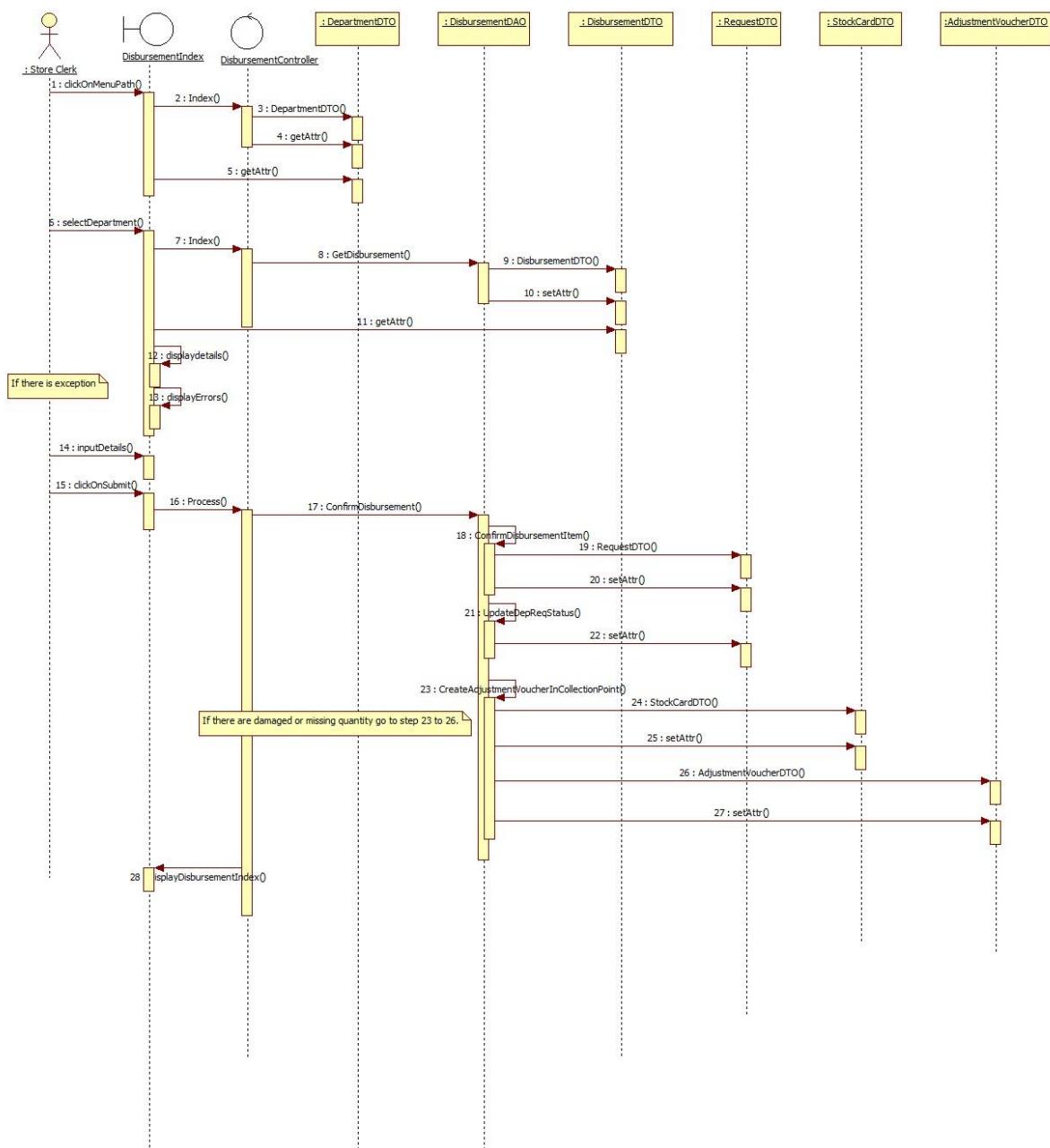


Modify Pending Request

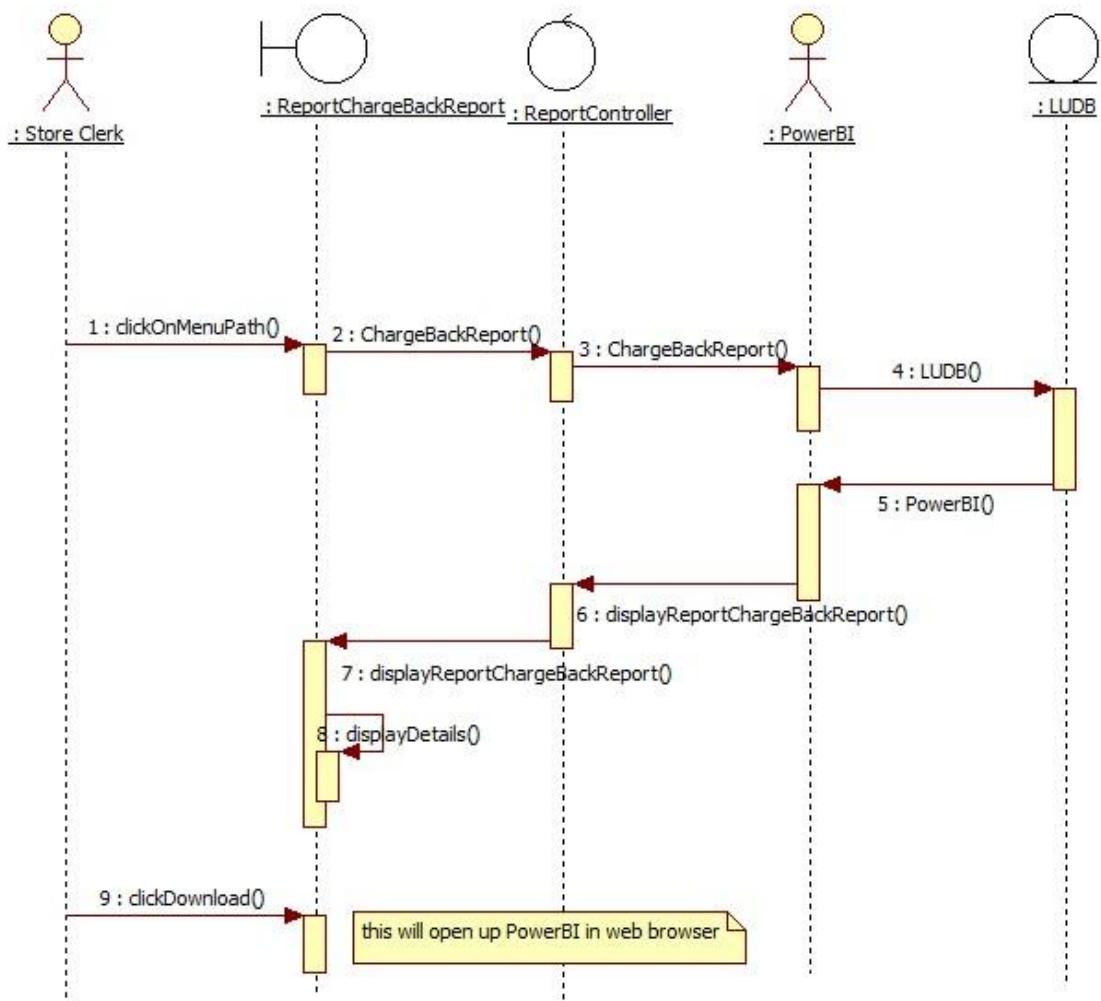


8.3 Web Application for Store Sequence Diagram

Process Disbursement List

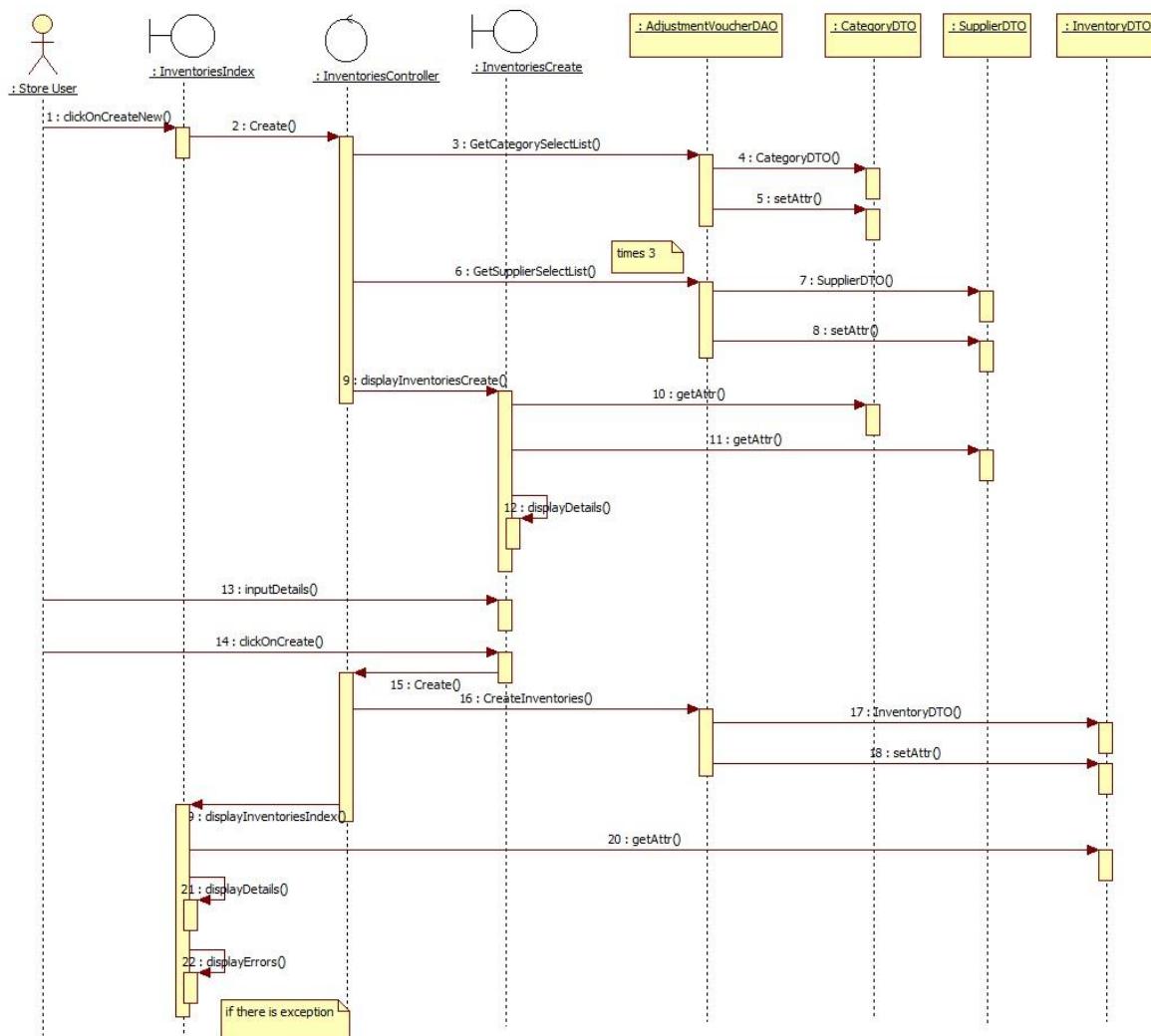


Generate Charge-Back Report

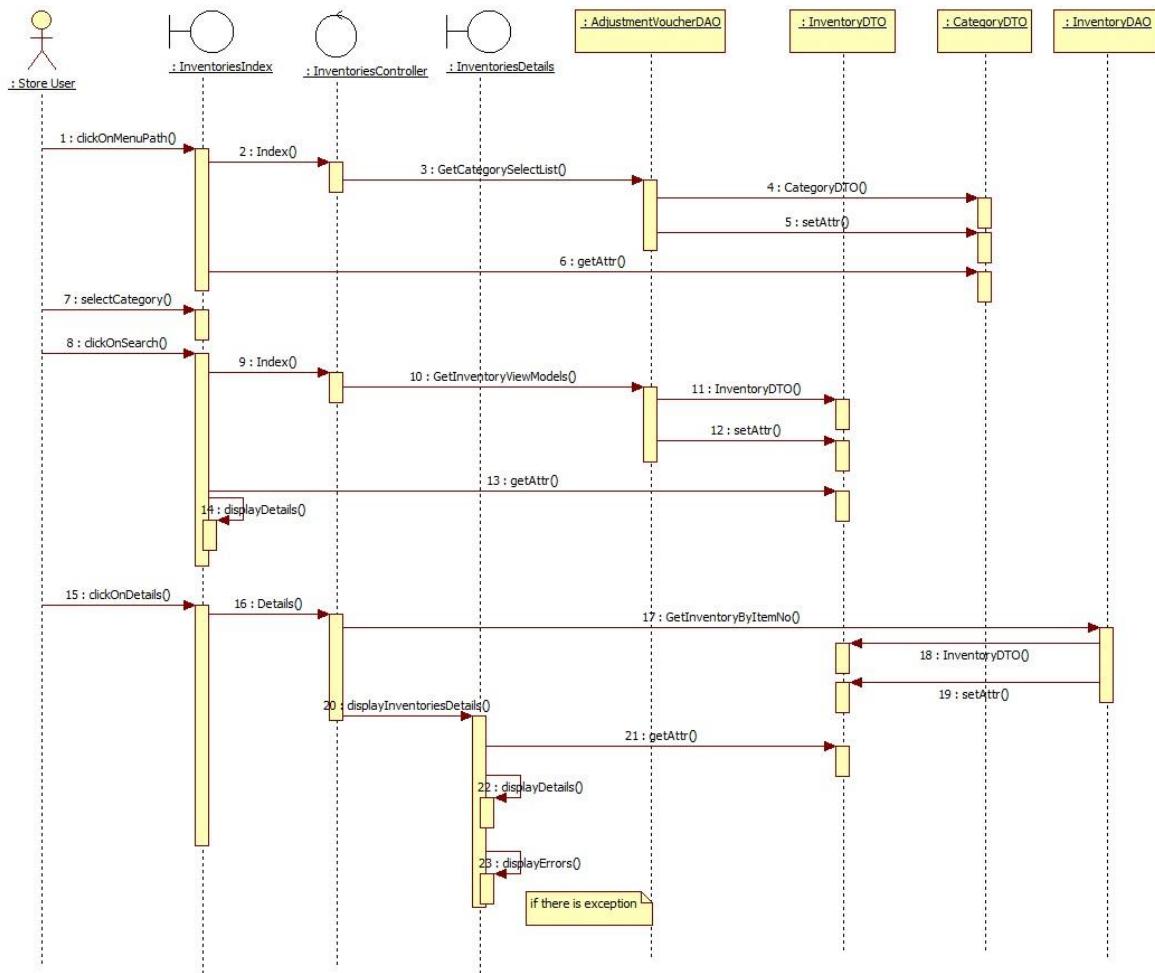


Modify Stationery Catalogue

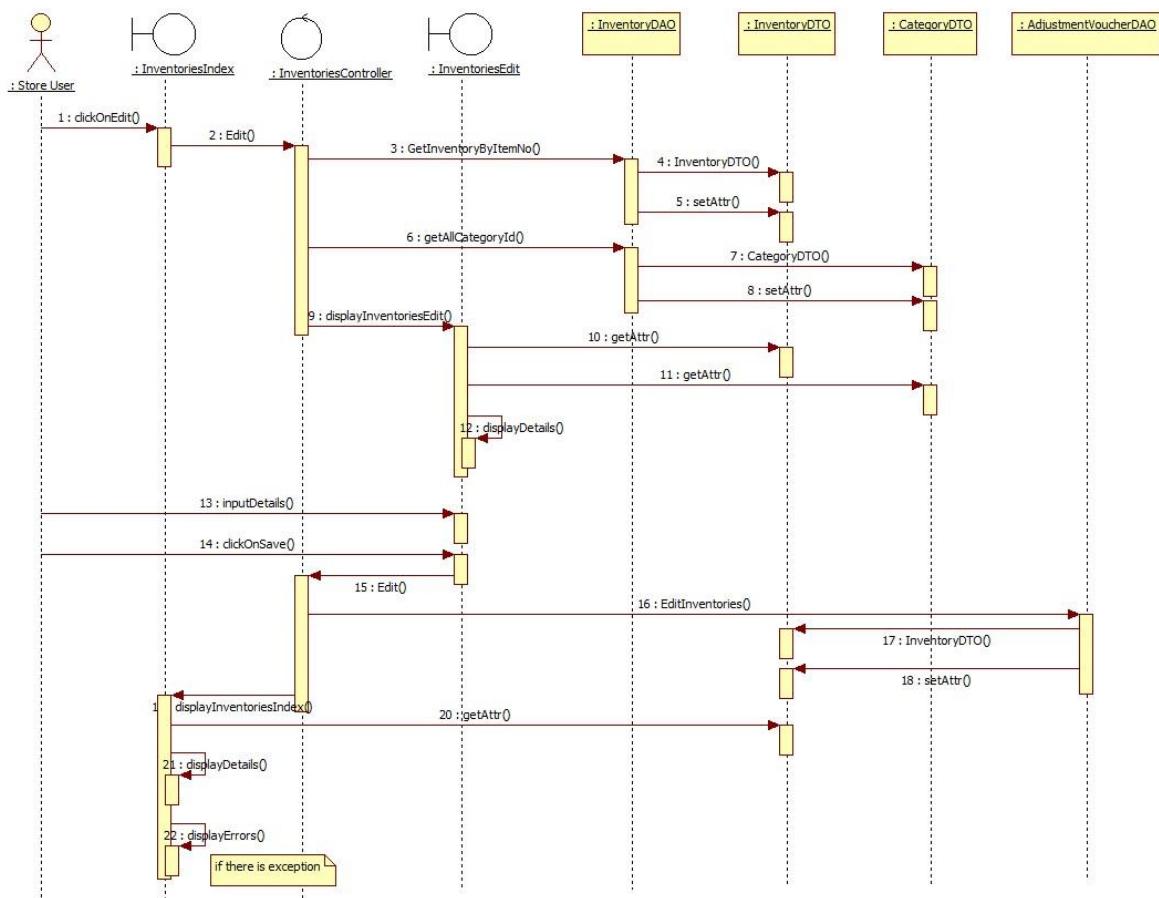
1. Create Inventory



2. View Inventories

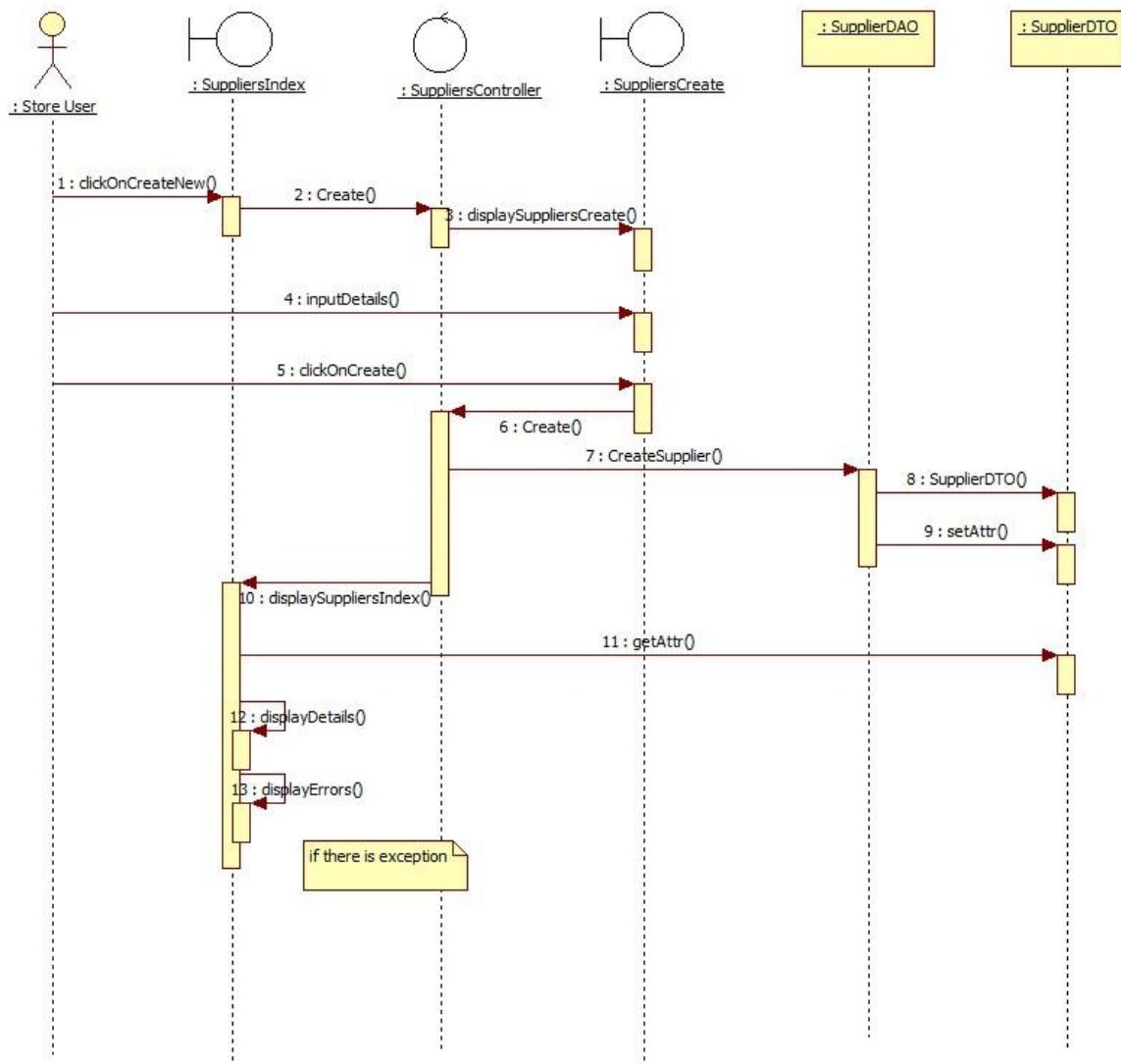


3. Edit Inventories

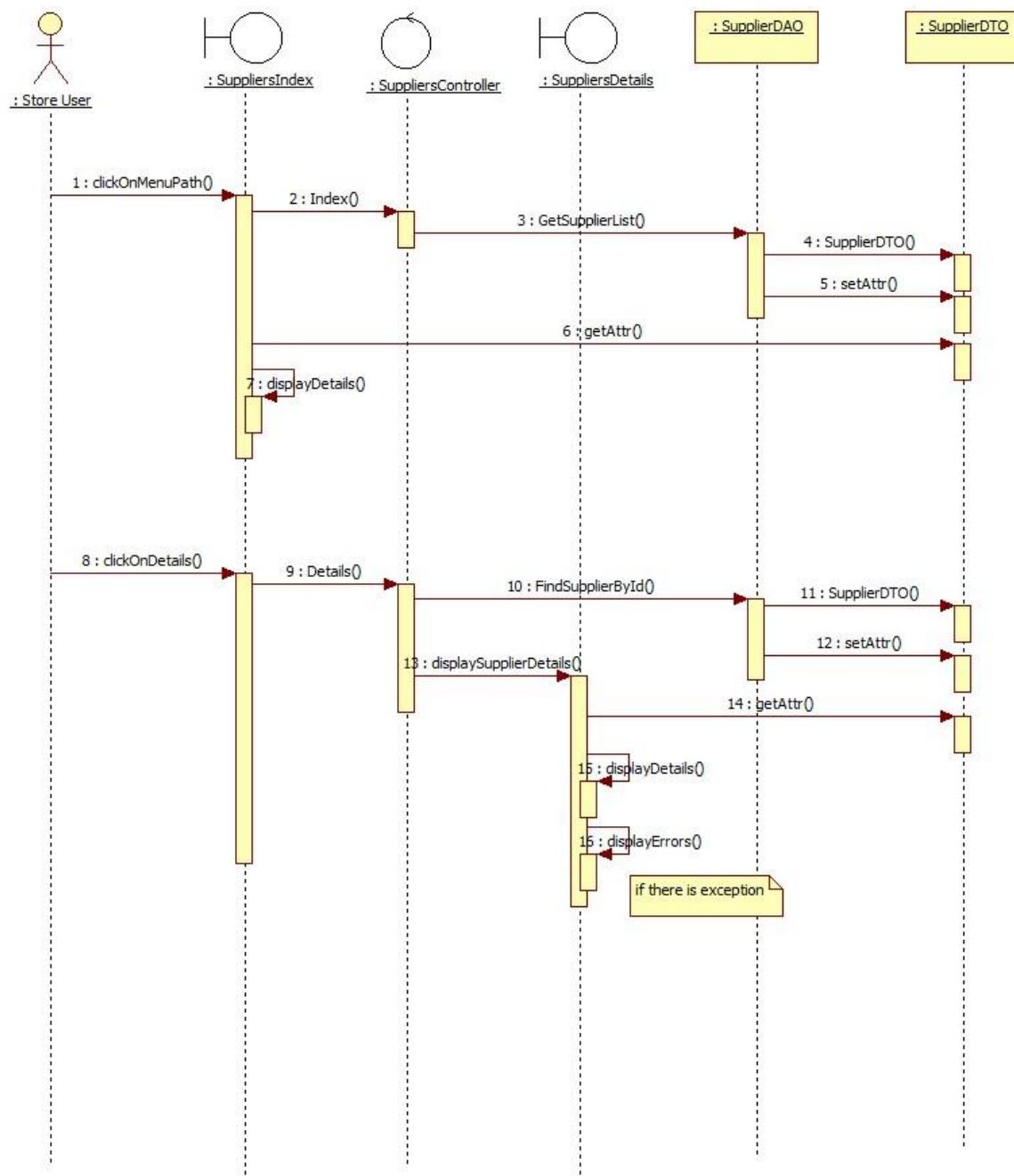


Modify Supplier Information

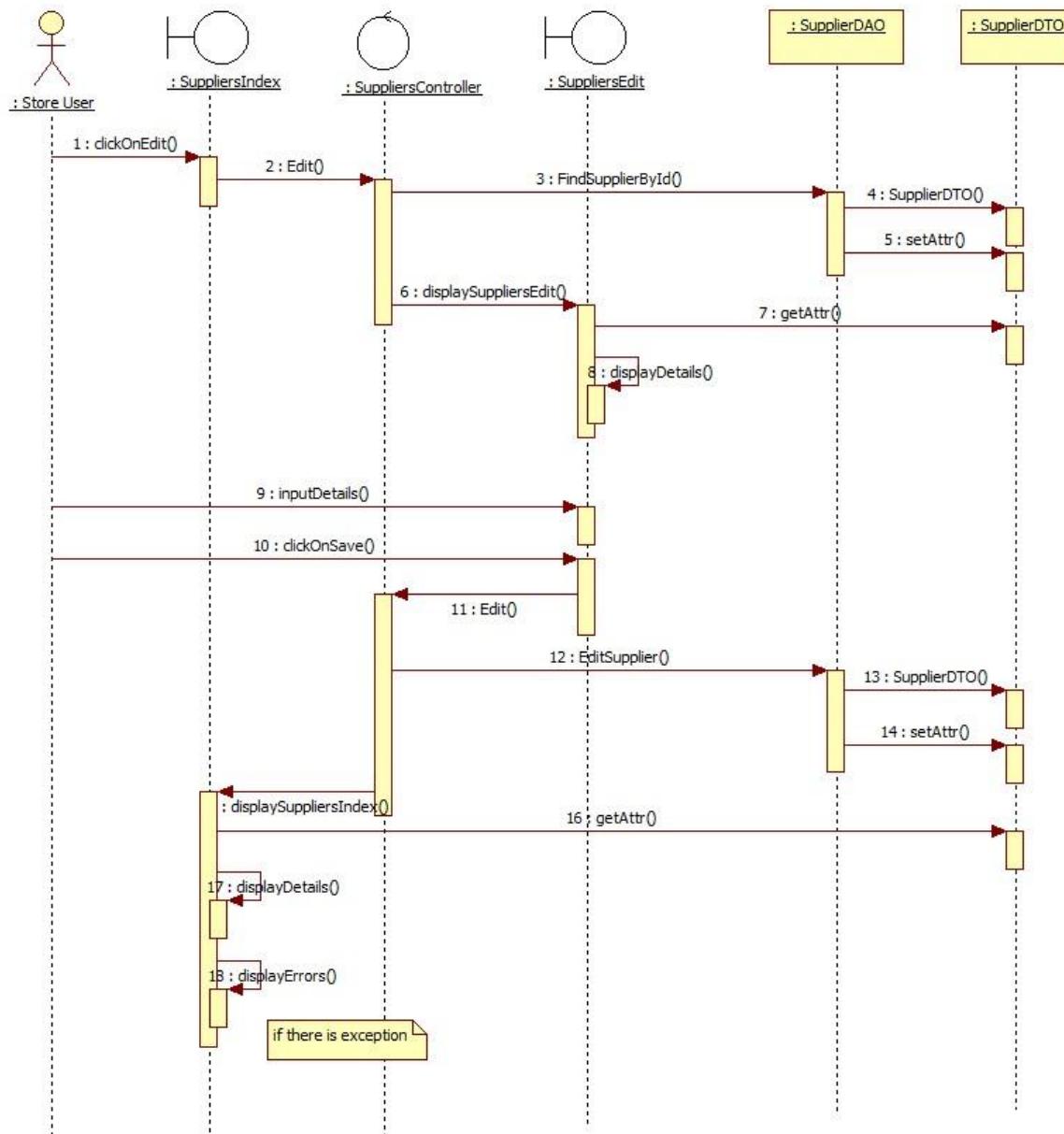
1. Create Suppliers



2. View Suppliers

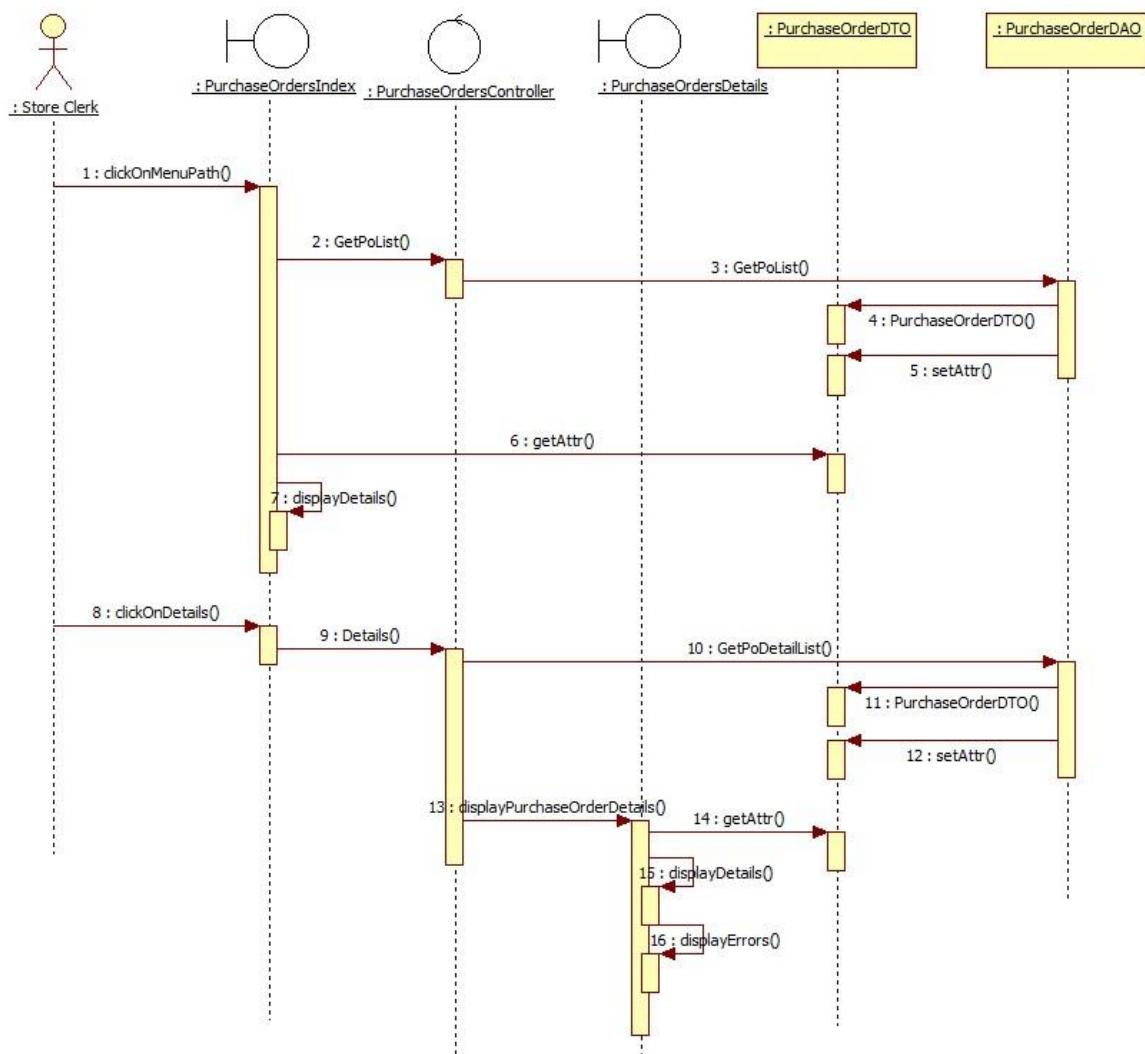


3. Edit Suppliers

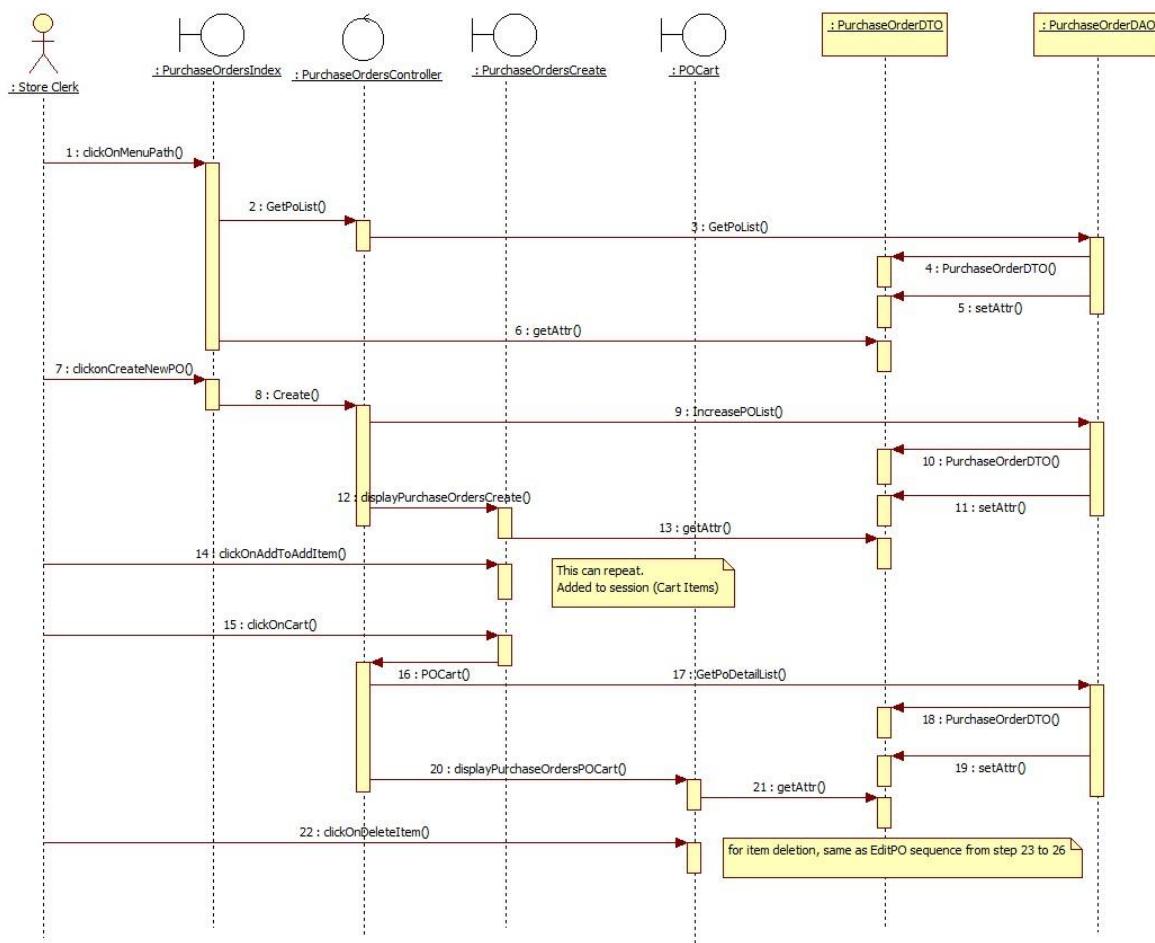


Process Purchase Order

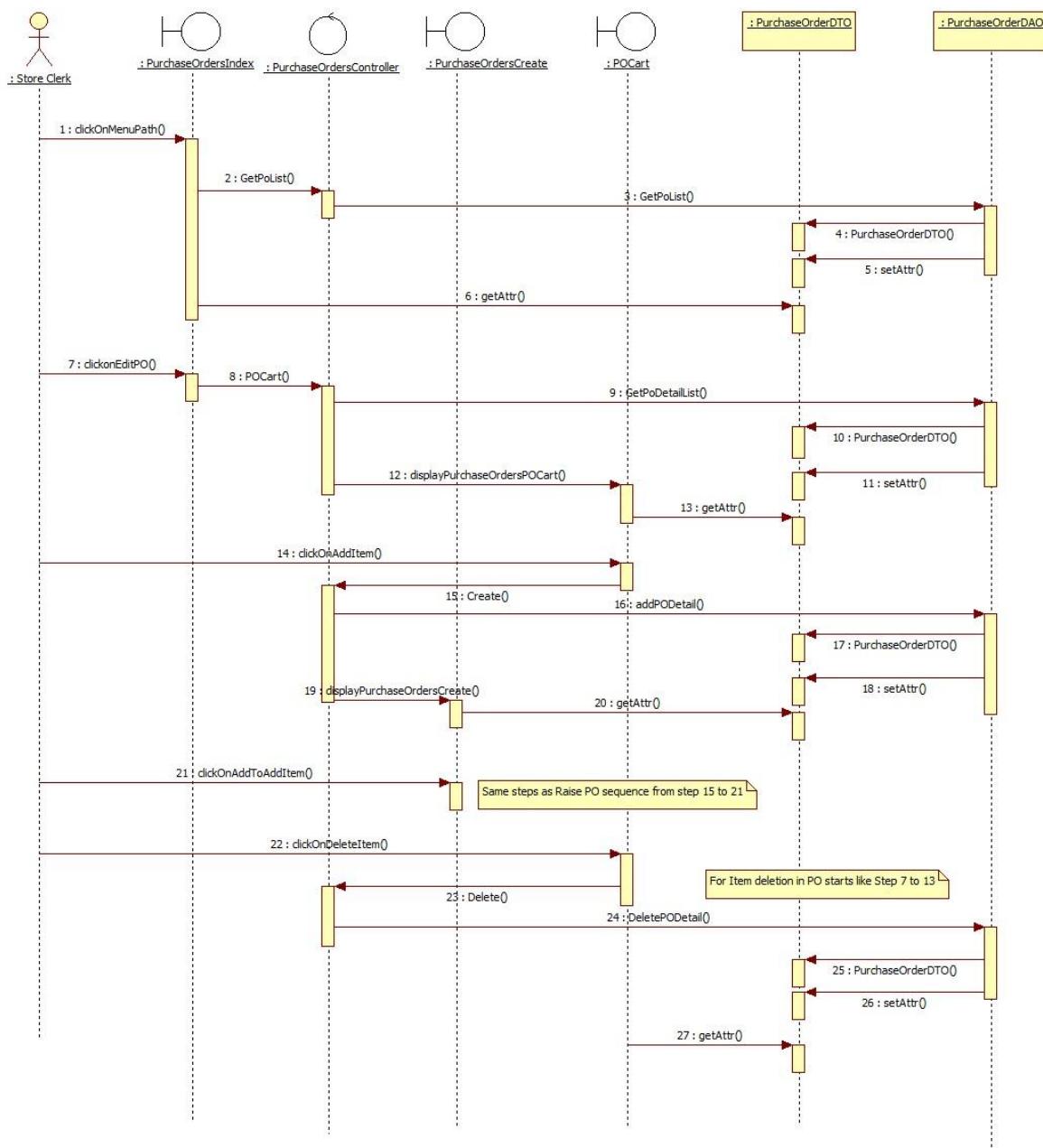
1. View Purchase Order details



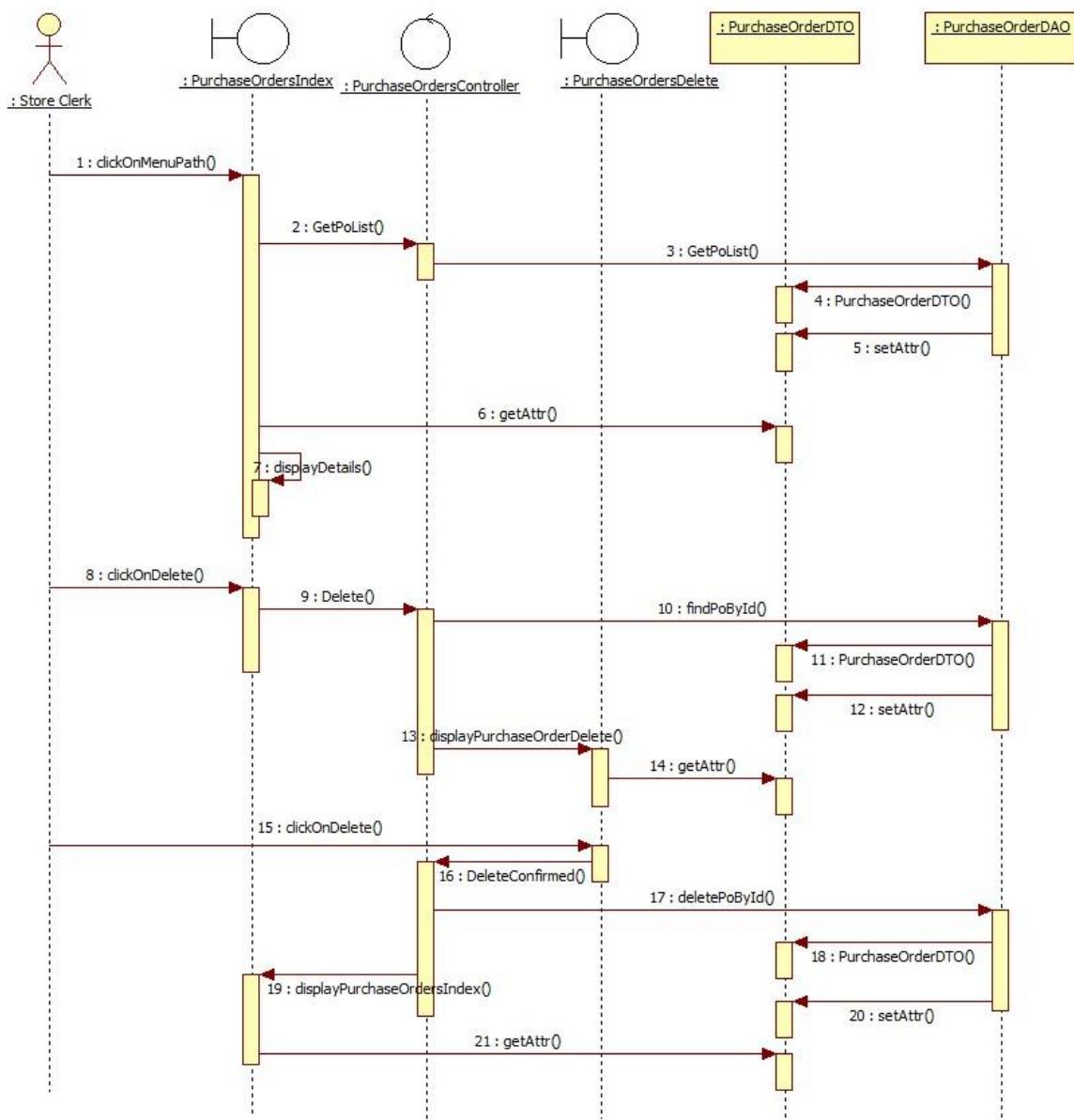
2. Raise Purchase Order



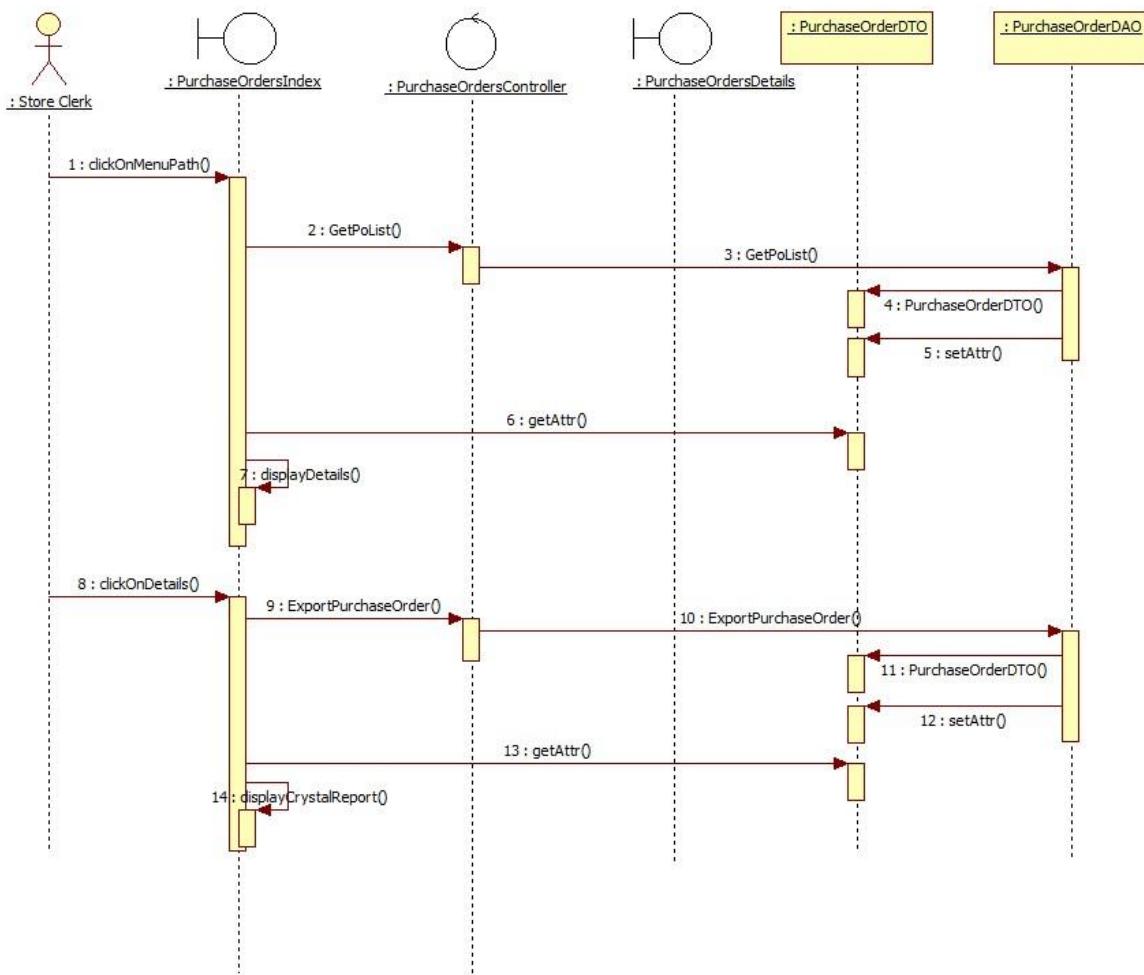
3. Edit Purchase Order



4. Delete Purchase Order

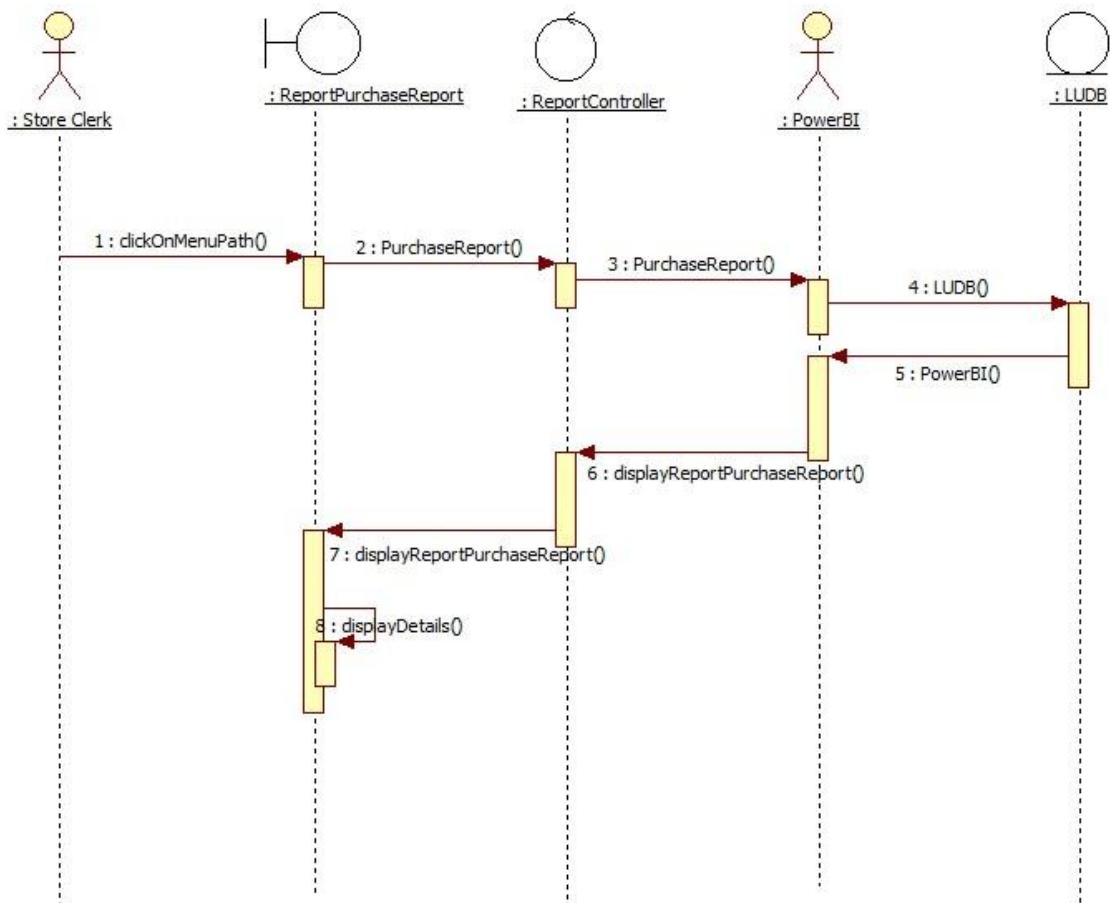


5. View Purchase Order Report

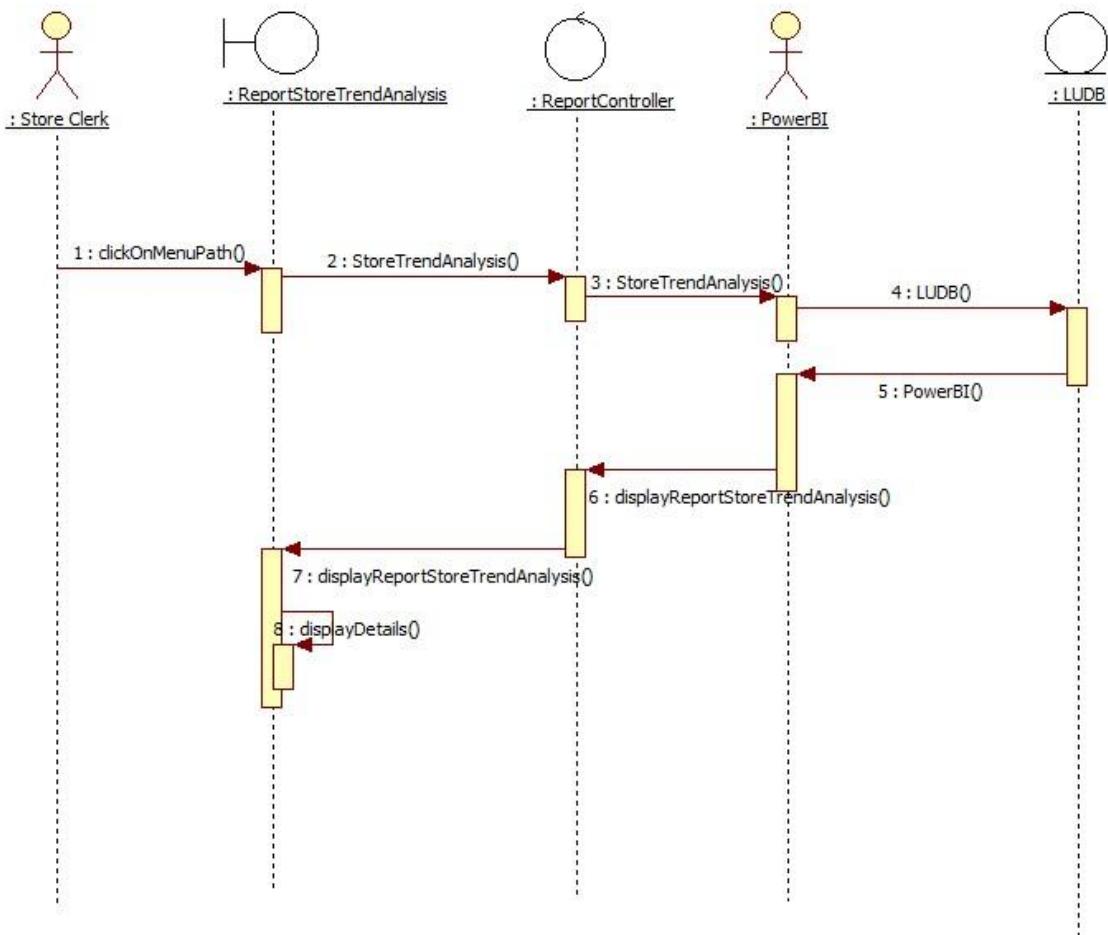


View Reports

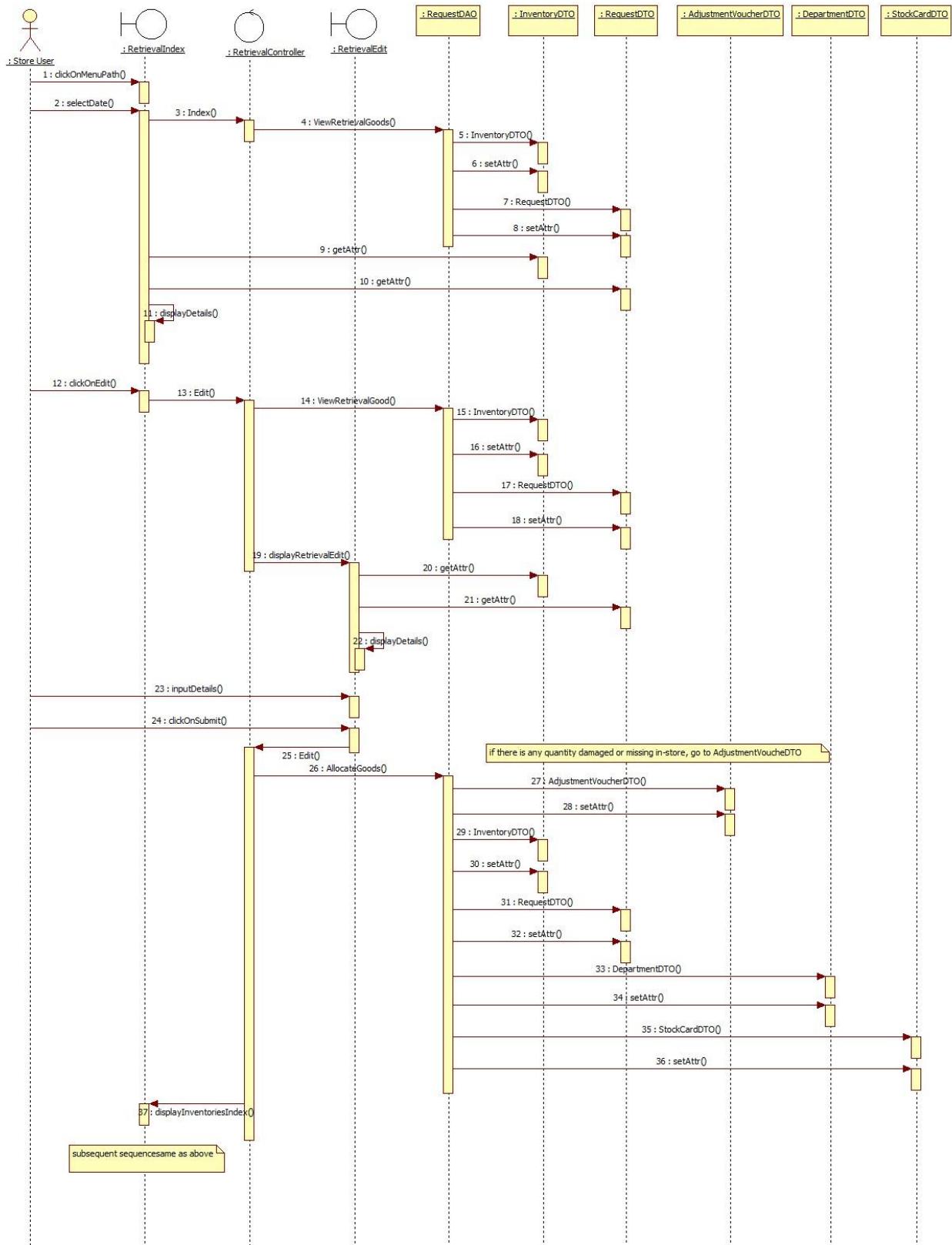
1. Generate Purchase Order Analysis



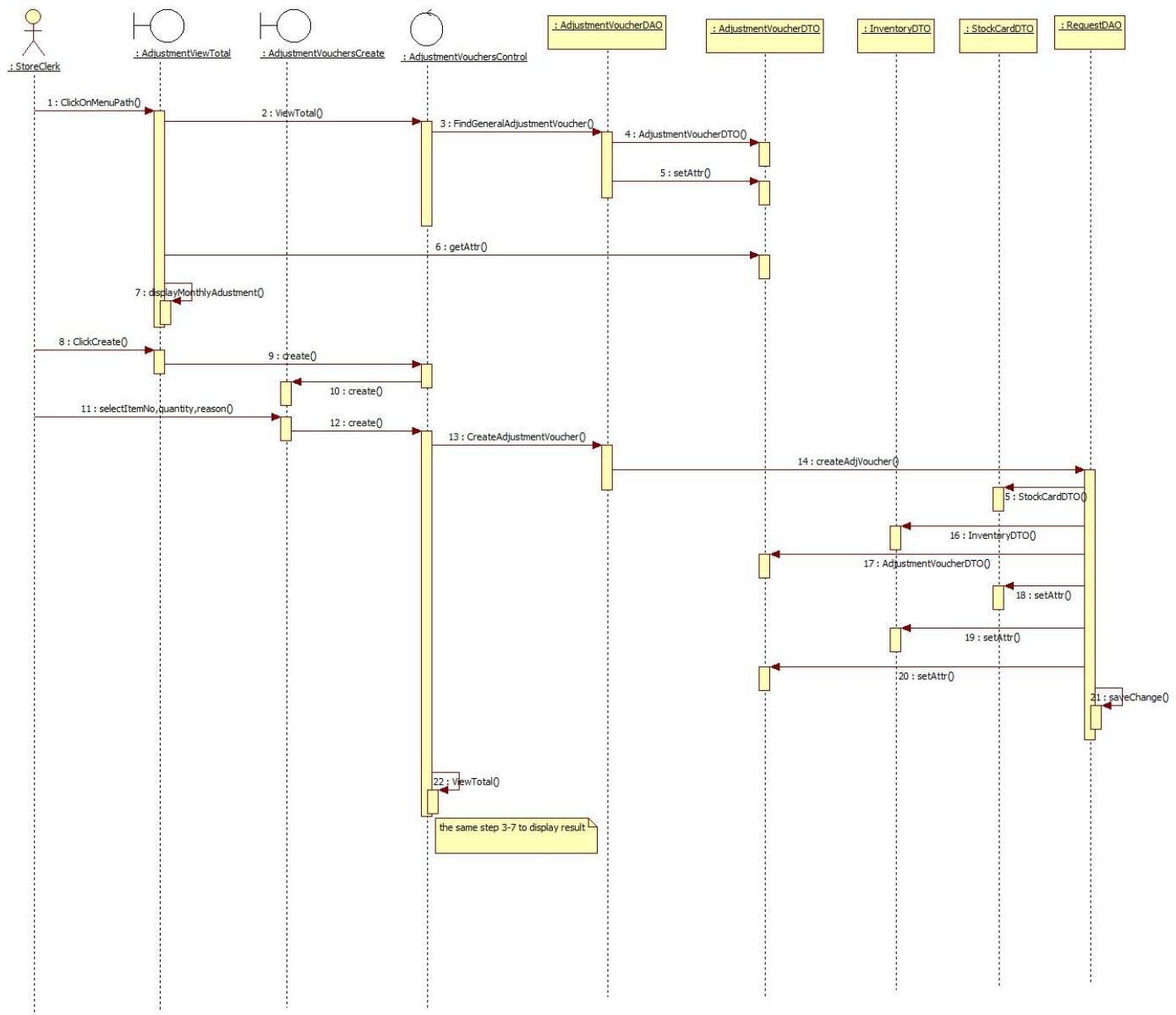
2. Generate Trend Analysis



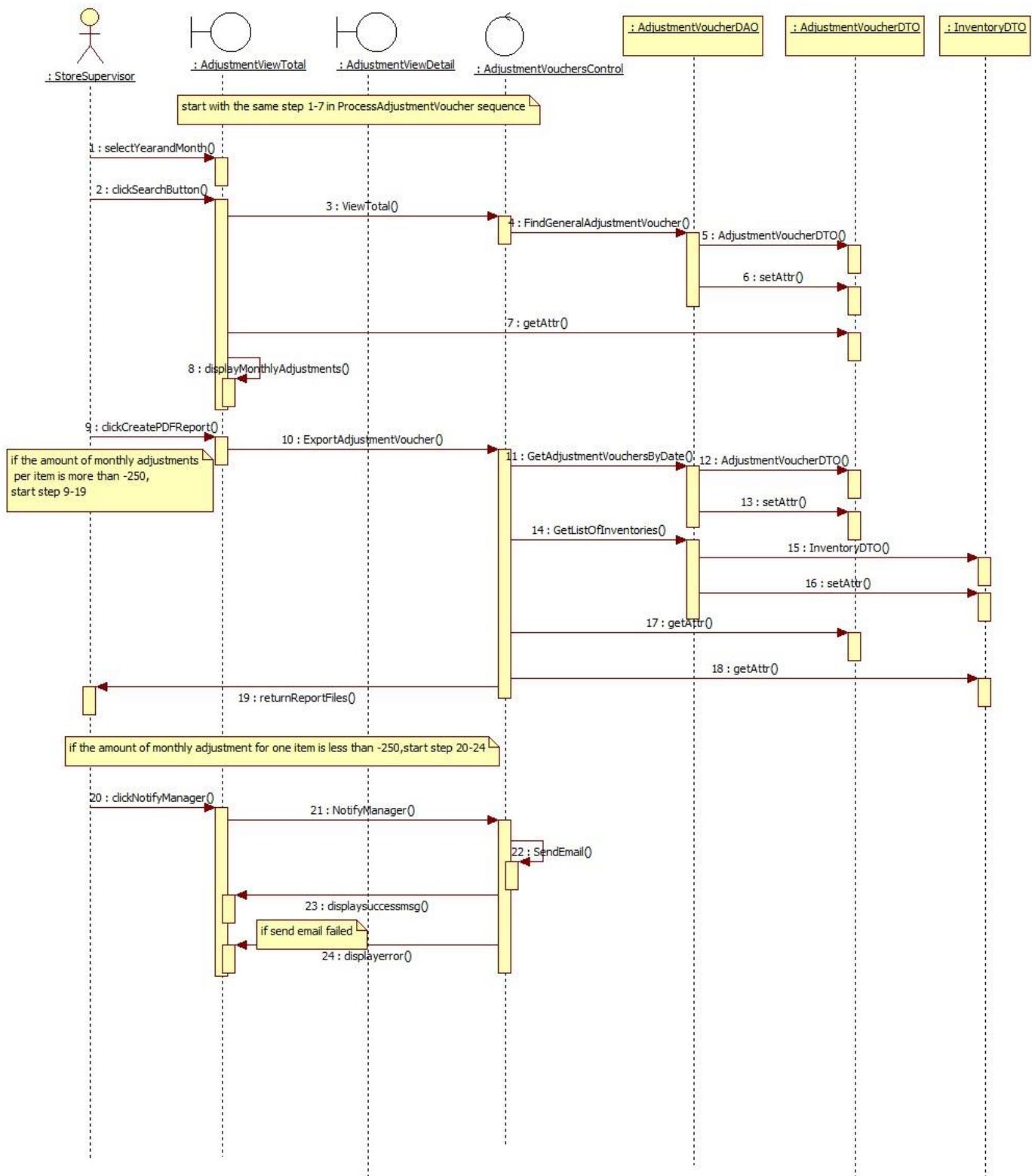
Process Retrieval List



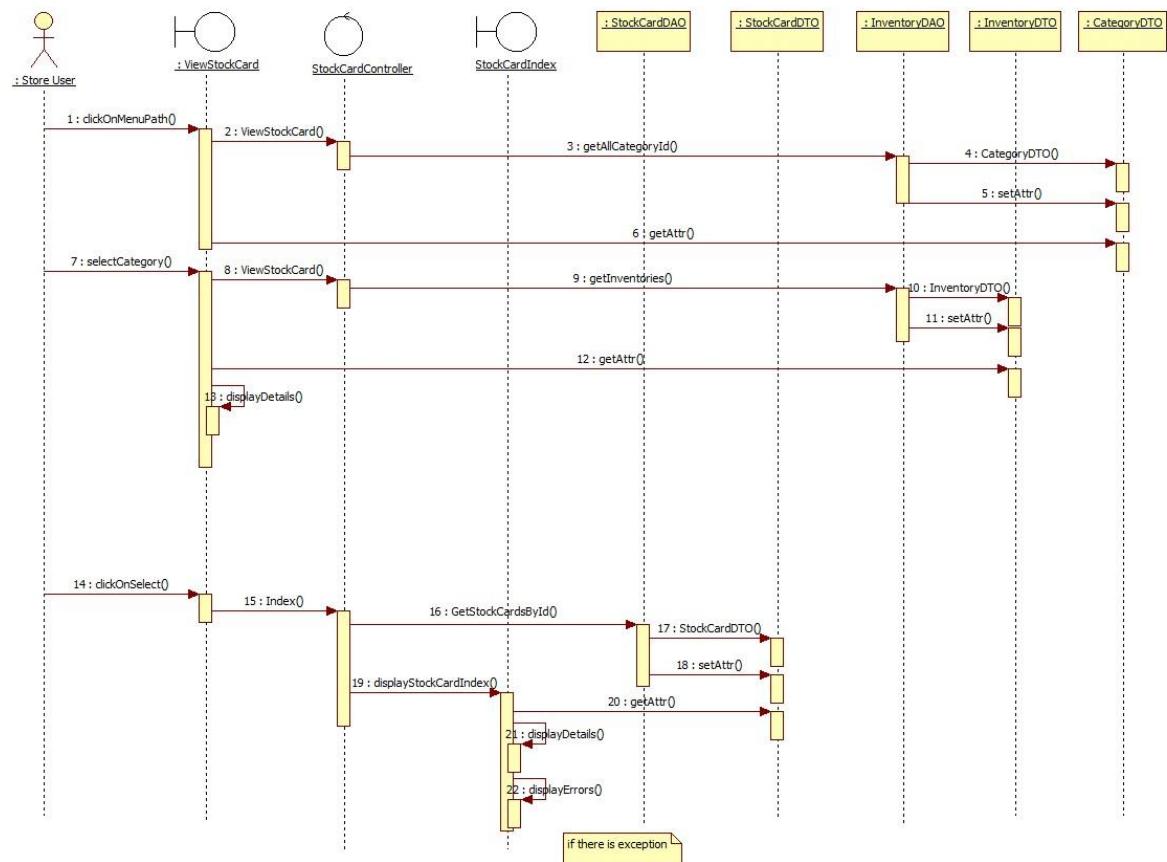
Process Adjustment Report



Issue Adjustment Voucher Report

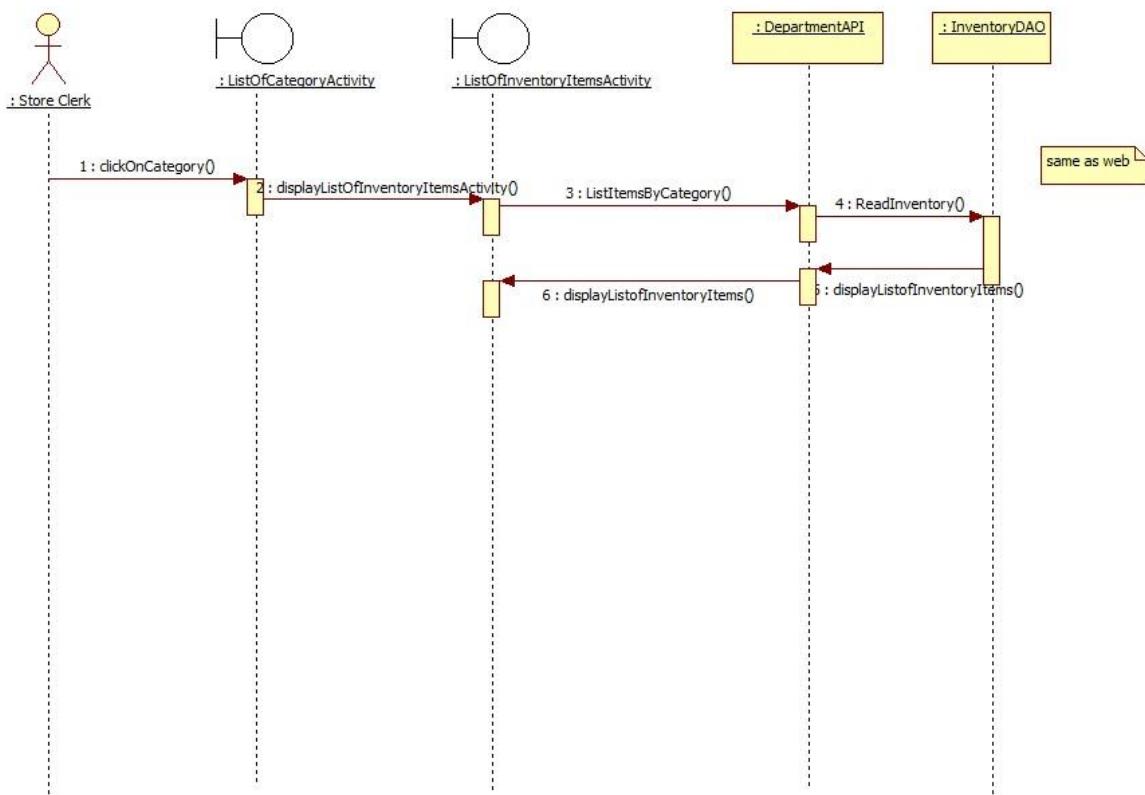


View Stock Card

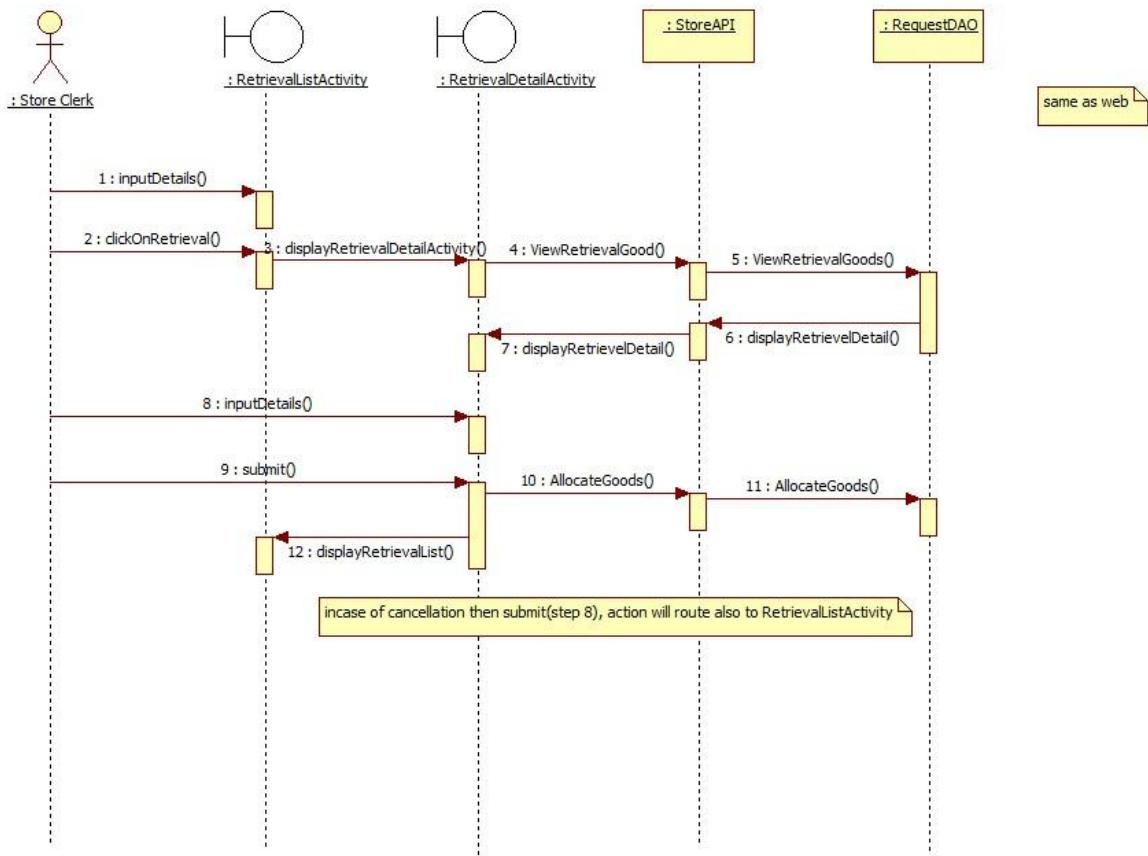


8.4 Mobile Application for Store Sequence Diagram

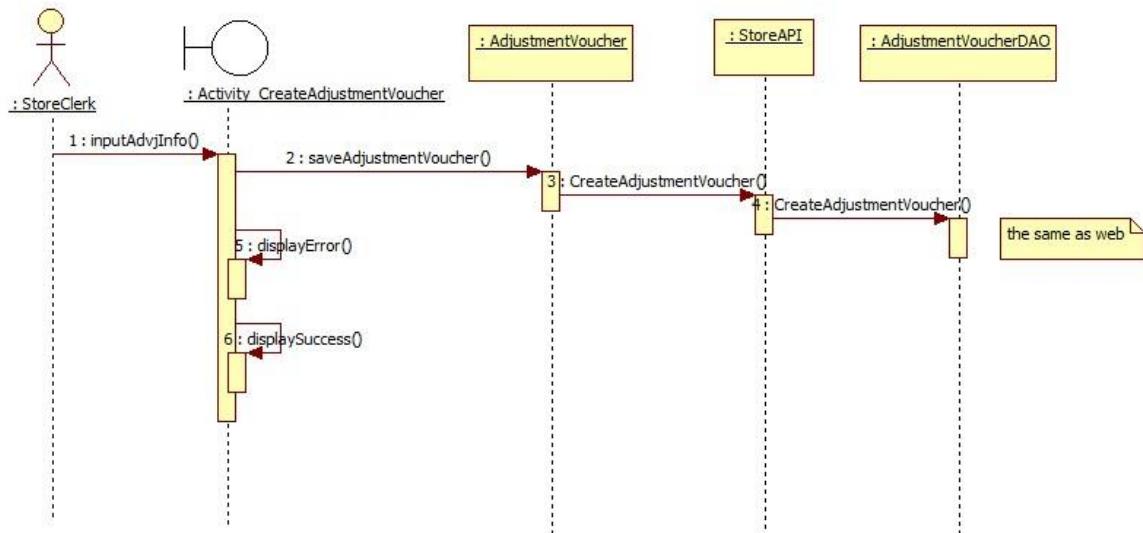
View Stock Card



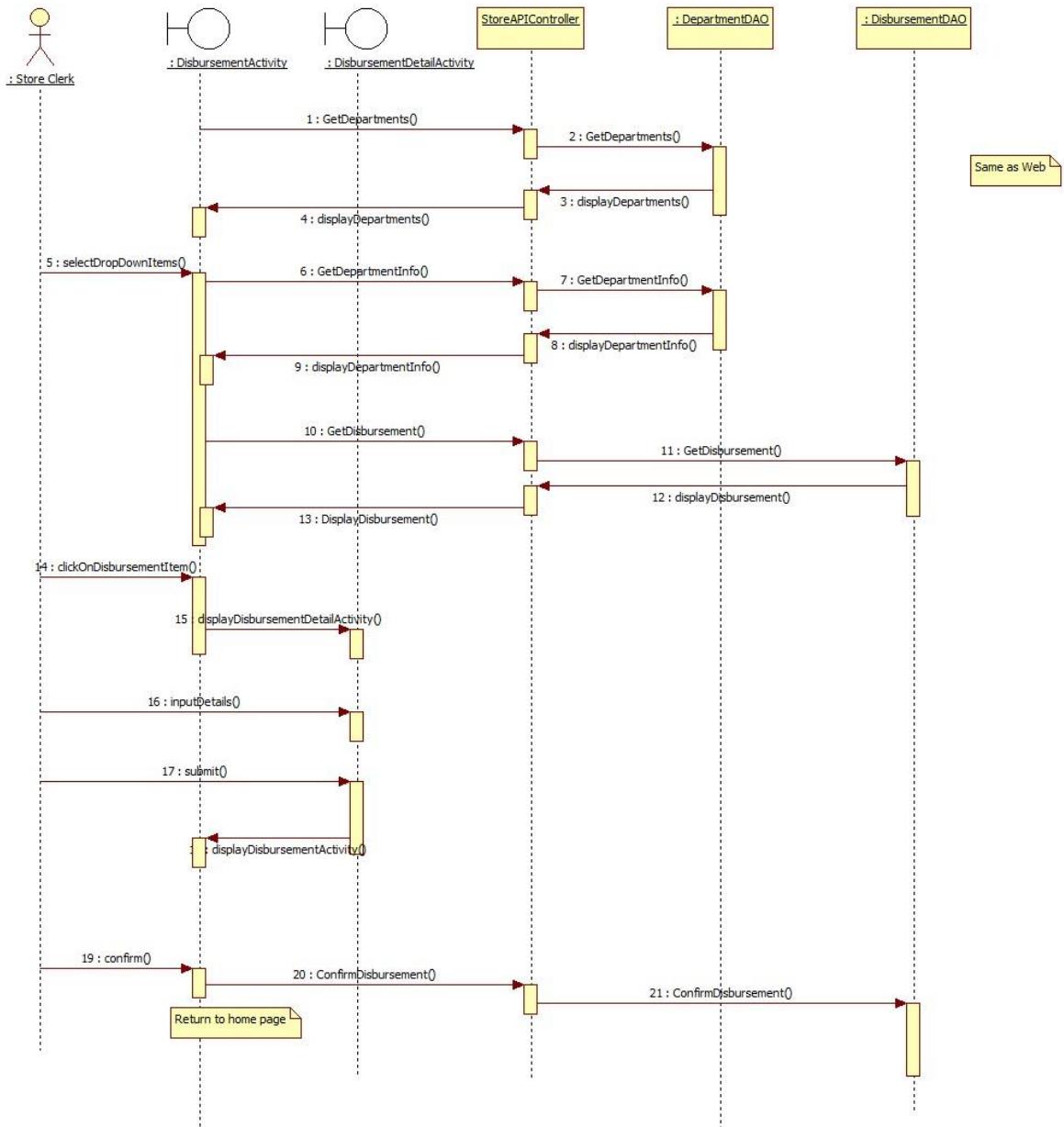
Process Retrieval List



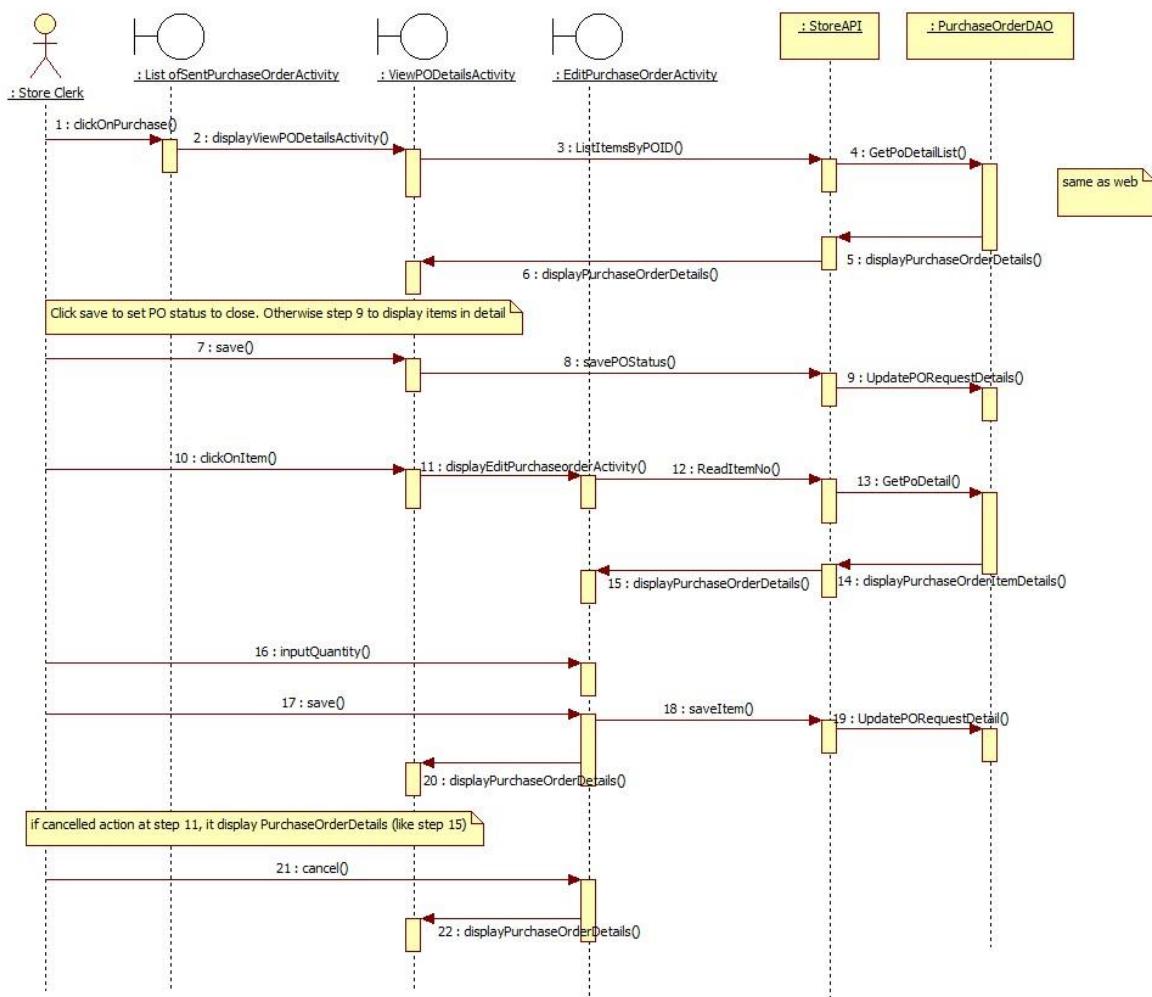
Process Adjustment Voucher



Process Disbursement List



Close Purchase Order upon supplier delivery



9. Test Scripts

Test Type: System			
TestID:1	Prepare by/ date: Madhumitha Murali, Aisha Ashiq/10th Feb,2019	Tested by: Madhumitha Murali, Aisha Ashiq /date: 10/02/2019	
Test Description		Program tested <u>Boundary Class:</u> AdjustmentViewTotal, AdjustmentVouchersCreate, AdjustmentViewDetail <u>Control Class:</u> AdjustmentVouchersController <u>Entity Class:</u> AdjustmentVoucher, Inventory, StockCard, Request <u>DB Broker</u> LUDB	
s/n	Test step	Expected Result	Actual Result
	<p>Step 1: Login to the “Logic University Application” using the Store Clerk login</p> <p>Step 2: Select the “Adjustment Voucher” link from the “Stock” dropdown menu</p> <p>Step 3: Select any year from the “Year” dropdown field and click on “Search” button</p> <p>Step 4: Select any year and month from the “Year” and “Month” dropdown fields and click on “Search” button</p> <p>Step 5: Click on “Create New” button</p> <p>Step 6: Select any category from the Category dropdown</p> <p>Step 7: Select any item from the item dropdown list and enter the quantity and remark. Click on “Add Item” button</p>	<p>Login should be successful without any error and home page for the store clerk should be displayed</p> <p>Adjustment voucher home page should be displayed with the table displaying the existing adjustment voucher details and the following dropdown fields: Year: Month:</p> <p>Table should be refreshed with the details as per the selected year and for all months with the rows highlighted, if the “Total Amount” is greater than \$250</p> <p>Table should be refreshed with the details as per the selected year and month with the rows highlighted, if the “Total Amount” is greater than \$250</p> <p>Create Adjustment Voucher page was displayed with the following fields: Category: Item Name: Quantity: Date: Remark: Add Item</p> <p>“Item Name” dropdown was populated as per the selected category</p> <p>Details was added successfully in the adjustment voucher table and was reflected in the details displayed without any errors</p>	<p>Login was successful without any error and home page for store clerk was displayed</p> <p>Adjustment voucher home page was displayed with the table displaying the existing adjustment voucher details and the following dropdown fields: Year: Month:</p> <p>Table was refreshed with the details as per the selected year and for all months with the rows highlighted, if the “Total Amount” is greater than \$250</p> <p>Table was refreshed with the details as per the selected year and month with the rows highlighted, if the “Total Amount” is greater than \$250</p> <p>Create Adjustment Voucher page was displayed with the following fields: Category: Item Name: Quantity: Date: Remark: Add Item</p> <p>“Item Name” dropdown was populated as per the selected category</p> <p>Details was added successfully in the adjustment voucher table and was reflected in the details displayed without any errors</p>

Test Type: System			
TestID:2	Prepare by/ date: Madhumitha Murali,Aisha Ashiq/10th Feb,2019	Tested by:/date: Madhumitha Murali, Aisha Ashiq /date: 10/02/2019	
Test Description		Program tested <u>Boundary Class:</u> RequestList, RequestStore, RequestDepartmentCart <u>Control Class:</u> RequestController <u>Entity Class:</u> Department, Staff, Inventory, Category, Request <u>DB Broker</u> LUDB	
s/ n	Test step	Expected Result	Actual Result

	<p>Step 1: Login to the “Logic University Application” using the Department Staff login</p> <p>Step 2: Add New Request: Select the “Request List” link from the “Catalogue” dropdown menu</p> <p>Step 3: Click on “Add Request” button</p> <p>Step 4: Select any category from the Category dropdown</p> <p>Step 5: Click on “+” button (Add to Cart) to add the required items to the cart</p> <p>Step 6: Click on “Cart” button to view the items in the cart</p> <p>Step 7: Click on “-” button to delete any item from the cart</p> <p>Step 8: Click on “Submit” button to submit the request for Department Head’s approval</p> <p>Step 9: Modify existing Request: Login and Select “Request List” link from the “Catalogue” dropdown menu</p> <p>Step 10: Click “Edit” link on any “Pending” requests</p> <p>Step 11: Click on “Add item” or “-” button to add or delete the items for the selected request</p> <p>Step 12: Click “Submit” button to update the request</p> <p>Step 13: Click “Delete” link on any “Pending” requests</p>	<p>Login should be successful without any error and home page for the department staff should be displayed</p> <p>Request List home page should be displayed with the existing request details</p> <p>Page should be navigated to the “Items” page displaying all the items in the catalogue from where it can be added to the cart</p> <p>Catalogue items should be refreshed based on the selected category</p> <p>Items should be added to the cart when the “+” button is clicked and the “Cart” icon should display the number of items in the cart</p> <p>“Cart” page should be displayed with the details of the items added</p> <p>Items should be deleted from the cart successfully without any error</p> <p>Request should be submitted for approval and new record should be created in the database with status as “Pending” and the same should be displayed in the “Request List” landing page</p> <p>Request List home page should be displayed with the existing request details and with the following links for the “Pending” requests: Edit Details Delete</p> <p>“Cart” page of the selected Request Id should be displayed with the item details</p> <p>Items should be added or deleted in the cart page as per the user’s selection</p> <p>Request list should be updated in the database and submitted for approval and Request List home page should be displayed</p> <p>Request should be deleted from the database and Request List page should be refreshed</p>	<p>Login was successful without any error and home page for the department staff was displayed</p> <p>Request List home page was displayed with the existing request details</p> <p>Page navigates to the “Items” page displaying all the items in the catalogue from where it can be added to the cart</p> <p>Catalogue items was refreshed based on the selected category</p> <p>Items were added to the cart when the “+” button is clicked and the “Cart” icon displays the number of items in the cart</p> <p>“Cart” page was displayed with the details of the items added</p> <p>Items were deleted from the cart successfully without any error</p> <p>Request was submitted for approval and new record was created in the database with status as “Pending” and the same was displayed in the “Request List” landing page</p> <p>Request List home page was displayed with the existing request details and with the following links for the “Pending” requests: Edit Details Delete</p> <p>“Cart” page of the selected Request Id was displayed with the item details</p> <p>Items were added or deleted in the cart page as per the user’s selection</p> <p>Request list was updated in the database and submitted for approval and Request List home page was displayed</p> <p>Request was deleted from the database and Request List page was refreshed</p>
--	---	--	---

Test Type: System		
TestID:3	Prepare by/ date: Madhumitha Murali,Aisha Ashiq/10th Feb,2019	Tested by:/date: Madhumitha Murali, Aisha Ashiq / date: 10/02/2019

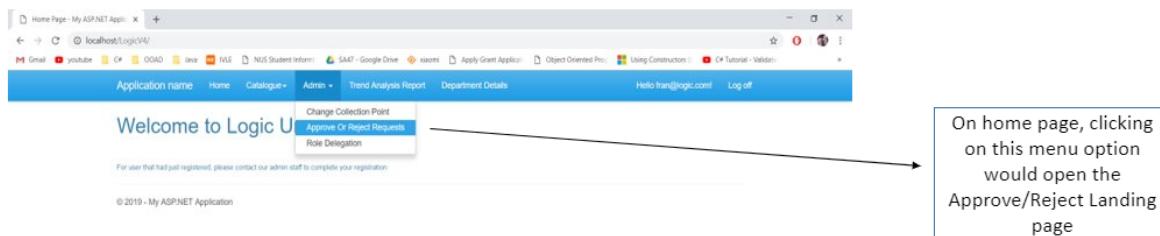
<p>Test Description</p> <p>"Change DR/Collection Point" for Department Head</p> <p>To validate if the user is able to change the department representative or the department collection point through the Logic University Android application</p>		<p>Program tested</p> <p><u>Boundary Class:</u> Home, ChangeCollectionPoint</p> <p><u>Control Class:</u> CollectionPointController</p> <p><u>Entity Class:</u> Department, Staff, CollectionPoint</p> <p><u>DB Broker</u> LUDB</p>	
s/n	Test step	Expected Result	Actual Result
	<p>Step 1: Login to the "Logic University Application" using the Department Head login</p> <p>Step 2: Click on "Change of Rep/collection Point" from the list</p> <p>Step 3: Click on "Check Names" button</p> <p>Step 4: Select any employee name</p> <p>Step 5: Enter text in the "Select Employee" input field and click on "Check Names" button</p> <p>Step 6: Select any employee name</p> <p>Step 7: Select any option from the "Collection Point" dropdown list</p> <p>Step 7: Click on "Submit"</p>	<p>Login should be successful without any error and home page for the department head should be displayed</p> <p>Change of Rep/collection Point screen should be displayed showing the current department rep and collection point</p> <p>Page should be navigated to the ListDRNames page which displays all the employee names available</p> <p>Control goes back to the previous page with the selected employee name populated in the "Select Employee" text field</p> <p>Page should be navigated to the ListDRNames page which displays the employee names starting with the entered text</p> <p>Control goes back to the previous page with the selected employee name populated in the "Select Employee" text field</p> <p>Collection Point should be selected from the dropdown list</p> <p>Selected department rep and collection point should be saved and navigates to the home page</p>	<p>Login was successful without any error and home page for the department head was displayed</p> <p>Change of Rep/collection Point screen was displayed showing the current department rep and collection point</p> <p>Page was navigated to the ListDRNames page which displays all the employee names available</p> <p>Control goes back to the previous page with the selected employee name populated in the "Select Employee" text field</p> <p>Page navigates to the ListDRNames page which displays the employee names starting with the entered text</p> <p>Control goes back to the previous page with the selected employee name populated in the "Select Employee" text field</p> <p>Collection Point was selected from the dropdown list</p> <p>Selected department rep and collection point was saved and navigates to the home page</p>

10. Wire Frames and UI Specifications

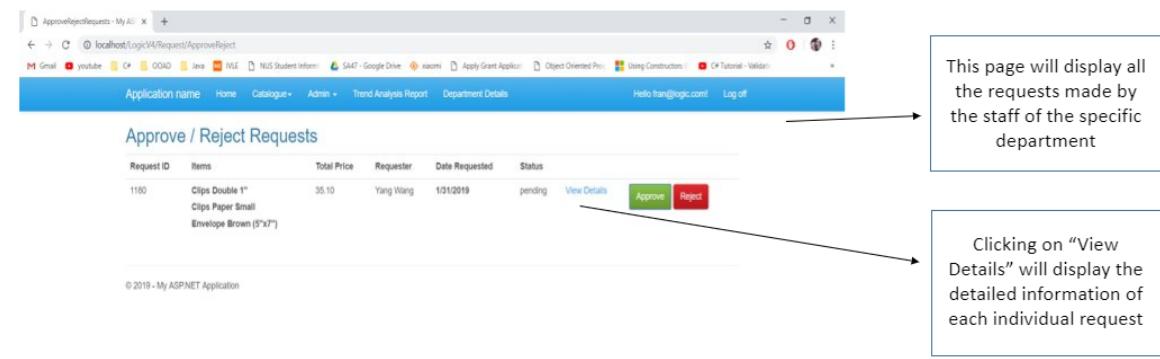
10.1 Department web User interface

Approve/Reject Requests

DH Home page



Approve/Reject page



Request Details page

The screenshot shows a web browser window titled "Detail - My ASP.NET Application". The URL is "localhost:LogiC4/Request/Detail?requestId=1180". The page has a header with links like Application name, Home, Catalogue, Admin, Trend Analysis Report, Department Details, Hello tran@logic.com!, and Log off. Below the header is a table titled "Detail: request 1180" with columns: Category, Item Name, Price, Qty Need, and Unit. The table contains three rows: Clip (Clip Double 1"), Clip (Clips Paper Small), and Envelope (Envelope Brown (5"x7")). A red arrow points from the table area to a callout box.

Category	Item Name	Price	Qty Need	Unit
Clip	Clip Double 1"	12.00	1	Dozen
Clip	Clips Paper Small	22.20	1	Box
Envelope	Envelope Brown (5"x7")	0.90	1	Each

Individual items in the request are displayed with their details like price, qty needed, etc

Approve/Reject page

The screenshot shows a web browser window titled "ApproveRejectRequests - My App". The URL is "localhost:LogiC4/Request/ApproveReject". The page has a header with links like Application name, Home, Catalogue, Admin, Trend Analysis Report, Department Details, Hello tran@logic.com!, and Log off. Below the header is a table titled "Approve / Reject Requests" with columns: Request ID, Items, Total Price, Requester, Date Requested, and Status. One row is shown: Request ID 1180, Items "Clips Double 1", "Clips Paper Small", "Envelope Brown (5"x7")", Total Price 35.10, Requester Yang Wang, Date Requested 1/1/2019, and Status pending. To the right of the table are "Approve" and "Reject" buttons. A red arrow points from the "Approve" button to a callout box. A modal dialog box is overlaid on the page, containing a text input field "Input your remarks:" and a green "Approve" button.

Request ID	Items	Total Price	Requester	Date Requested	Status
1180	Clips Double 1" Clips Paper Small Envelope Brown (5"x7")	35.10	Yang Wang	1/1/2019	pending

Approve

Reject

Input your remarks:

Approve

Clicking on Approve will open a pop-up for entering remarks for approval, entering remarks and clicking on “Approve” will submit the request to the store team

Approve/Reject page

The screenshot shows a web application window titled "Approve/Reject Requests". A single row in a table represents a request:

Request ID	Items	Total Price	Requester	Date Requested	Status
1180	Clips Double 1" Clips Paper Small Envelope Brown (5"x7")	35.10	Yang Wang	1/31/2019	Pending

Below the table is a green "Approve" button and a red "Reject" button. A callout box points to the "Reject" button with the text: "Clicking on Reject will open a pop-up for entering remarks for rejection". A second callout box points to the "Reject" button after a user has entered remarks in a modal dialog, with the text: "Clicking on Reject after entering comments updates the status of the request to \"rejected\"".

Change DR/Collection Point – DH

Change DR/Collection Point page

The screenshot shows a web application window titled "Change Department Representative and Collection Point". It displays current information for the "Zoology department": "current representative" is Pedro Alonso and "current check point" is Management School (11:00am) by store clerk 1. Below these, there are dropdown menus for "New Department Representative" and "New collection point".

A callout box points to the "New Department Representative" dropdown menu with the text: "Here the DH can change the DR for the department". Another callout box points to the "New collection point" dropdown menu with the text: "Here the DH can change collection point of the stationery items to be collected by the DR for the department". A third callout box points to the "Update" button at the bottom left with the text: "Clicking on update will change the selected DR and collection point for the department".

Change DR/Collection Point page

The screenshot shows a web application interface titled "Change Department Representative and Collection Point". It has a top navigation bar with links like Application name, Home, Catalogue, Admin, Trend Analysis Report, and Department Details. The main content area has two dropdown menus. The first dropdown, "Zoology department's current representative:", shows "Pedro Alonso" as the selected item. The second dropdown, "Zoology department's current check point:", lists several collection points with their descriptions and times, such as "Management School (11:00am) by store clerk 1", "Medical School (9:30am) by store clerk 2", etc. A blue box on the right says "Here the DH can change the DR for the department". Another blue box below it says "Here the DH can change collection point of the stationary items to be collected by the DR for the department". A third blue box at the bottom right says "Clicking on update will change the selected DR and collection point for the department".

Change Collection Point – DR

DR Home page

The screenshot shows a web application interface titled "Home Page - My ASP.NET App". It has a top navigation bar with links like Application name, Home, Catalogue, Admin, and a "Change Collection Point" button. The main content area says "Welcome to Logic University!". A blue box on the right says "Clicking on \"Change Collection Point\" in the menu navigates to the \"Change Collection point\" page".

Change Collection Point page

The screenshot shows a web browser window titled "ChangeCollectionPoint - My ASP.NET App". The URL is "localhost:49241/CollectionPoint/ChangeCollectionPoint". The page has a header with "Application name", "Home", "Catalogue", "Admin", "Hello pedro@logic.com", and "Log off". The main content area is titled "Change Collection Point". It displays a table with two rows. The first row is for "Zoology department current check point" and lists "Medical School (9:30am) by store clerk 2". The second row is for "New collection point:" and lists several options: "Stationery Store - Administration Building (9:30am) by store clerk 1", "Management School (11:00am) by store clerk 1", "Medical School (9:30am) by store clerk 2", "Engineering School (11:00am) by store clerk 2", "Science School (9:30am) by store clerk 3", and "University Hospital (11:00am) by store clerk 3". There is an "Update" button at the bottom. A copyright notice at the bottom left says "© 2019 - My ASP.NET Application".

Collection can be changed using the Radio buttons

Clicking on "Update" saves the selected collection point for the department and goes back to the DR home page

Role Delegation

DH Home page

The screenshot shows a web browser window titled "Home Page - My ASP.NET App". The URL is "localhost:LogicUniversityApp/". The page has a header with "Application name", "Home", "Catalogue", "Admin", "Trend Analysis Report", "Department Details", "Hello fran@logic.com", and "Log off". The "Admin" menu is open, showing three options: "Change Collection Point", "Approve Or Reject Requests", and "Role Delegation". The "Role Delegation" option is highlighted. A copyright notice at the bottom left says "© 2019 - My ASP.NET Application".

Clicking on "Role Delegation" in the menu navigates to the "Role Delegation" page

Role Delegation page

This screenshot shows the 'Role Delegation' page. At the top, there's a navigation bar with links like Application name, Home, Catalogue, Admin, Trend Analysis Report, Department Details, Hello fran@logic.com!, and Log off. Below the navigation is a table titled 'Role Delegation' with columns: Name, Delegated, Start Date, and End Date. Three rows are listed: Yang Wang, Ileen Holmwood, and Imke Klerkse. Each row has a checkbox under 'Delegated' and a 'Select' button under 'End Date'. A note at the bottom says 'No staff had been given authority'.

This page displays the Role delegation details with the list of DR names who are eligible for delegation

Clicking on "Select" button displays the delegation details below

This screenshot shows the 'Role Delegation' page after selecting Ileen Holmwood. The table now shows 'Ileen Holmwood' under 'Delegation representative selected'. Below the table, there are fields for 'From:' and 'To:' with a date picker between them. The date picker shows February 2019, with the current date set to 31/01/2019. Buttons for 'Remove' and 'Submit' are at the bottom left.

Clicking on "Select" in the previous page displays this page with the selected DR name

"From" and "To" date for the delegation can be selected using the date picker

Clicking on "Submit" saves the delegation details

Role Delegation page

The screenshot shows a web application window titled "RoleDelegation - My ASP.NET App". The URL is "localhost/LogicUniversityApp/RoleDelegation/RoleDelegation". The page has a header with "Application name", "Home", "Catalogue", "Admin", "Trend Analysis Report", "Department Details", "Hello fran@logic.com!", and "Log off". Below the header is a table titled "Role Delegation" with columns "Name", "Delegated", "Start Date", and "End Date". Three rows are listed: Yang Wang (unchecked), Ileen Holmwood (checked, with start date 1/2/2019 and end date 13/2/2019), and Imke Klerkse (unchecked). A callout box points to the "Ileen Holmwood" row with the text: "Clicking on \"Submit\" in the previous page displays this page with the Delegated field marked and the period of delegation for the selected DR". Below the table is a section titled "Delegation representative selected" with "Ileen Holmwood" highlighted. It includes fields for "From" (1/2/2019) and "To" (13/2/2019). At the bottom are "Remove" and "Submit" buttons. A callout box points to the "Remove" button with the text: "Clicking on \"Remove\" removes the currently selected DR from delegation". The footer shows "© 2019 - My ASP.NET Application" and the URL "localhost/LogicUniversityApp".

View Stationery Catalogue

Home page

The screenshot shows a web application window titled "Home Page - My ASP.NET App". The URL is "localhost/LogicUniversityApp". The page has a header with "Application name", "Home", "Catalogue", "Admin", "Hello pedro@logic.com!", and "Log off". A dropdown menu labeled "Items" is open, showing "Request List". A callout box points to the "Items" menu with the text: "Clicking on Items menu will open the Catalogue page". The main content area says "Welcome to Logic University!" and "For user that had just registered, please contact our admin staff to complete your registration". The footer shows "© 2019 - My ASP.NET Application".

Catalogue page

Item Name	Store Balance	Unit	Price	Cart
Clips Double 1"	237	Dozen	12.00	
Clips Double 2"	180	Dozen	18.00	
Clips Double 3/4"	240	Dozen	21.00	
Clips Paper Large	180	Box	27.00	
Clips Paper Medium	120	Box	24.00	
Clips Paper Small	180	Box	22.20	
Envelope Brown (3"x6")	2000	Each	0.70	
Envelope Brown (3"x6") w/ Window	1600	Each	0.80	

Select any category from the dropdown to display the item list as per selection

Catalogue page

Store Balance	Unit	Price	Cart	
240	Packet	3.00		
300	Packet	3.50		
180	Packet	3.00		
120	Packet	3.50		
0	Packet	3.50		
Pad Post Memo 2"x6"	180	Packet	3.50	
Pad Post Memo 3"x6"	120	Packet	3.50	

Selecting category from the dropdown displays the item list as per selection

Raise Request – DR/Staff

Request List page

The screenshot shows a web application window titled "List - My ASP.NET Application". The URL is "localhost/LogicUniversityApp/Request/List". The page has a header with "Application name", "Home", "Catalogue", "Admin", "Hello pedro@logic.com", and "Log off". Below the header is a section titled "MY REQUEST" containing a table of user requests. The table has columns: Request ID, Item Name, Total Price, Date, Status, and Remark. There are four rows of data. At the top left of the table area is a blue button labeled "Add Request".

Request ID	Item Name	Total Price	Date	Status	Remark
1174	Clips Double 3/4" Clips Paper Large Clips Double 1"	60.00	31/1/2019	Rejected	Detail
1164	Transparency Blue Transparency Clear Transparency Green Transparency Red Transparency Reverse Blue Transparency Cover 3M	48.00	4/1/2019	approved	Detail
1157	Hole Puncher Adjustable Trays In/Out Pen Whiteboard Marker Black	99.00	14/12/2018	approved	Detail
1155	Pen Felt Tip Black Eraser (soft)	151.00	26/11/2018	approved	Detail

This "Request List" Page displays all the existing requests of the logged in user

Clicking on "Add Request" button navigates to the "Catalogue" page from where the items can be added to the cart

Clicking on "Detail" link will navigate to the "Request Details" page which displays the details of all the items of the selected request

Catalogue page

The screenshot shows a web application window titled "Store - My ASP.NET Application". The URL is "localhost/LogicUniversityApp/Request/Store". The page has a header with "Application name", "Home", "Catalogue", "Admin", "Hello pedro@logic.com", and "Log off". Below the header is a section titled "STORE" containing a table of store items. The table has columns: Item Name, Store Balance, Unit, Price, and a "Cart (0)" button. There are eight rows of data.

Item Name	Store Balance	Unit	Price	Cart (0)
Clips Double 1"	237	Dozen	12.00	
Clips Double 2"	180	Dozen	18.00	
Clips Double 3/4"	240	Dozen	21.00	
Clips Paper Large	180	Box	27.00	
Clips Paper Medium	120	Box	24.00	
Clips Paper Small	180	Box	22.20	
Envelope Brown (3"x6")	2000	Each	0.70	
Envelope Brown (3"x6") w/ Window	1600	Each	0.80	

Clicking on "Add Request" from the Request List page displays this "Catalogue" page

Clicking on "Cart" button navigates to the "Cart" page

Clicking on "+" button adds the item to the cart

Cart page

This page displays the list of items added to the cart

Clicking on "Add Item" button goes back to the "Catalogue" page

Clicking on "-" button deletes the item from the cart

Clicking on "Submit" button creates a new request in the request list and goes back to the "Request List" page

Category	Item Name	Price	Qty Need	Unit
Clip	Clips Double 1"	12.00	<input type="text" value="1"/>	Dozen <input type="button" value="-"/>
Clip	Clips Double 2"	18.00	<input type="text" value="1"/>	Dozen <input type="button" value="-"/>
Clip	Clips Double 3/4"	21.00	<input type="text" value="1"/>	Dozen <input type="button" value="-"/>
Clip	Clips Paper Large	27.00	<input type="text" value="1"/>	Box <input type="button" value="-"/>

© 2019 - My ASP.NET Application

Submit

Request Details page

This "Request Details" Page displays all the item details of the selected requests from the "request List" page

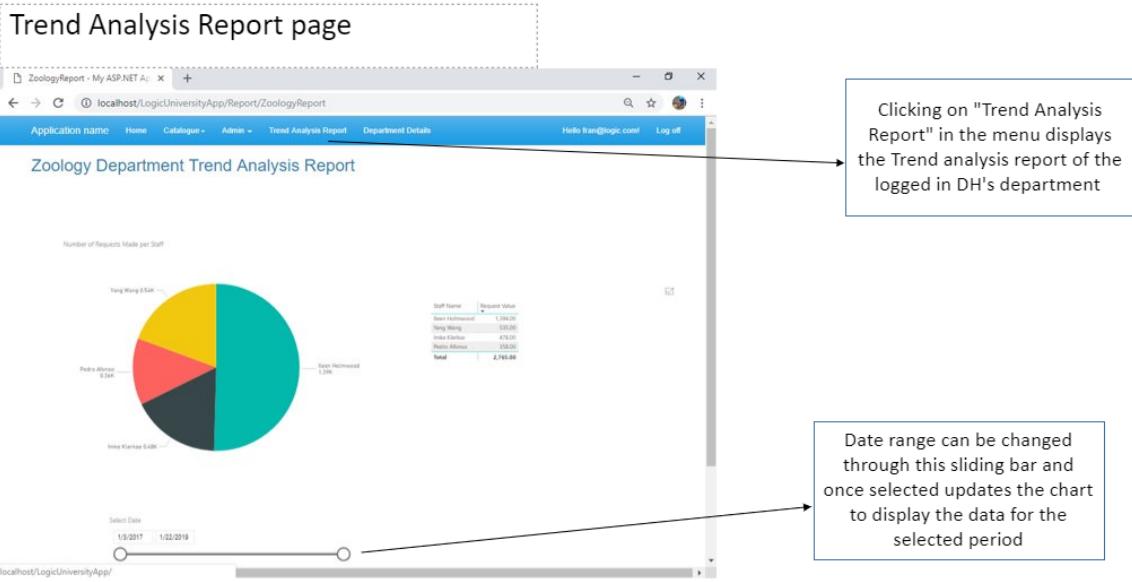
Category	Item Name	Price	Qty Need	Unit
Clip	Clips Double 3/4"	21.00	1	Dozen
Clip	Clips Paper Large	27.00	1	Box
Clip	Clips Double 1"	12.00	1	Dozen

Detail: request 1174

Back

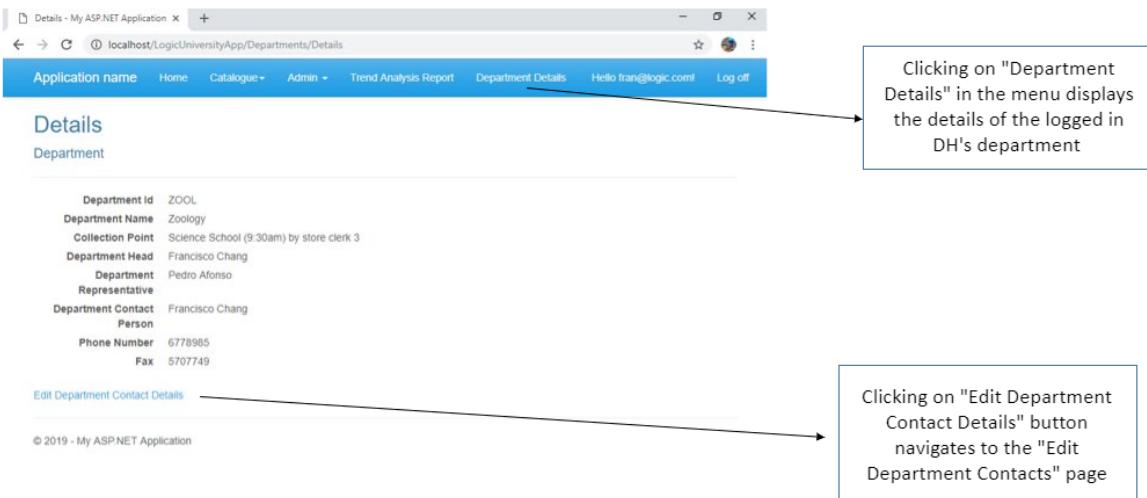
© 2019 - My ASP.NET Application

[View Department Trend Analysis Report- DH](#)



Modify Department Information

Department Details page



Edit Department Contacts page

The screenshot shows a web browser window titled "Edit - My ASP.NET Application". The URL is "localhost:LogicUniversityApp/Departments/Edit/ZOOL". The page header includes links for Application name, Home, Catalogue, Admin, Trend Analysis Report, Department Details, and Log off. The main content area is titled "Edit Department Contact Details" and displays a form for the department "Zoology". The form has four input fields: "Contact Person" (dropdown menu showing "Francisco Chang"), "Phone Number" (text input field containing "6778965"), "Fax" (text input field containing "5707749"), and a "Save" button. A copyright notice at the bottom left reads "© 2019 - My ASP.NET Application".

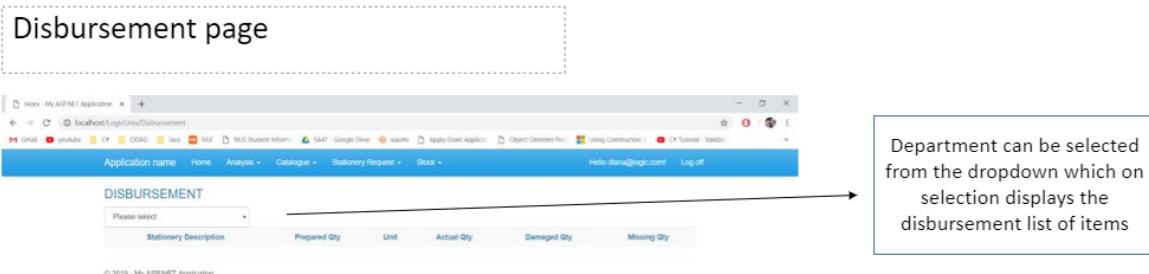
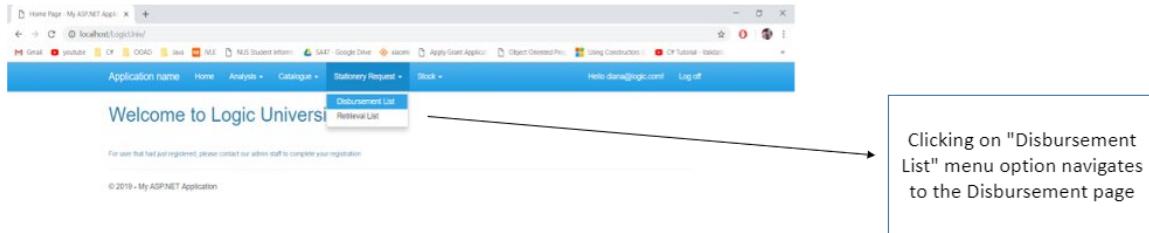
This page displays the editable department details

Clicking on "Save" after editing the fields saves the updated data in the database of the department and goes back to the Department Details page

10.2 Store web User interface

Process Disbursement List

Store Home page



Disbursement page

Stationery Description	Prepared Qty	Unit	Actual Qty	Damaged Qty	Missing Qty
Envelope Brown (x60)	25	Each	25	0	0
File-Brown w/o Logo	20	Each	20	0	0

On selecting the department, the disbursement item list of the department will be displayed

Clicking "submit" will remove the entry from the disbursement UI, and will change the status of the request to "finished"

View Stock Card

Store Home page

Welcome to Logic University!

For user that had just registered, please contact our admin staff to complete your registration.

© 2019 - My ASP.NET Application

Clicking on "Stock Card" menu option navigates to the "Stock Card" page

Stock Card page

This page displays the item list in the stock with the current balance of each item.

Individual item can be selected from the dropdown, on selecting displays the selected item details.

Clicking on "Select" link navigates to the Details page.

Item Number	Item Name	Store Balance	Action
C001	Clips Double 1"	230	Select
C002	Clips Double 2"	170	Select
C003	Clips Double 3/4"	240	Select
C004	Clips Paper Large	180	Select
C005	Clips Paper Medium	120	Select
C006	Clips Paper Small	180	Select
E001	Envelope Brown (3"x7")	1874	Select
E002	Envelope Brown (3"x7") w/ Window	2000	Select
E003	Envelope Brown (5"x7")	1600	Select
E004	Envelope Brown (5"x7") w/ Window	1200	Select
E005	Envelope White (3"x7")	1200	Select
E006	Envelope White (3"x7") w/ Window	1200	Select
E007	Envelope White (5"x7")	800	Select
E008	Envelope White (5"x7") w/ Window	800	Select

Stock Card page

On selecting the item from the dropdown displays the selected item details.

Clicking on "Select" link navigates to the Details page.

Item Number	Item Name	Store Balance	Action
E020	Eraser (hard)	80	Select
E021	Eraser (soft)	94	Select

Stock Card Details page

The screenshot shows a web browser window titled "Index" under the "Stock" category. The URL is `localhost:50101/StockCard/Index/E020`. The page displays a table with one row for item E020, which has a quantity of 70 and a balance of 70. A "Create New" link is visible at the top left of the table. The browser's address bar shows the full URL, and the top navigation bar includes links for Home, Analysis, Catalogue, Stationery Request, and Stock.

This page displays the stock details of the selected item

Clicking on "Create New" link navigates to the "Create Stock Card" page

Create Stock Card page

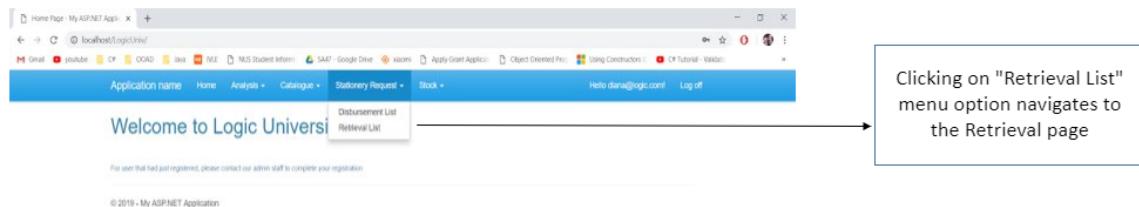
The screenshot shows a web browser window titled "Create" under the "Stock" category. The URL is `localhost:50101/StockCard/Create`. The page displays a form for creating a new stock card entry. The form fields include ItemNo (C001), Category (Clip), Description (Clips Double 1"), Balance (258), Reason (empty), Quantity (0), and Date (21/10/2019). A "Create" button is at the bottom right. The browser's address bar shows the full URL, and the top navigation bar includes links for Home, Analysis, Catalogue, Stationery Request, and Stock.

This page displays the fields for creating the stock card entry

Clicking on "Create" button after entering the details creates a new entry in the Stock card and goes back to the "Stock Card Details" page

Process Retrieval List

Store Home page



Clicking on "Retrieval List" menu option navigates to the Retrieval page

Retrieval page

The screenshot shows a web browser window titled "Index - My ASP.NET Application". The address bar shows the URL "localhost:LogicUniv/Retrieval". The navigation bar is identical to the home page. The main content area is titled "RETRIEVAL" and displays a table of items due. The table has columns: Item Name, Bin, Current Balance, Request Quantity, and Edit link. The table data is as follows:

Item Name	Bin	Current Balance	Request Quantity	
Clips Double 1"	A1	336	1	Edit
Clips Paper Large	A1	180	2	Edit
Envelope Brown (3"x6")	A2	1974	1	Edit
Envelope Brown (3"x6") w Window	A2	2000	70	Edit
Eraser (hard)	A3	80	1	Edit
Exercise Book (120 pg)	A4	200	1	Edit
File-Brown w/o Logo	A5	429	1	Edit
Highlighter Blue	A6	238	3	Edit
Highlighter Pink	A6	160	1	Edit
Highlighter Yellow	A6	160	3	Edit
Hole Puncher 3 holes	A7	80	4	Edit
Paper Photostat A3	A9	2000	30	Edit
Paper Photostat A4	A9	1500	50	Edit
Pencil 2B	A6	250	4	Edit

This page displays the retrieval list of items due until the current date irrespective of the department

This Due date field displays the current date by default

Clicking on "Edit" link navigates to the "Retrieval Details" page

Retrieval Details page

This page displays the details of the selected retrieval item

Retrieval quantity is auto populated based on the Request quantity and Current balance. If the balance is less than the request quantity, then the Retrieval quantity is set to the current balance by default

Instore damage and Instore missing fields can be updated if any

Clicking on "Submit" button saves the updated data and navigates to the "Retrieval" page and the record is removed from the UI

Modify Supplier Information:

Supplier List Page

Clicking on Supplier Data will open the supplier details page where the store clerk can add new suppliers or view & modify existing supplier details

This page displays the list of suppliers

Clicking on "Edit" navigates to the "Edit Supplier" page

Clicking on "Delete" navigates to the "Delete Supplier Details" page

Clicking on "Details" navigates to the "Supplier Details" page

Clicking on "Create New" will navigate to "Create New Supplier Page"

Create New Supplier Page

The screenshot shows a web browser window titled "Create - My ASP.NET Application". The URL is "localhost:1050/LogIn/Views/Suppliers/Create". The page has a header with links like "Home", "Analytics", "Catalogue", "Stationery Request", "Stock", "Supplier Data", "Hello info@logiplus.com", and "Log off". The main content area is titled "Create a New Supplier" under "Supplier". It contains input fields for "Supplier ID", "Supplier Name", "Address", "Contact Name", "Phone Number", "Fax", and "Email". Below these is a "Create" button. At the bottom left is a "Back To List" link, and at the bottom right is a copyright notice "© 2019 - My ASP.NET Application".

Clicking on "Create New" will open the Create a New Supplier Page

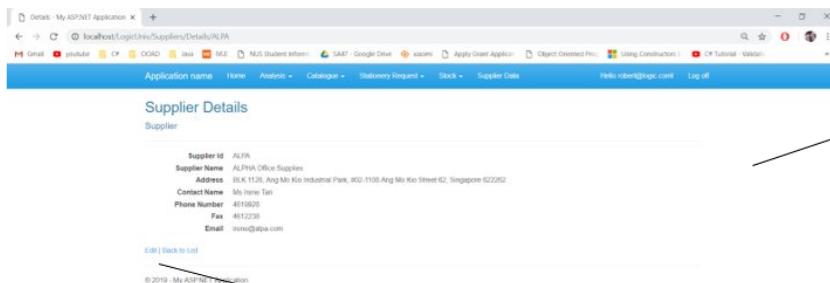
On clicking the create button, after entering the required supplier details, a supplier record is created in the Supplier List Page

Edit Supplier Page

The screenshot shows a web browser window titled "Edit - My ASP.NET Application". The URL is "localhost:1050/LogIn/Views/Suppliers/Edit/101A/UR". The page has a header with links like "Home", "Analytics", "Catalogue", "Stationery Request", "Stock", "Supplier Data", "Hello info@logiplus.com", and "Log off". The main content area is titled "Edit Supplier" under "Supplier". It contains input fields for "Supplier Name" (set to "ALPHA Office Supplies"), "Address" (set to "BLK 102B, Ang Mo Kio Industrial Park, #02-1108 Ang Mo Kio Street 62, Singapore 522092"), "Contact Name" (set to "Ms Irene Tan"), "Phone Number" (set to "40119625"), "Fax" (set to "40122356"), and "Email" (set to "irene@aya.com"). Below these is a "Save" button. At the bottom left is a "Back To List" link, and at the bottom right is a copyright notice "© 2019 - My ASP.NET Application".

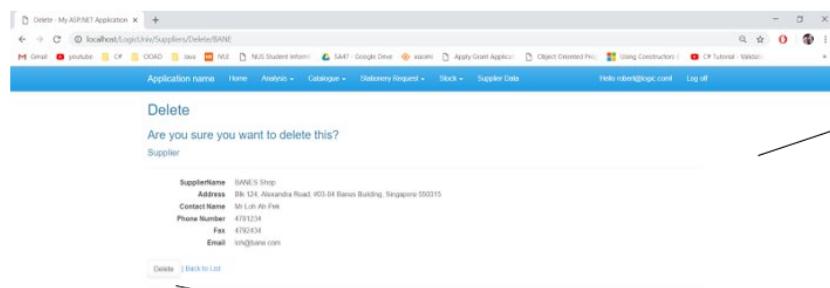
Clicking on "Edit" button on the Supplier List Page will open "Edit Supplier" page, where existing supplier details can be edited and saved

View Supplier Details Page



Clicking on "Details" button on the Supplier List Page will open the "Supplier Details" Page

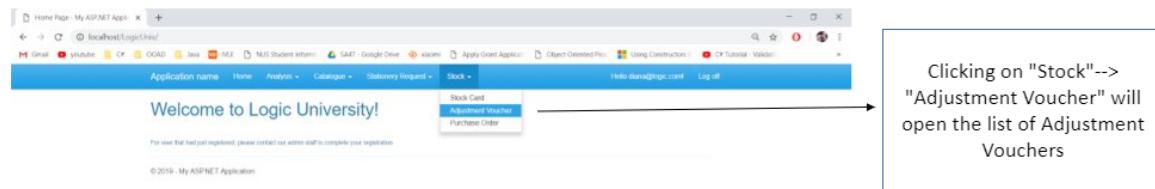
Delete Supplier Details Page



Clicking on "Delete" button on the Supplier List Page will open the "Delete Supplier" page

Issue Adjustment Voucher

Adjustment Voucher Menu



localhost:LogicUniversity/AdjustmentVouchers/ViewTotal

View Adjustment Voucher Page

The screenshot shows a web browser window titled "ViewTotal - My ASP.NET App". The address bar says "localhost:LogicUniversity/AdjustmentVouchers/ViewTotal". The page header includes links for "Application name", "Home", "Analysis", "Catalogue", "Stationery Request", "Stock", "Hello dinesh@logic.com", and "Log off". A dropdown menu under "Stock" is visible. A search bar with the placeholder "Search" is present. Below it, a table lists adjustment vouchers with columns for "Item Name" and "Total Amount". A callout box points to the search bar with the text: "This option allows to search for specific month's Adjustment Vouchers". Another callout box points to the table with the text: "The list of vouchers whose total amount is greater than 250, will be highlighted in red. All the adjustment vouchers are submitted to the Supervisor in order to be Issued to Finance". A third callout box points to a "Details" link in the table with the text: "Clicking on \"Details\" will go to the \"Supplier Details\" page".

Item Name	Total Amount	Details
Stapler No. 36	1040.00	Details
Highlighter Yellow	252.00	Details
Highlighter Blue	216.00	Details
Highlighter Green	180.00	Details
Clips Paper Large	135.00	Details
Paper Photostat A4	110.00	Details
Shorthand Book (100 pg)	78.00	Details
File Separator	64.00	Details
Paper Photostat A3	60.00	Details
Scotch Tape Dispenser	60.00	Details
Clips Paper Medium	48.00	Details
Ruler 12"	40.00	Details
Clips Double "	36.00	Details

Search Adjustment Voucher Page

The screenshot shows a web application window titled "ViewDetail - My ASP.NET App". The URL is localhost:1395/Views/AdjustmentVouchers/ViewTotal?year=2018&month=1. The page has a header with links like Application name, Home, Analytics, Catalogue, Stationery Request, Stock, Help, and Log off. Below the header, there's a section titled "ADJUSTMENT VOUCHER" with dropdown menus for "Create New", "Year" (set to 2018), and "Month" (set to January). A "Search" button is present. The main content area displays a table with columns: Item Name, Total Amount, and Details. One row is visible: "Stapler No. 36" with a total amount of 400.00 and a "Details" link. At the bottom left, it says "© 2018 - My ASP.NET Application".

The Adjustment list can be filtered on choosing the year and month from the dropdown

Search Adjustment Voucher Page

The screenshot shows a web application window titled "ViewDetail - My ASP.NET App". The URL is localhost:1395/Views/AdjustmentVouchers/ViewDetail?ItemNo=PO30&year=2018&month=1. The page has a header with links like Application name, Home, Analytics, Catalogue, Stationery Request, Stock, Help, and Log off. Below the header, there's a section titled "ViewDetail" with a table showing details for a single item. The table has columns: Voucher ID, Quantity, Remark, Date, Items, Unit Price, Item Name, and Total Amount. One row is visible: "24" with "1", "Damage", "8/1/2018 12:00:00 AM", "PO30", "8.00", "Pen Whiteboard Marker Black", and "8.00". At the bottom left, it says "© 2018 - My ASP.NET Application".

Clicking on "Details" in the View Adjustment Voucher Page will display the details of the individual items in the Adjustment Voucher

Create New Adjustment Voucher Page

The screenshot shows a web application window titled "Create - My ASP.NET Application". The URL is "localhost:1024/LogicUniversity/AdjustmentVoucher/Create". The page has a header with links like Application name, Home, Analysis, Catalogue, Stationery Request, Stock, and Supplier Data. It also shows "Hello robert@logic.com" and "Log off". The main content area is titled "Create Adjustment/Voucher" and contains the following form fields:

- Category: Select Category (dropdown menu)
- Item Name: (text input field)
- Quantity: (text input field)
- Date: 21/1/2019 (date input field)
- Remark:
 - Damaged
 - Missing
 - Free of Charge
 - Wrong Input
- Add Item: (button)

Below the form are "Back to List" and "© 2019 - My ASP.NET Application" links.

Clicking on "Create New" in the View Adjustment Voucher Page will open "Create New Adjustment Voucher" page, where new item can be added

Issue Adjustment Voucher – Manager and Supervisor

Create New Adjustment Voucher PDF Page

The screenshot shows a web application window titled "Logic University". The URL is "localhost:1024/LogicUniversity/AdjustmentVoucher/Print". The page has a header with links like Home, Analysis, Catalogue, Stationery Request, Stock, and Supplier Data. It also shows "Hello robert@logic.com" and "Log off". The main content area is titled "ADJUSTMENT VOUCHER" and contains the following form fields:

Item Name	Total Amount	Details
Highlighter Blue	\$72.00	Details
Clips Double 1"	\$2.00	Details
Clips Paper Small	\$2.00	Details
Scissors	\$2.00	Details
Envelope Brown (37x67)	\$0.70	Details

Below the table is a "Create Adjustment Voucher PDF" button. A modal dialog box is shown at the bottom with the message "What do you want to do with AdjustmentVoucher(1-2019).pdf? From: localhost" and buttons for Open, Save, Cancel, and Close.

Clicking on "Create Adjustment Voucher PDF" will generate Adjustment Voucher for the selected month

Adjustment Voucher in Crystal Report

ADJUSTMENT VOUCHER

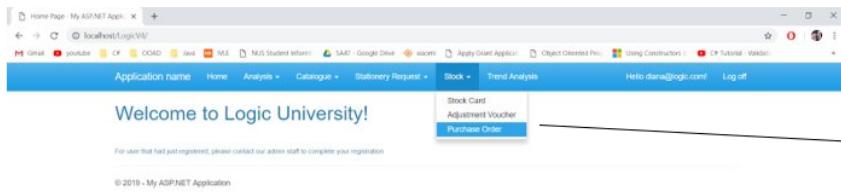
Voucher: 1 / 2019 Date Issued: 9/2/2019

Date	ItemNo	Description	Quantity	StdPrice	Value	Remark
01/21/2019	S100	Scissors	-1	2.00	-2.00	instore damaged
01/21/2019	C006	Clips Paper Small	-1	2.00	-2.00	instore damaged
01/30/2019	C001	Clips Double 1"	-3	2.00	-6.00	instore missing
01/30/2019	H011	Highlighter Blue	-2	36.00	-72.00	instore damaged
01/31/2019	E001	Envelope Brown (3"x6")	-1	0.70	-0.70	instore missing
01/31/2019	C001	Clips Double 1"	-1	2.00	-2.00	damaged at collection point
						Total Value -84.70

Sample of the Adjustment Voucher(crystal report) after you selected "Create Adjustment Voucher PDF"

Process Purchase Order

Purchase Order Menu



Clicking on "Stock"
-->"Purchase Order" will open the "Purchase Order List Page"

Purchase Order List Page

This page will list all the purchase orders which are in status- open, sent and closed

Clicking on "Details" will go to "Purchase Order Details" page

Clicking on "Create New PO" will navigate to "Create New PO Page"

Clicking on "View Report" will go Purchase Order Report page

Create New PO Page

Clicking on "Create New" in the "Purchase Order List" page will open "Create/Edit Purchase Order" page

Clicking on "Cart" button will open the "Cart" page

Clicking on "+" button will add the specific item to the cart

Purchase Order Cart Page

The screenshot shows a web application window titled "POCart - My ASP.NET Application". The URL is "localhost:50101/LogIn/PurchaseOrders/POCart". The page has a header with links like "Home", "Analytics", "Catalogue", "Stationery Request", "Stock", "Trend Analysis", "Hello diana@logic.com", and "Log off". Below the header, the title "Purchase Order Cart" and "New PO" are displayed. There are input fields for "Order Date" (2/1/2019) and "Delivery Date" (2/1/2019). A table lists items with columns: Item Description, Category, Supplier Name, Tender Price, and Quantity. Two rows are shown: "Clips Double 1*" (Category: Clip, Supplier: ALPHA Office Supplies, Price: 12.00, Quantity: 30) and "Clips Double 1*" (Category: Clip, Supplier: Cheap Stationer, Price: 13.00, Quantity: 30). Each row has a "Delete Item" button. At the bottom left is a "Submit" button, and at the bottom center is a copyright notice: "© 2019 - My ASP.NET Application".

This page will list all the items added to cart using "+" button in the "Create/Edit Purchase Order" page

Clicking on "Add Item" will open the "Create/Edit Purchase Order" page from where items can be added to cart

Clicking on "Submit" button will create a new record in the "Purchase Order List" page with status "Open"

Purchase Order Details Page

The screenshot shows a web application window titled "Details - My ASP.NET Application". The URL is "localhost:50101/LogIn/Details/PurchaseOrders/Detail?Id=174". The page has a header with links like "Home", "Analytics", "Catalogue", "Stationery Request", "Stock", "Tender Price", "Order Date", and "Delivery Date", along with "Hello diana@logic.com" and "Log off". Below the header, the title "PO 174 Purchase Order Details" is displayed. A table shows the following data:

Supplier ID	Category	Item Number	Item Description	Quantity	Tender Price	Order Date	Delivery Date
CHEP	Clip	0001	Clips Double 1"	30	13.00	2/1/2019 12:00:00 AM	2/1/2019 12:00:00 AM

At the bottom left is a "Back to List" link, and at the bottom center is a copyright notice: "© 2019 - My ASP.NET Application".

On clicking the "Details" button in the "Purchase Order List" page, "Purchase Order Details" page is opened

Edit Purchase Order Page- Open to Sent

PO 148

Order Date: 2/1/2019

Delivery Date: 2/1/2019

Status: Open Sent

Item Description	Category	Supplier Name	Tender Price	Quantity
Folder Plastic Green	File	Dollar stationer	2.70	150

[Delete Item](#)

[Submit](#)

© 2019 - My ASP.NET Application

On clicking the "Edit PO" button in the "Purchase Order List" page for the "Open" POs, it opens the "Purchase Order Cart",

Radio buttons to change status from "Open" to "Sent"

Clicking on "Add item" allows to add new items to the cart

On changing status from "Open" to "Sent", the PO gets submitted to the respective suppliers when the submit button is clicked.

Edit Purchase Order Page- Sent to closed

PO 167

Order Date: 2/1/2019

Delivery Date: 2/1/2019

Status: Sent Closed

Item Description	Category	Supplier Name	Tender Price	Quantity
Trays InOut	Tray	From office store	10.00	10

[Delete Item](#)

[Submit](#)

© 2019 - My ASP.NET Application

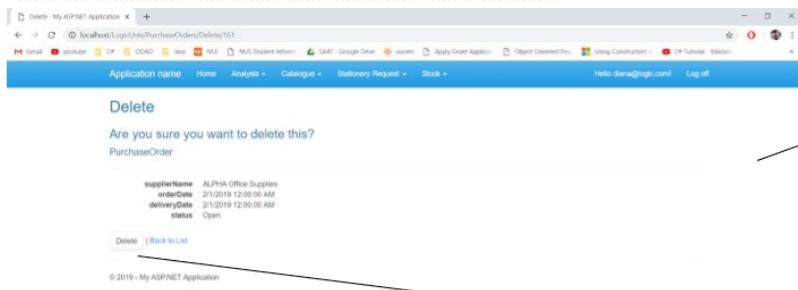
On clicking the "Edit PO" button in the "Purchase Order List" page for the "Open" POs, it opens the "Purchase Order Cart",

Radio Button to change status from "Sent" to "Closed"

Clicking on "Add item" allows to add new items to the cart

On changing status from "Sent" to "Closed", the PO's status gets closed when the submit button is clicked.

Delete Purchase Order Page



On clicking the "Delete" button in the "Purchase Order List", it opens the "Delete Purchase Order" page

View Report Purchase Order Page

	Supplier Name	Order Date	Delivery Date	Status	Details View Report
18	Hexagonal stationer	1/4/2017	4/4/2017	Closed	Details View Report
17	OMEGA Stationery Supplier	27/3/2017	30/3/2017	Closed	Details View Report
16	Hexagonal stationer	22/3/2017	25/3/2017	Closed	Details View Report
15	Popular Stationery shop	17/3/2017	29/3/2017	Closed	Details View Report
14	Hexagonal stationer	12/3/2017	15/3/2017	Closed	Details View Report
13	TOTO supplies store	7/3/2017	10/3/2017	Closed	Details View Report
12	OMEGA Stationery Supplier	2/3/2017	5/3/2017	Closed	Details View Report
11	OMEGA Stationery Supplier	25/2/2017	28/2/2017	Closed	Details View Report
10	Kinokuniya book store	20/2/2017	23/2/2017	Closed	Details View Report
9	Yeye official store	15/2/2017	18/2/2017	Closed	Details View Report
8	TOTO supplies store	10/2/2017	13/2/2017	Closed	Details View Report
7	Cheap Stationer	5/2/2017	8/2/2017	Closed	Details View Report
6	Hexagonal stationer	31/1/2017	3/2/2017	Closed	Details View Report
5	Yeye official store	26/1/2017	29/1/2017	Closed	Details View Report
4	TOTO supplies store	21/1/2017	24/1/2017	sent	Details Edit PO
3	ALPHA Office Supplies	16/1/2017	19/1/2017	sent	Details Edit PO
2	ALPHA Office Supplies	11/1/2017	14/1/2017	Closed	Details View Report
1	BANES Shop	6/1/2017	9/1/2017	Closed	Details View Report

On clicking the "View Report" button in the "Purchase Order List", it opens the purchase order in crystal report format



Purchase Order in Crystal Report

Supplier:
BANES
BANES Shop
Address:
B# 124, Alexandra Road,
#03-04 Banes Building,
Singapore 550315

Contact Person:
Mr Loh Ah Pek

Office Number: 4781234 **Email Address:** loh@lune.com

Purchase Order Number: 1 **Order Date:** 01/06/2017

ItemNo	Description	Quantity	Unit	tenderPrice	Value
C001	Clips Double 1"	6	Dozen	11.00	\$66.00
C002	Clips Double 2"	30	Dozen	18.50	\$555.00
C003	Clips Double 3/4"	30	Dozen	24.00	\$720.00
C004	Clips Paper Large	30	Box	26.30	\$789.00

Total Value \$2,130.00

Please supply following items by: 01/09/2017

Ordered by: _____
Approved by: _____

Sample of a purchase order(crystal report) after you selected "View Report"

Modify Stationery Catalogue

Store Home page

Application name Home Analysis Catalogue Stationery Request Stock

Maintain Item

Welcome to Logic University!

For users that has just registered, please contact our admin staff to complete your registration.

© 2019 - My ASP.NET Application

Clicking on "Maintain Item" in the menu navigates to the "Inventories" page which displays the list of items in the inventory with details

localhost/LogicUniversity/inventories

Inventories page

Index

Item Number	Category	Item Name	Reorder Level	Reorder Quantity	Unit	Action
C001	Clip	Clips Double 1"	50	30	Dozen	Edit Details Delete
C002	Clip	Clips Double 2"	50	30	Dozen	Edit Details Delete
C003	Clip	Clips Double 3/4"	50	30	Dozen	Edit Details Delete
C004	Clip	Clips Paper Large	50	30	Box	Edit Details Delete
C005	Clip	Clips Paper Medium	50	30	Box	Edit Details Delete
C006	Clip	Clips Paper Small	50	30	Box	Edit Details Delete
E001	Envelope	Envelope Brown (3"x6")	600	400	Each	Edit Details Delete
E002	Envelope	Envelope Brown (3"x6") w/ Window	600	400	Each	Edit Details Delete
E003	Envelope	Envelope Brown (5"x7")	600	400	Each	Edit Details Delete
E004	Envelope	Envelope Brown (5"x7") w/ Window	600	400	Each	Edit Details Delete
E005	Envelope	Envelope White (3"x6")	600	400	Each	Edit Details Delete
E006	Envelope	Envelope White (3"x6") w/ Window	600	400	Each	Edit Details Delete
E007	Envelope	Envelope White (5"x7")	600	400	Each	Edit Details Delete
E008	Envelope	Envelope White (5"x7") w/ Window	600	400	Each	Edit Details Delete

Annotations:

- Clicking on "Create New" link navigates to the "Create Inventory" page
- Category can be selected from the dropdown list, on selection displays the item list as per the selected category
- Substring search can be performed by entering in the "Title" field and clicking on "Search" button displays the list of items matching the string entered
- Clicking on "Delete" navigates to the Delete Inventory Details page
- Clicking on "Details" navigates to the "Inventory Details" page
- Clicking on "Edit" navigates to the "Edit Inventory Details" page

Inventories page

Index

Item Number	Category	Item Name	Reorder Level	Reorder Quantity	Unit	Action
P020	Paper	Paper Photostat A3	500	500	Box	Edit Details Delete
P021	Paper	Paper Photostat A4	500	500	Box	Edit Details Delete

This page displays the item list based on the category selected from the dropdown

Create Inventory page

This page displays the editable fields to create a new item in the inventory

Category can be selected from the dropdown

After selecting the category, the item details can be entered in the displayed fields

Clicking on "Create" button after entering the item details creates a new entry in the inventory for the new item and goes back to the "Inventories" page

Edit Inventory Details page

This page displays the editable fields of the item and supplier details of the selected inventory item

Clicking on "Save" button after editing the fields saves the updated data in the inventory for the selected item and goes back to the "Inventories" page

Inventory Details page

The screenshot shows a web browser window titled "Details - My ASP.NET Application". The URL is "localhost:4201/Logi/Univ/inventories/Details/C001". The page has a header with "Application name", "Home", "Analysis", "Catalogue", "Inventory Request", "Stock", "Hello daria@logi.com", and "Log off". Below the header, there's a "Details" section for "InventoryViewModel". It displays the following data:

Item Number	C001
Category	Clip
Item Name	Clips Double 1"
Store Balance	339
Reorder Level	50
Reorder Quantity	20
Unit	Dozen
Price	12.00
Supplier 1	ALPA
Supplier 2	CHEP
Supplier 3	BANE

At the bottom left is a "Back to List" link, and at the bottom right is a copyright notice: "© 2019 - My ASP.NET Application".

This page displays item and supplier details of the selected inventory item

Delete Inventory Details page

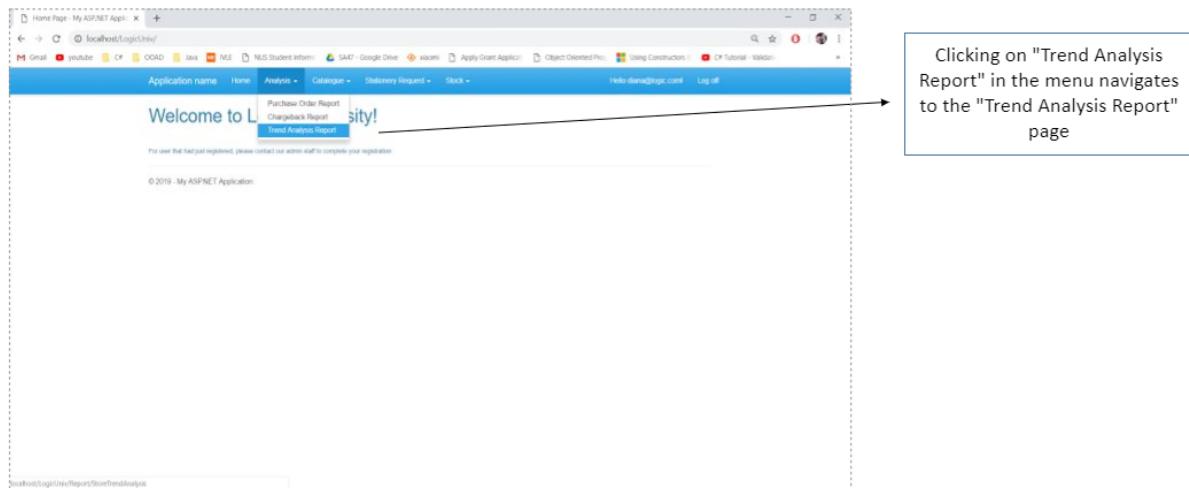
The screenshot shows a web browser window titled "Delete - My ASP.NET Application". The URL is "localhost:4201/Logi/Univ/inventories/Delete/C001". The page has a header with "Application name", "Home", "Analysis", "Catalogue", "Inventory Request", "Stock", "Hello daria@logi.com", and "Log off". Below the header, there's a "Delete" section asking "Are you sure you want to delete this?". It lists the same item details as the previous page. At the bottom is a "Delete" button with the text "I'm sure I want to delete this item".

This page requests user confirmation to delete the particular Inventory item.

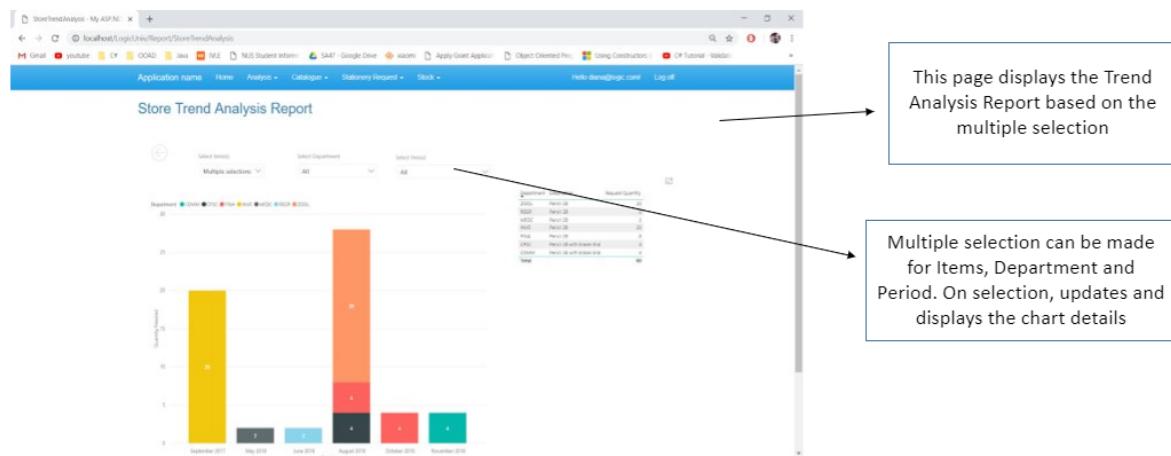
Clicking the "Delete" button deleted the inventory item from the inventory list

View Reports

Store Home page



Trend Analysis Report page



Generate Charge-back Report

Store Home page

The screenshot shows a web browser window titled "Home Page - My ASP.NET App". The address bar says "localhost:LogOn". The page has a blue header with the text "Welcome to LogOn Store!". Below the header is a menu bar with items like "Purchase Order Report", "Chargeback Report", and "Trend Analysis Report". A sidebar on the left says "For user that had just registered, please contact our admin staff to complete your registration." At the bottom, it says "© 2010 - My ASP.NET Application".

Clicking on "Chargeback Report" in the menu navigates to the "Chargeback Report" page

<http://localhost:LogOnReport/ChargebackReport>

Chargeback Report page

The screenshot shows a web browser window titled "Chargeback Report - My ASP.NET". The address bar says "localhost:LogOnReport/ChargebackReport". The page has a blue header with the text "Charge Back Report". Below the header is a table titled "Chargeback Report Per Department". The table has columns for Category, Arts, Business, Commerce, Computer Science, Digital, Finance, Inventory, Marketing, Medicine, Religion, and Total. The table shows data for various departments across different years (2010, 2011, 2012, 2013). A sidebar on the left shows a year selection dropdown with options 2010, 2011, 2012, 2013, and 2014. At the bottom, there is a Microsoft Power BI download link.

This page displays the amount details of the items used by various departments based on year selection

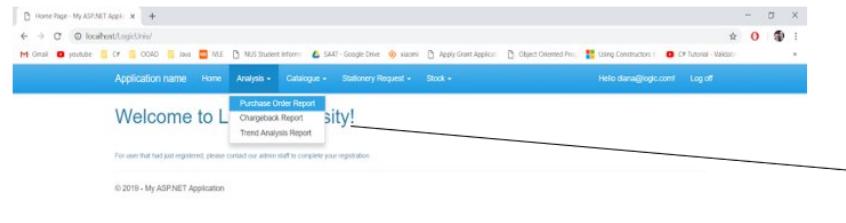
Year selection can be made using the checkbox

This column displays the department wise total for each item

This row displays the item wise total for each department

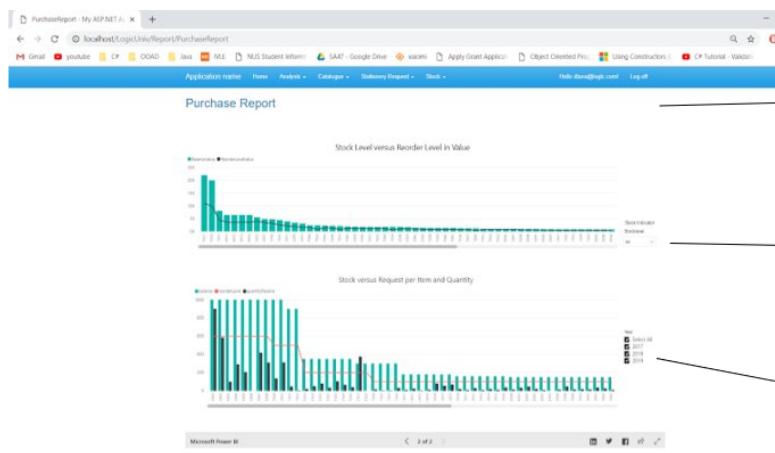
View Reorder List

Request List page



Clicking on "Purchase Order Report" in the menu navigates to the "View Reorder List" page

View Reorder List page

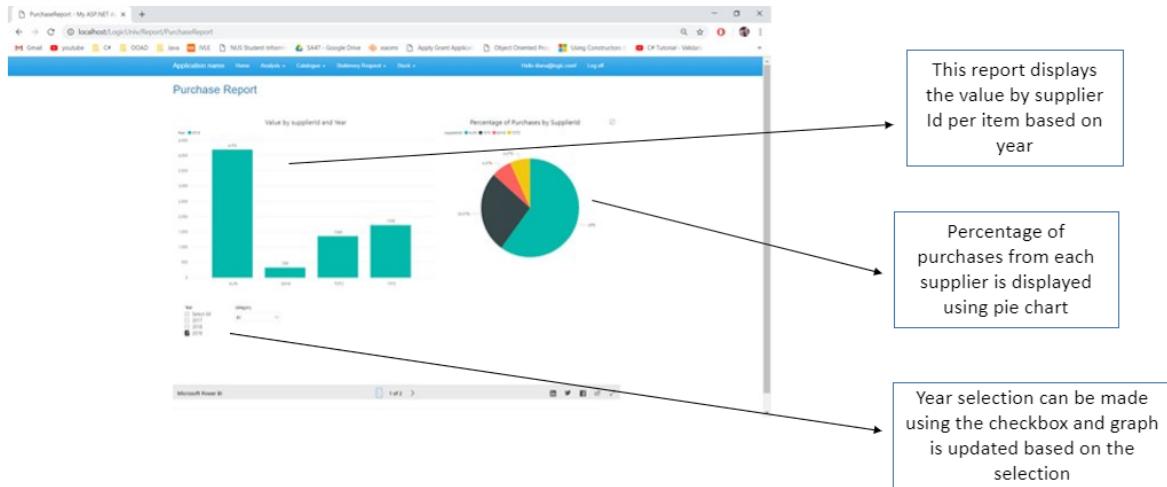


This report displays the comparison stock level with the reorder level and request per item and quantity

Stock level can be selected from the dropdown and graph is updated based on the selection

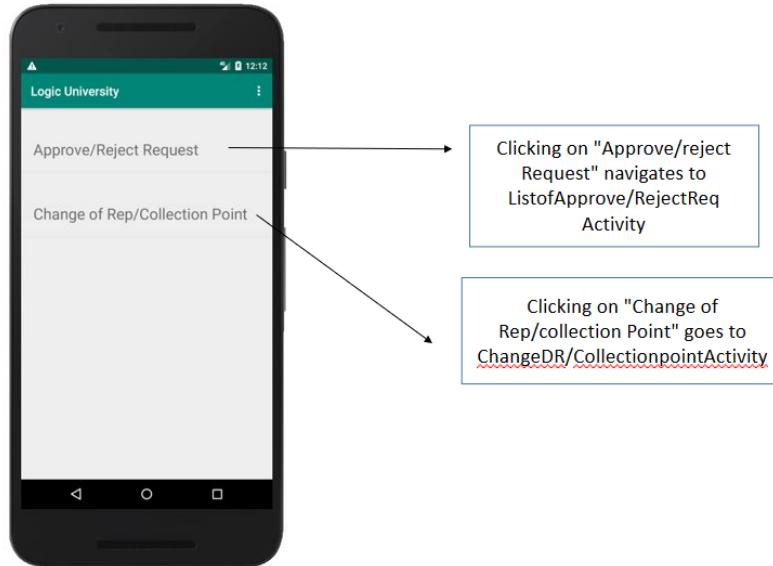
Year selection can be made using the checkbox and graph is updated based on the selection

View Reorder List page



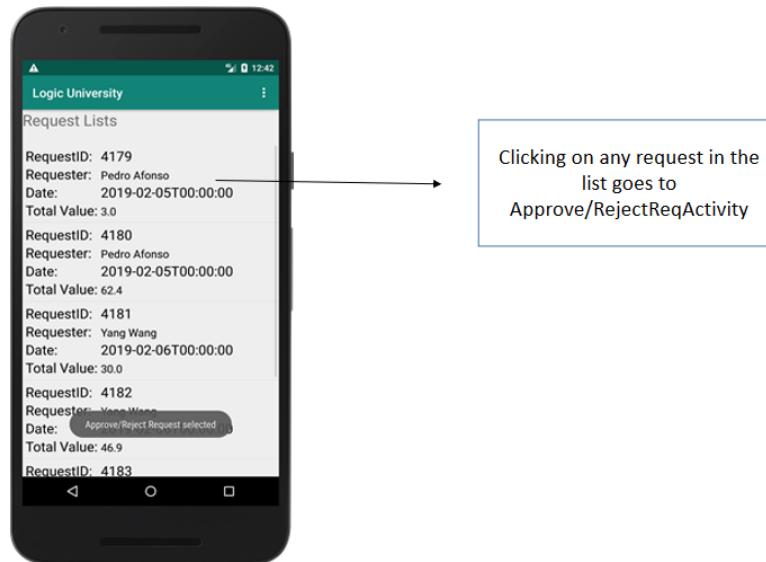
10.3 Department Mobile User Interface

DH – Home page

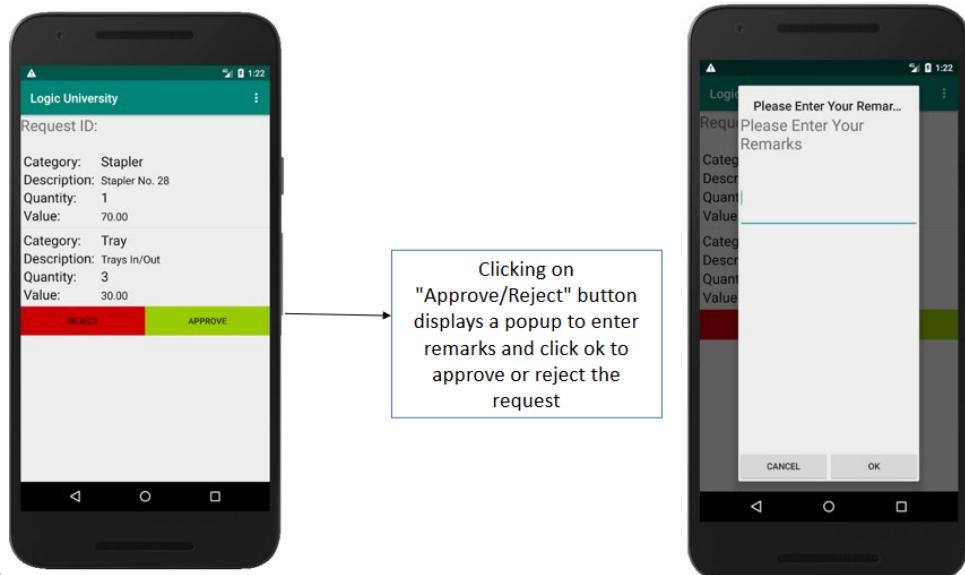


Approve/Reject Request

ListofApprove/RejectReq Activity

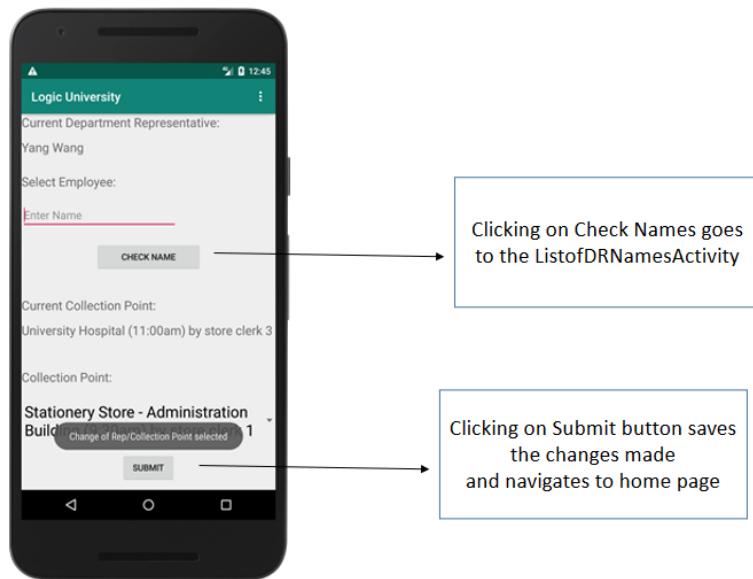


Approve/RejectReqActivity

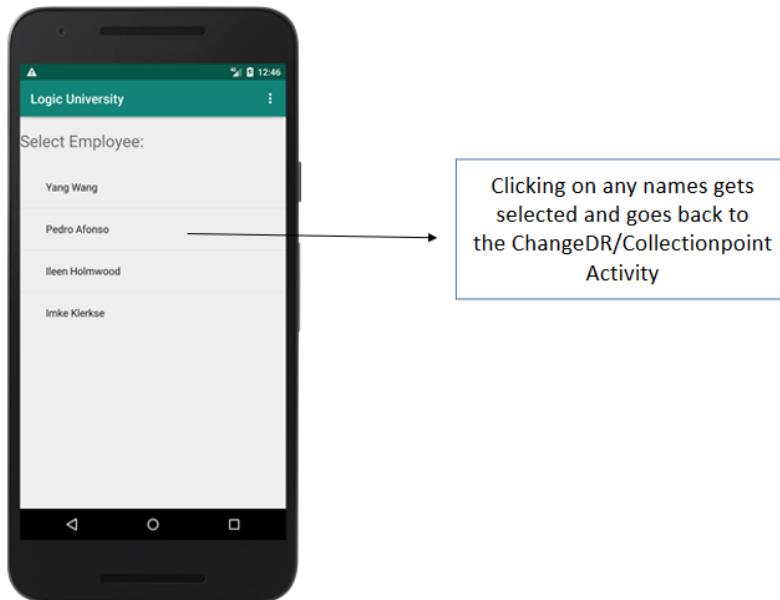


Change of Rep/Collection Point

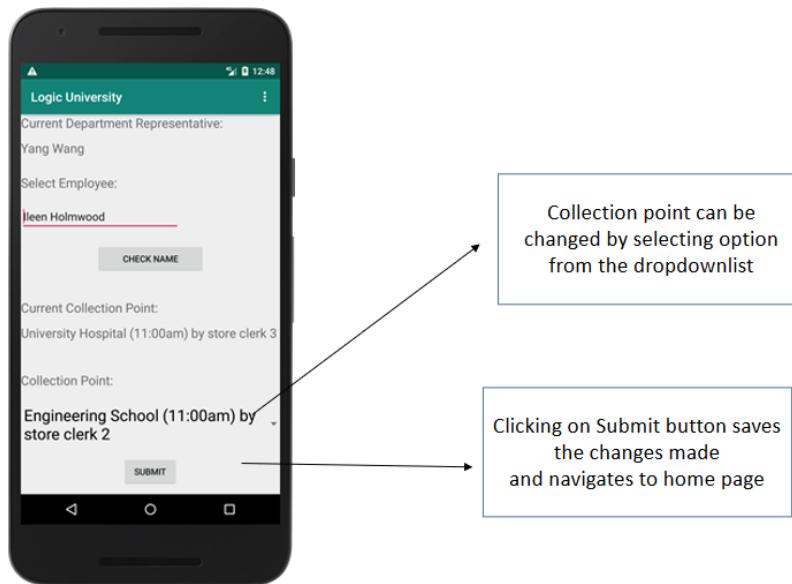
ChangeDR/CollectionpointActivity



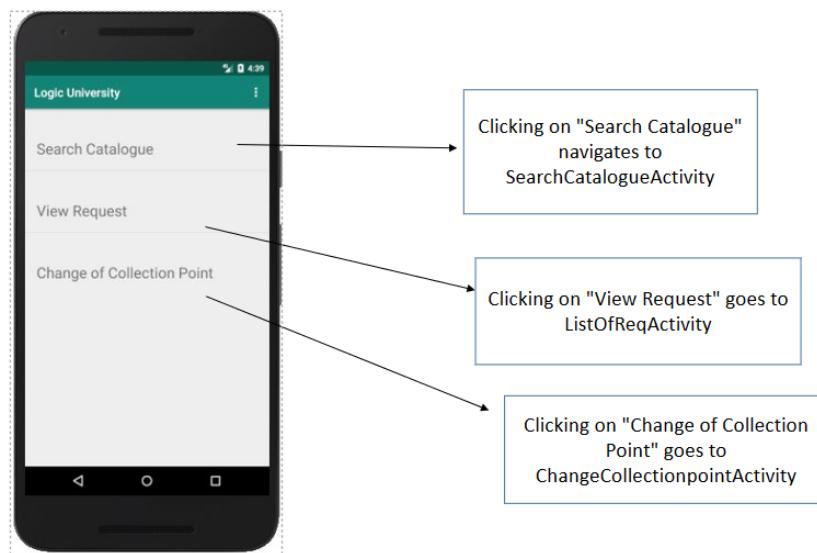
ListofDRNamesActivity



ChangeDR/CollectionpointActivity

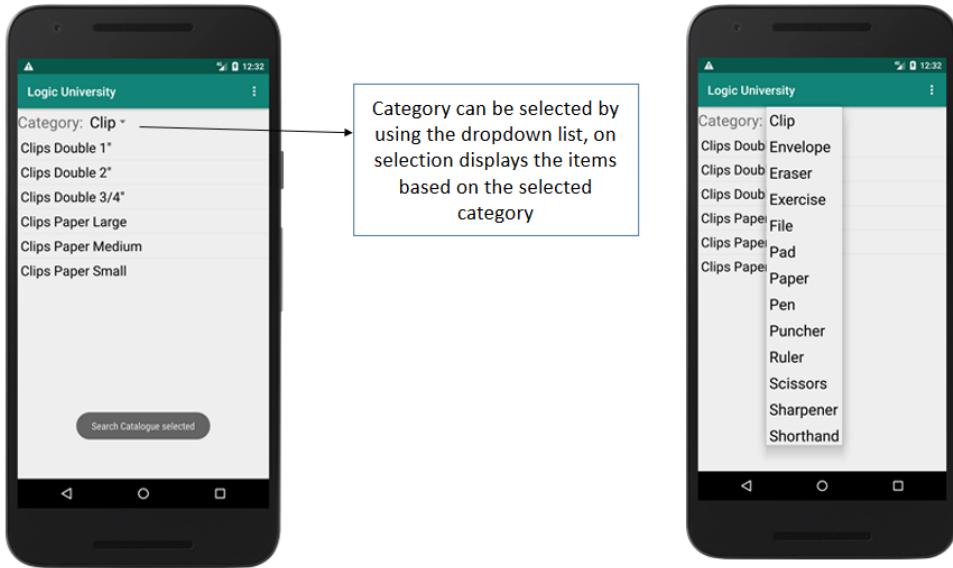


DR – Home page



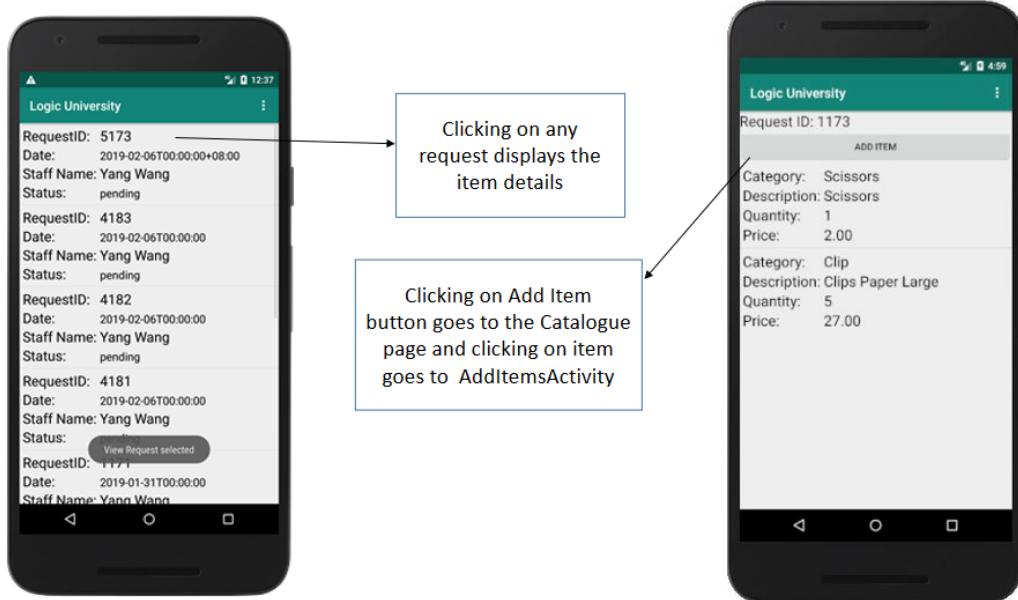
Search Catalogue

SearchCatalogueActivity

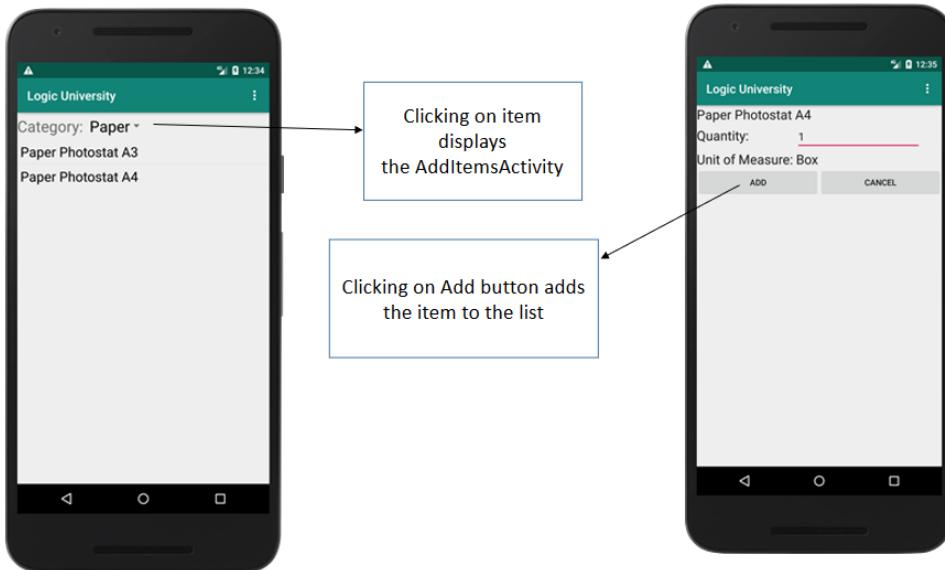


View Request

ListofReqActivity

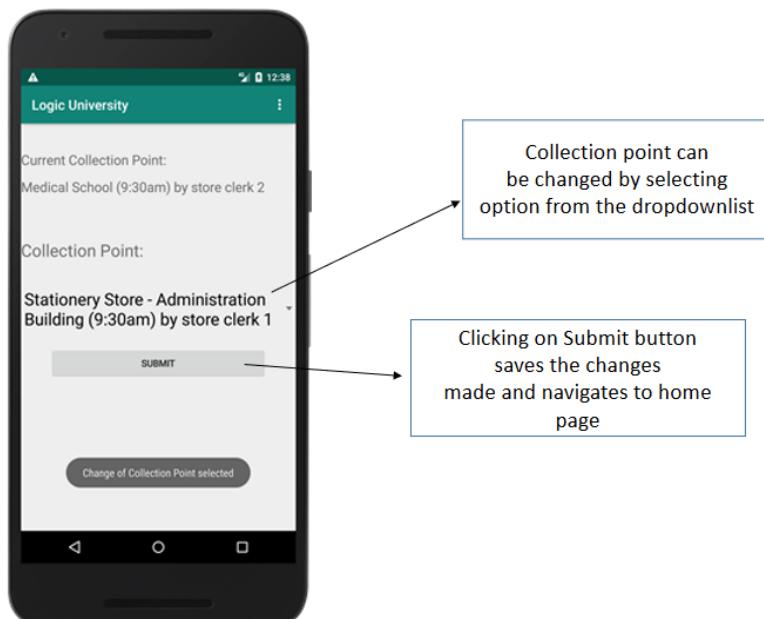


AddItemsActivity

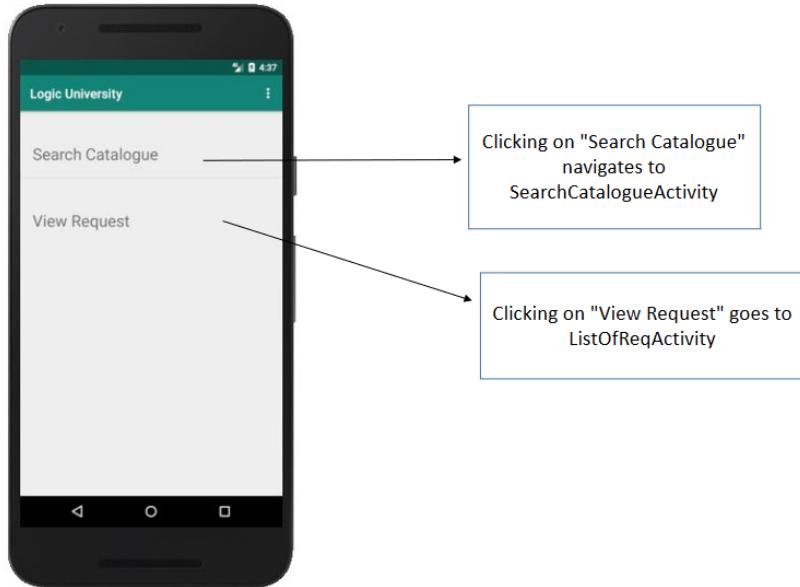


Change Collection Point - DR

ChangeCollectionPointActivity

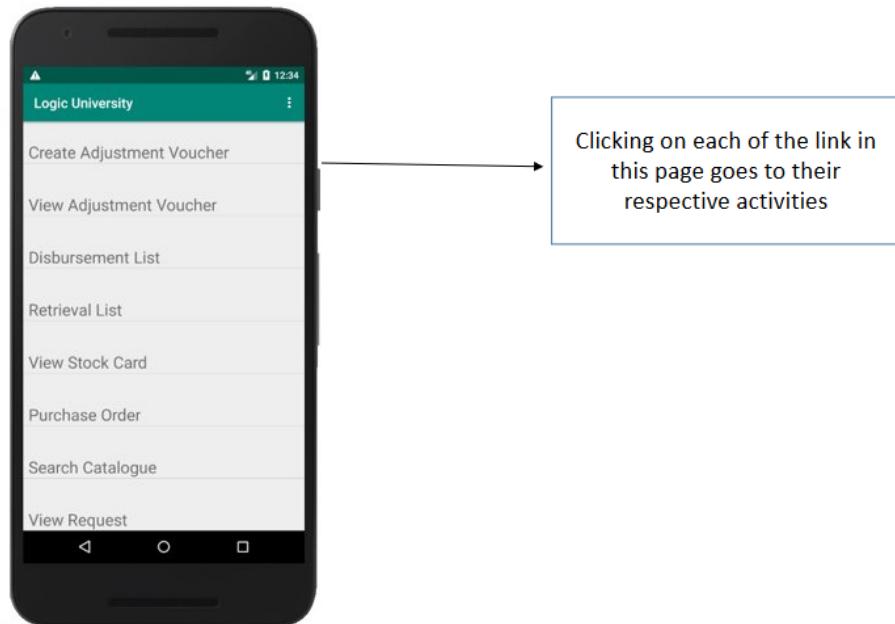


Department Staff – Home page



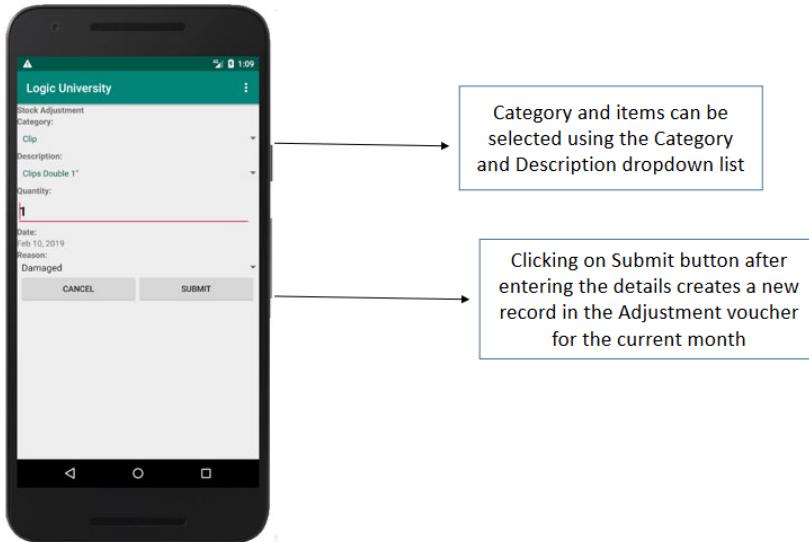
10.4 Store Mobile User Interface

Store Clerk – Home page

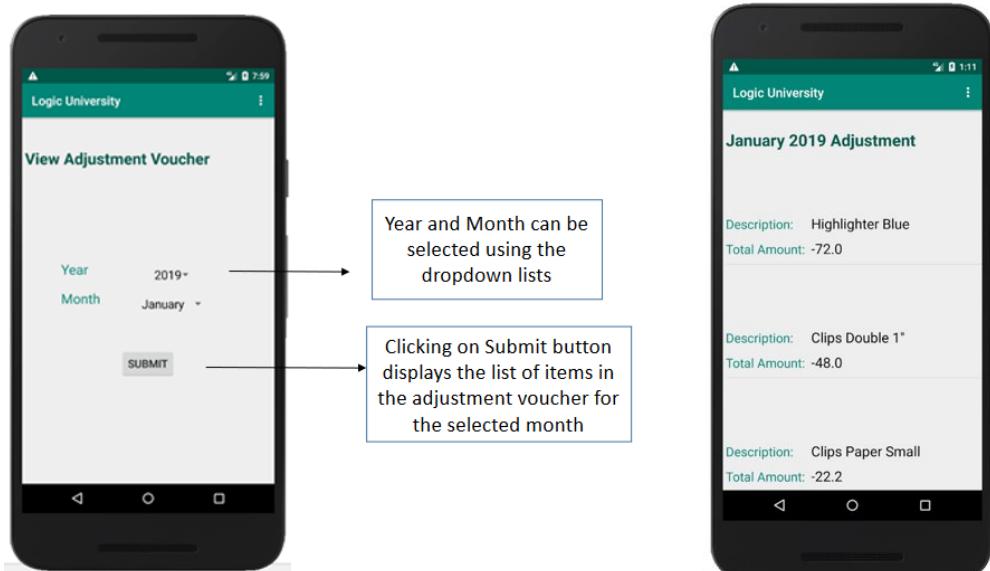


Process Adjustment Voucher

CreateAdjVoucherActivity

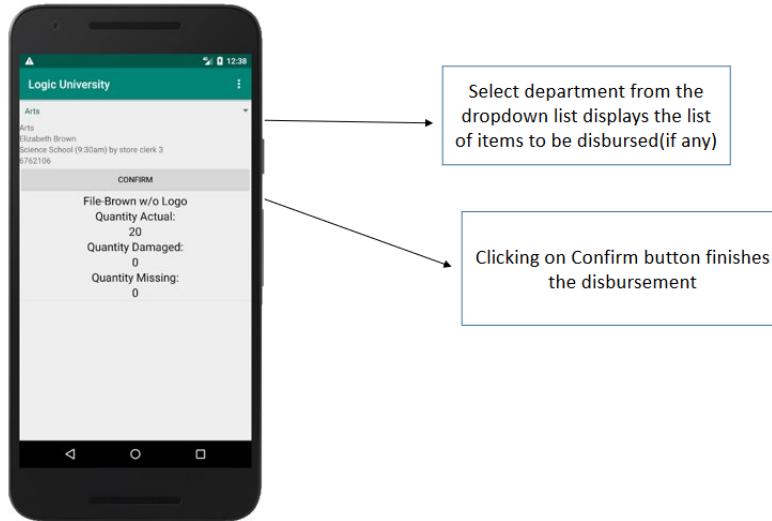


ViewAdjVoucherActivity



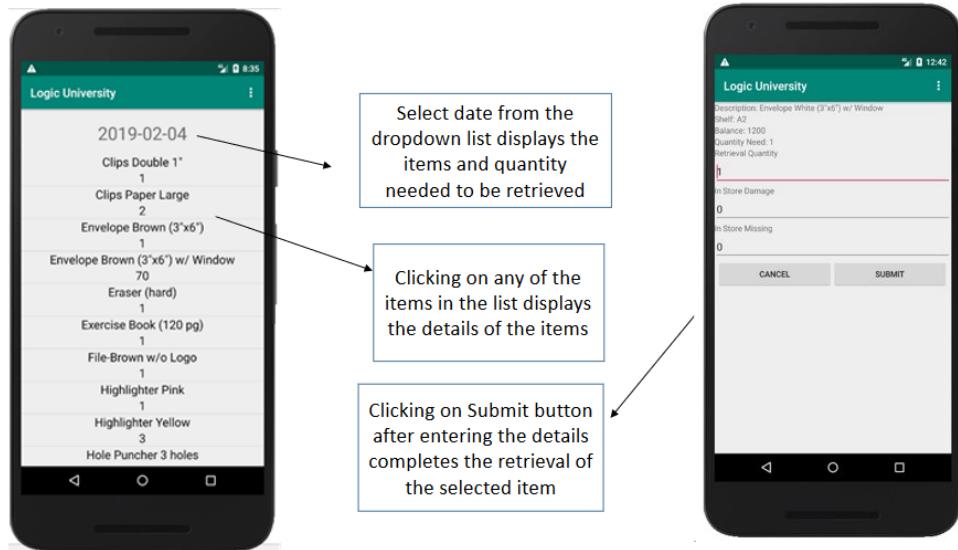
Process Disbursement List

DisbursementListActivity



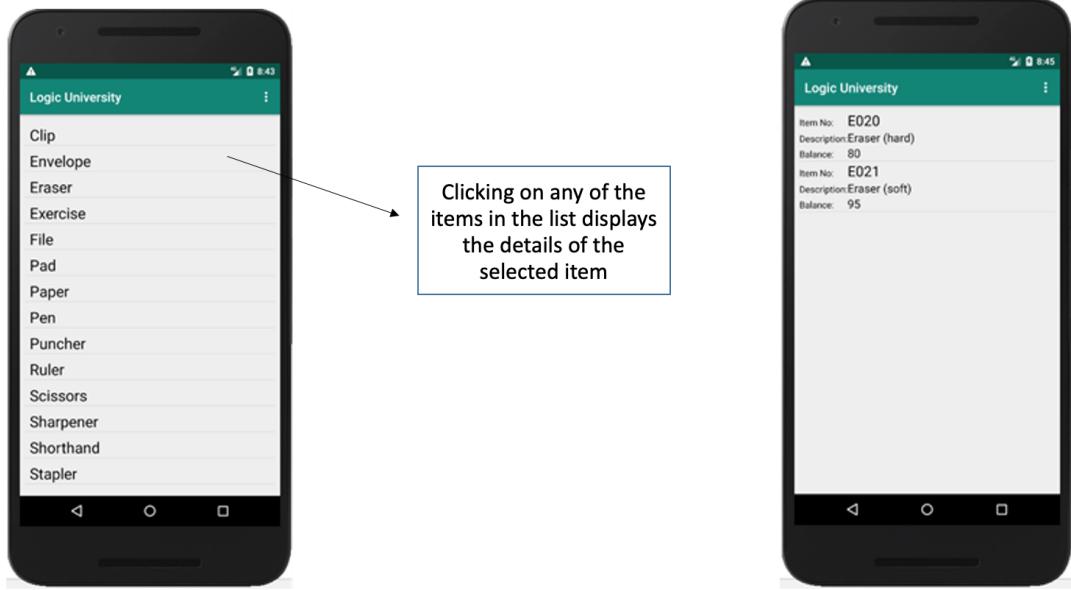
Process Retrieval List

RetrievalListActivity



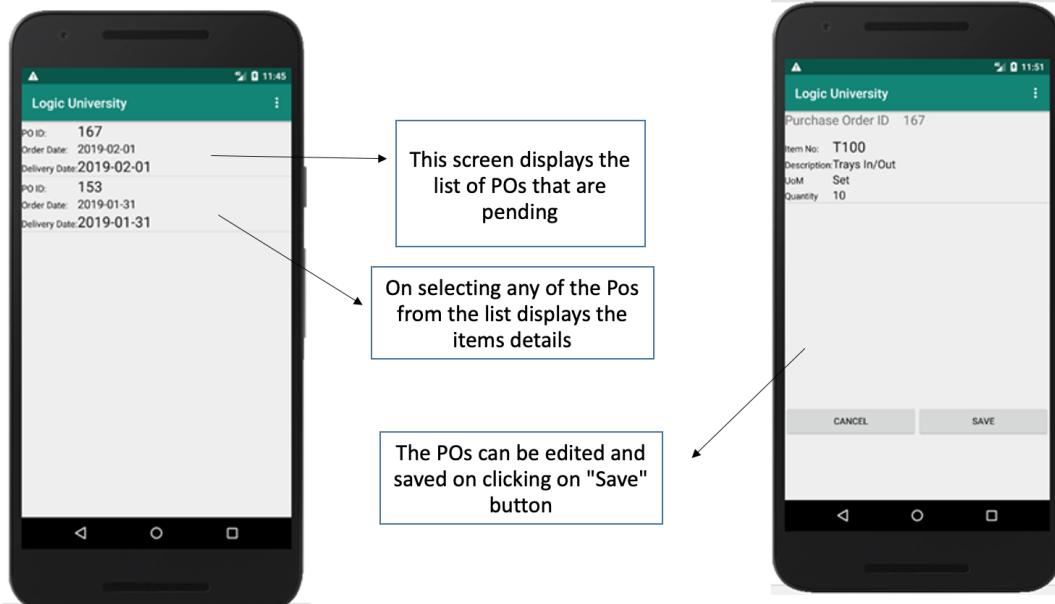
View Stock Card

ViewStockCardActivity



Close PO on delivery

ClosePOActivity



Store Supervisor/Manager – Home page

