Invenstock

A GAP Inventory System

Design Patterns Report

Group 2

Debkanya Mazumder

Sireesha Basamsetty

Onome Ovedje

Diana Carrillo

CS 442

University of Illinois at Chicago

March 2016

CONTENTS

Project Drivers

Project Constraints

Functional Requirements

Design Patterns

Table of Figures

Figure 21: Proxy Design Pattern

PROJECT DRIVERS

1. The Purpose of the Project

The clothing industry has product that changes with the seasons. The product usually comes in regular price, goes through several promotions, finally landing on the clearance rack. Despite the advancements in technology, many companies continue to use older inventory hardware and software. Often, older hardware/software can be expensive to maintain and inaccurate.

The Gap Flagship on Michigan Ave, uses a software that updates their product count every 72-hours. In those 3 days, it is possible that:

1. An item was bought.
2. An item was put on hold on one of the GAP's 3 floors.
3. It was placed randomly by a customer on the sales floor.
4. It was placed randomly by an employee anywhere within the store (including employee areas).

It is very easy to cause chaos with misplaced products, and 3 days gives the product a big head start to get lost. The "LRT guns", used by the GAP as inventory software, require nightly updates that cause many to crash. It is common for GAP managers to have to send these guns for service repair because of the age of the technology, meaning their employees are either forced to share or work without one, which can amount to a certain level of frustration. With these LRT guns, employees are able to keep track of products (within a 72-hour grace period), transfer product into or out of the store's system to another store within the company, create shipping labels, etc. Just about anything that has to do with product inventory management, is done on these LRT guns.

The goal of our project is to create an app that facilitates all of that. We aim to create an inventory app that connects to a local server for more accurate, faster counts of product. A secondary purpose is to help the company better organize and map out their product within their respective stockrooms, in efforts to keep better track of each item.

2. The Stakeholders

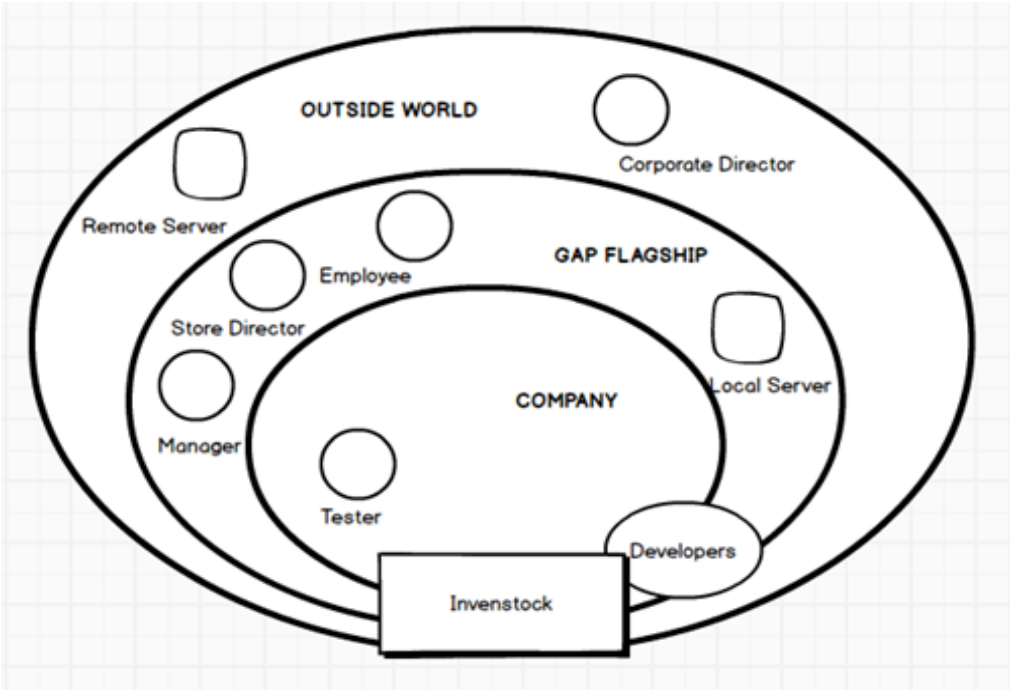

Fig1: Stakeholders

2a. The Client:

The client for Invenstock is Shannon Valvano. She is the store director of the GAP Michigan Ave Flagship store in Chicago. She wants her store to run smoothly, so she insists on better technology to facilitate her stores logistical operations in order to sell clothes at the highest selling point within its season.

2b. The Customer:

The customers for Invenstock are the employees, both managers and hourly employees, of the flagship store that are part of the logistics team. The product is to replace their current software and dated hardware used to keep track of inventory, to increase efficiency in replenishing the floor with seasonal product.

2c. Other Stakeholders

1. Stakeholder: Customer & Visual Team Managers

Knowledge the project requires: Knowing how each team's managers use their current software the same and differently. It is important to know what

information each team's leaders use to manage their employees and strategize their product marketing, placement, etc. to fulfill the requirements appropriately.

Involvement necessary: It is important to know this information when planning the product functionalities in order to plan the options available that will satisfy the needs of each team, as well as a store as a whole.

Degree of involvement: These managers will have a significant influence on the project in order to determine the project scope and boundaries. Their input will be taken into considerations at various checkpoints throughout the project to assure the needs are being satisfied.

2. Stakeholder: Customer & Visual Team Employees

Knowledge the project requires: It is important to understand how team members use their current software to perform their daily tasks, to know which functionalities work. Understanding how the employees use their current software allows the developers to understand how the functionalities work in practice, as opposed to theory.

Involvement necessary: This information is required as the product is being developed, as well as once it is released, in order to know if the project is performing the necessary functions. The employees will also serve as the testers of the product, for their knowledge will better assist developers in understanding bugs and glitches.

Degree of involvement: The employees will have significant influence once the product as it is being developed, as well as after it has been released in order to make improvements to the product.

3. Stakeholder: Developers

Knowledge the project requires: The developers will be designing and creating the product based on the requirements given by the client.

Involvement necessary: The developers are responsible for the outcome of the product based on their understanding of the client.

Degree of Involvement: The developers are heavily involved throughout the entire process, from designing to developing to testing to releasing the product.

4. Stakeholder: Corporate Directors

Knowledge the project requires: Approval or not

Involvement necessary: Once the product is released, corporate directors need to approve the product or request improvements.

Degree of involvement: Minimal, any input will be given via the store director. Main involvement is once the product is fully developed.

5. Stakeholder: Company testers

Knowledge the product requires: Once the product is developed, it is important to check for gaps in functionality via usability tests. During the testing phase, any errors found will be reported to the product developers.

Involvement necessary: It is important for the tester to understand the necessary functions the Gap employees require from the product, thus heavy involvement from the design phase to development phase is necessary, perhaps even sitting in on meetings with the client to better understand their needs.
Degree of involvement: The testers will report any bugs discovered to the developers with as much detail as possible, which will result in a more pleasurable experience for the user.

2d. The Hands-On Users of the Product

1. User Name: Logistics Team Employees

User Role: The primary users of the product are the employees from the logistics team who will likely use the product during every work shift. Using their employee credentials, employees log into the app that will help them perform their assigned daily tasks.

Attitude towards job: Most employees generally have a positive attitude towards their job and have strong work ethic. They appreciate work done quickly and efficiently and expect their technology to work as hard as they do, often admitting that their current outdated technology sometimes is cause for frustration.

2. User Name: Logistics Team Managers

User Role: The secondary users of the product are the logistics team leaders that use the product regularly to create product management plans and marketing in order to sell clothes at the highest price point possible. The

leaders will also use the product to determine trends and report to the store director and corporate.

Attitude towards job: In general, logistics team leaders have one of the harder jobs in the store because they're required to manage not only their employees but also thousands of items, which must be sorted and sized.

2e. Personas

Fig2: Persona

Monica Davis is a reserved, respectable woman from South Carolina, currently living in Hyde Park. She received a bachelor's degree in biology, and is currently looking for a job in the field; however, she is also a current employee at the GAP flagship store on Michigan Avenue as part of the logistics team.
Monica's quiet demeanor often misleads people to believe she is mean, but she is a true sweetheart with a heart of gold. She has a very strong work ethic, and appreciates efficiency and a fast-paced shift. She is a very fast learner, and even taught herself how to use several functions of the GAP's current software. Now, she is often called on to train new employees on the inner workings of the store, which she is always happy to do.
Monica is a tech guru, but she does understand how to follow logical prompts. She prefers apps with a simple look, but powerful features.

2f. Priorities Assigned to Users

Key Users: The key users for Invenstock will be the logistics team employees and leaders that will use the app during each work shift.

Secondary Users: The secondary users will be all other employees of the GAP, who will use the product with less frequency.

2g. User Participation

Client: Shannon V., the store director of the flagship GAP store on Michigan Ave. She will analyze the statistical data provided by the product in order to find trends and strategize with corporate GAP directors for future product sales, etc.

Customer: Employees of the GAP on Michigan Ave, who will be using the product to complete their assigned tasks.

2h. Maintenance Users and Service Technicians

The GAP flagship store has a contract with IBM for technical issues with their systems, who are available to assist with other IT issues. Employee credentials are also set up by the GAP's contracted IT services.

Once the local server is set up, these technicians will be overseeing its maintenance. Software related issues and bugs would be reported to the developers, who will then make the effort to fix them.

PROJECT CONSTRAINTS

3. Mandated Constraints

3.a.1. Solution Constraints

1. **Description:** The web browser must have JavaScript enabled.

Rationale: The users will need to log on to the application on the browser that requires JavaScript to support the application.

Fit Criterion: The user will be able to use the application on a web browser.

2. Description: If using IOS, the version must be at least 8.4.1 and above.

Rationale: The user will be needed to log on to the application on IOS.

Fit Criterion: The users would able to log in to the application on iPod or iPad,

3. Description: If the user is Internet Explorer, the version must be at least 9 and above

Rationale: The users will need to log on to the application on the browser that requires JavaScript to support the application.

Fit Criterion: The user will be able to use the application on a web browser.

4. Description: The application system must accommodate a maximum of 50 users logged in at once in local store and 50000 users nationally.

Rationale: The users will need open application, log in on a web browser, which requires JavaScript to support application.

Fit Criterion: Users would be able to log in to application, access the inventory system and search for items locally in stores.

5. Description: The application system must have access to the Internet
Rationale: Users will need to log in to the application, verify their log in credentials via the Internet

Fit Criterion: The user will be able to use the application on a web browser.

3.a.2 Project Constraints

1. Description: The entire application development team must use Eclipse IDE team server for collaboration.

Rationale: Setting up a uniform working environment, the development process of the application will be error free.

Fit Criterion: All developers on the team will be supervised in order to ensure compliance with set standard.

2. Description: The development team must use scrum, a form of agile development methodology.

Rationale: Using Agile methodology will ensure the development will happen in incrementally, the right timetable and prone to change as needed.

Fit Criterion: Use Agile software like ice scrum.

3. Description: The budget for the project will be $4 million.

Rationale: The budget will ensure that there is sufficient capital to meet the expenditure generated in the development process of the application

Fit Criterion: Expenses will be carefully documented to keep within the constraint of the budget

4. Description: The application should be developed within nine months

Rationale: If the developing process of the application is slow, we will stand the chances of not meeting the deadline, which is aimed at the busiest shopping session of the year.

Fit Criterion: The planned development schedule in accordance with the deadline must be maintained and adhered to.

3.b. Implementation Environment of the Current System

The application will would mainly consist of two sections which are the Front-End part of the application that deals with the user interface of the application; and the Back-End part of the application that deals mainly with the database where the inventory information is been stored.

- Front-End: HTML5/CSS3, AJAX, Java Server Pages, AngularJS
- Back-End: Apache Tomcat/MySQL
- Framework: Spring, REST angular, RESTful web-services

The application dashboard accessible by all the employees which they can check of items, ring out a customer, call other stores. Managers and Storages have access to other level of the application where they can offer discounts, price change override and so on.

3.c. Partner or Collaborative Applications

1. Description: The application shall be able to have all employees' credentials stored in the database on the server

Rationale: Since all users, which are all employees and their credentials, users can logon from multiple locations.

Fit Criterion: The users will be able to login upon entering valid credentials.

2. Description: The application shall users log on to multiple devices at the same time.

Rationale: User can log on to the desktop, iPad or iPod at the same time with valid credentials.

Fit Criterion: The users will be able to login upon entering valid credentials.

3. Description: The application shall update the inventory of the store's items every second and the nationwide inventory will be updated every 48 hours.
Rationale: Users will be able to know what item is in stock and in what quantity.

Fit Criterion: User will be able to order or sell items to customers.

4. Description: The application shall be able to location items in the stockroom of every store.

Rationale: Since application have every location of each items on each shelves in the stock room of store

Fit Criterion: User should be able to locate every item in the store.

3.d. Off-the-Shelf Software

N/A

3.e. Anticipated Workplace Environment

Since the users will be only current employees that have valid usernames and password, they will be able to log on to the application on the desktop, iPad and iPod. The Store will be where the application is being used, employees can log in successfully, start a transection of a customer, check availability of items. Only the Store manager and other managers in the store can give discount, price override, give employee discount, classify an item Damaged and adjust pricing for a defected item. The application shall be able to save every transection in the system.

3.f. Schedule Constraints

1. Description: The application shall be developed, tested and operational by November 2nd, 2016.

Rationale: The stores shall be able to replace existing retail system with the current one in order to prepared for busy thanks giving and Christmas shopping.

Fit Criterion: Users should be to login at that time.

2. Description: The Beta version of the application shall be designed and developed I four months after the requirement is been gathered.

Rationale: Once the Beta version is developed, client can look at the application and confirm if the application meets the scope of the requirement.

Fit Criterion: Clients at the point can test the application and give inputs.

3. Description: The development team shall dedicate six weeks for testing the application.

Rationale: There shall be a usability testing to check where the application is easy to use and if a user can use it with out training.

Fit Criterion: The goal is to have users with little knowledge or they are not technologically savvy use the application.

4. Description: The potential user of the application shall be provided training in order to use the application

Rationale: Employees must know how to navigate the application. Search for items, start a transection and so on

Fit Criterion: Users should be able to use that application training

3.g. Budget Constraints

1. Description: The amount of money used to develop the application shall not exceed $4,000,000

Rationale: This amount is funded by The GAP clothing company and will cover the cost of maintenance, salaries of the software developers, business analysts and testers working on the application.

Fit Criterion: Funds by the GAP clothing company do not exceed $4,000,000.

2. Description: The budget allocated for the application does not include the maintenance budget for the application.
Rationale: Maintenance budget will come from the company's profits generated from the usage of the application due to projected increase in sales.

Fit Criterion: Other updates shall be added during maintenance.

3.h. Enterprise Constraints

1. Description: The application shall make all permission and privileges to Store Managers.

Rationale: The Store Managers shall be able to view the profile of all the students and see the usage.

Fit Criterion: The store manager shall be able to view information and statistics of usage of all the employees.

2. Description: The application shall be able to automatically switch the price of the items as they transition from full priced items to discounted sale priced items nationwide.

Rationale: Users don't have to adjust the price of the items at the point of sale.

Fit Criterion: Employees can be efficient and timely when closing a transection.

3. Description: The system shall prompt for a manager in case of any price override or discounts that is to make to a customer.

Rationale: Only managers are allowed to make exceptional judgment in terms of customer discount.

Fit Criterion: Non-management employees are not allowed to give discount.

4. Description: Application shall be able to show where each employee is being zoned in the store for the duration of his or her shift.

Rationale: User should the area where they are zone in order to help customers

Fit Criterion: Users will be proactive helping customers.

5. Description: Application shall have a planogram of the store so that users can pin any part of the store that things need to be fixed or repair.

Rationale: Creates any atmosphere of efficiency and quick response to any hazardous situation.

Fit Criterion: Making the working environment server.

4. Naming Conventions and Terminology

4a. Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

Customer:
Customer is the one who will be asking for assistance in the GAP store from any of the staff working there. He will be requesting for products, which can be tracked using the application that is being developed.

Manager:
Manager is responsible for controlling the entire store such as

Customer Care Staff:
Customer Care Staff is responsible for answering any queries that are asked by customers on phone calls or in person in the store. They help with queries such as product damage, returns, offers, discounts etc.

Logistics Staff:
The Logistics Staff is responsible for helping out customers in the store by selecting the product of their choice, taking out the desired size in the clothing and directing the customer around the store.

Driver:
The Driver transports the products from the factory to the respective stores and later on from the stores to the outlet malls once they have stayed for a particular amount of time in the store.

LRT Guns:
These are devices possessed by the Logistics Staff in the stores. They are used for mapping a certain product and retrieving other product related information. They have regularly updated information stored. All the staff needs to do is scan the bar code of the product and get information or use the search bar for any other product.

Developer:
The Developer is the one who writes the code for the development of the desktop, tablet and mobile application that will be used by the Manager, Logistics and Customer Care Staff of the GAP store.

Tester:
The Tester is the one who tests the application produced by the developer for any kind of bugs or fixes. All the functionalities of the application will be tested and reported for improvement to the developer.

Front-End:
This is the code for the Graphical User Interface of the application making it interactive for the user to use the application more easily.

Back-End:
This includes the code for the functionalities of the application such as how the information is retrieved, what happens on button click, link selection.

5. Relevant Facts and Assumptions

There are certain assumptions that are made in this project as follows:

- The LRT guns used by the employees have regularly updated information; say 1hr in each store using a local server instead of using the company server that updates information every 72hrs.

- Everything in the store is placed according to its location i.e. what shelf and what position for easy mapping when a customer requests for a certain product to try on.

- Once the new stock arrives, it should remain in the store for a particular amount of time say for a specific season for seasonal clothes and maybe around 6 months for other products before it is set in the sale section then sent out to the outlet malls.

- If a customer wants to know whether a certain product is available in another GAP store in the same town, then every store must have information regarding the other stores i.e. what stock is available at a particular time.
- Shipping of goods must be done according to the demands of the store.

- A different section of items are placed in the store, which are used for testing purpose. Based on which product is sold more frequently, that particular store can order more of those products next time.

- All information stored in this application i.e. customer credit card details, personal information, product related information is confidential and not accessible to unauthorized personnel.

FUNCTIONAL REQUIREMENTS

6. The Scope of The Work

6a. The Current Situation

The GAP stores has product that changes with the seasons. But despite the advancements in technology, many companies continue to use older inventory hardware and software. Often, older hardware/software can be expensive to maintain and inaccurate.

The goal of our project is to create an app that facilitates all of that: to create an application that is efficient and accurate. We aim to create an inventory app that connects to a local server for more accurate, faster counts of product. A secondary purpose is to help the company better organize and map out their product within their respective stockrooms, in efforts to keep better track of each item.

6b. The Context of the Work

The game works for the staff of the GAP stores. Target Audience of the game is logistic, customer care staff and the manager of the GAP stores. A strong emphasis of the system will be on the accuracy and efficiency of the data. Our game aims to comprehensively track product location and details that can be accessed by the GAP staff. Managers using this game will be able to evaluate individual employee's records and the state of the store equipment like from computer systems to the bulbs in the changing rooms. The manager can then work on improving areas to make the working of the store more efficient.

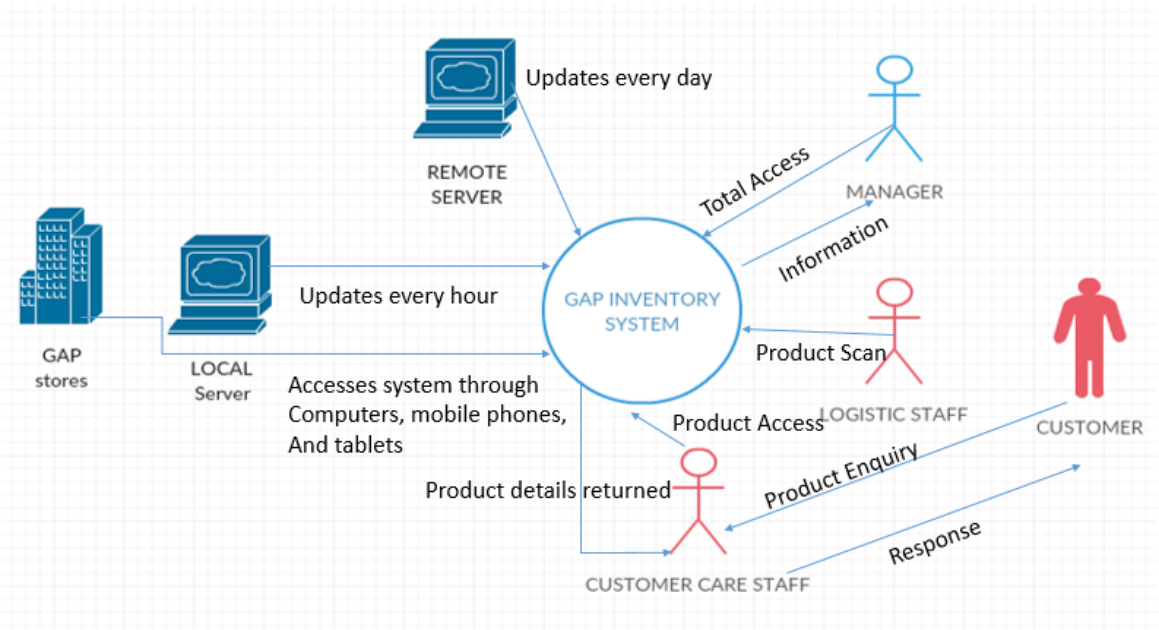

Figure 3 Context diagram for GAP inventory System

The context of the GAP inventory system is to efficiently manage the GAP inventory system. Manager has total access to all the information in the GAP stores. When the manager logs in through his id and password, the system generates and displays the entire required users to the manager. Logistic staff scans the products and adds the details into the system for inventory management. Logistic staff is also responsible for the mapping of products to its location in the storage area. Customer care staff is responsible for communicating with the customers and to resolve any doubts they have. They have access the system through their systems, mobile phones or tablets. Customer care staff checks for the product details and conveys it to the customer. GAP inventory system has two servers:

a. Local server: updates every 1-hour within the local store.
b. Remote server: updates every 24-hour with all the GAP stores in the nation.

6c. Work Partitioning

Business Event List

Event Name	Input and Output	Summary of BUC
1. Mapping of Product	Input – product name or ID Output – product location in the store	Every product in the store is mapped to it location i.e. what section and what shelf.
2. Product Separation	Input – product name or	Products are separated

and Maintenance	ID Output – Section name where product is located	based on seasonal and non-seasonal, sale clothes and testing products.
3. Product Testing	Input – product name or ID, Output- Correct result or Error	The application is tested for any bugs and fixes that will be reported to the developer.
4. Check Product Availability	Input – product name or ID Output – Available or not available	The application checks whether the product requested by the customer is available in the store or not.

6d. Business Use Case Templates

Use Case ID: 1 **Name:** Mapping of Product

Pre-conditions:

1. The product must be physically present in the store.
2. The software used by the GAP store must be installed and working correctly.

Post-conditions:

1. The products must be found in their appropriate locations in the store as per their mapping in the inventory software.
2. All expired products are supposed to be discarded.

Initiated by: Logistics Staff

Triggering Event: Start of the GAP inventory software triggers the system.

Additional Actors:

1. Customer Care Staff - check product availability in nearby stores
2. Driver - product transfer
3. Manager - employee management

Sequence of Events:

1. Start the software.
2. Enter the Login credentials like username and password.
3. Scan the bar code of the product to view details.
4. Enter details of the product if any update is required.
5. Enter the location of the product in the store i.e. what section and shelf.
6. Store the product in that particular location which makes it easier to locate later on.

Alternatives:

1. A particular product is not available or out of stock so contact nearby stores to check for availability.

Exceptions:

1. Login has failed i.e. wrong username or password entered/ user has forgotten the credentials.
2. The computer located in the store is not working.

Use Case ID: 2 **Name:** Product Separation and Maintenance

Pre-conditions:

1. The product must be physically present in the store.
2. The software used by the GAP store must be installed and working correctly.

Post-conditions:

1. The seasonal and all-year round clothes must be separated in different sections.
2. The seasonal clothes should be shipped to appropriate locations. For example, Bikinis need to be sent to California in the winter season rather than being sold in Chicago.
3. All expired products are supposed to be discarded.

Initiated by: Logistics Staff

Triggering Event: Start of the GAP inventory software triggers the system.

Additional Actors:

1. Customer Care Staff - check product availability in nearby stores
2. Driver - product transfer
3. Manager - employee management

Sequence of Events:

1. Start the software.
2. Enter the Login credentials like username and password.
3. Scan the bar code of the product to view details.
4. Check if it's a seasonal or all-year round product.
5. Check the expiry date of the seasonal clothes:

    a. If expired, then discard the product
    b. If not expired, then place in the seasonal section of the store.
6. Transfer the seasonal clothes to appropriate locations.
7. Products that have remained in the store for a long time can be either placed in the Sale section or sent to the outlet malls.

Alternatives:

1. A particular product is not available or out of stock so contact nearby stores to check for availability.

Exceptions:

1. Login has failed i.e. wrong username or password entered/ user has forgotten the credentials.
2. The computer located in the store is not working.

Use case ID: 3 **Name:** Product Testing

Pre-conditions:
1. Logistic Staff must be logged into the system.
2. Product must be available in the store

Post-conditions:
1. Expired products are segregated from the non-expired products.
2. Seasonal products separated from the all-year products.

Initiated by: Logistic Staff, Manager

Triggering Event: Logistic staff presses the "Product Testing" button on the system.

Additional Actors: Customer-Care staff.

Sequence of Events:

1. Logistic Staff presses the "Product Testing' button available on the screen.
2. System asks the user to enter the date.
3. Logistic staff enters the date into the system
4. System displays products that are currently in season, the products that are out of season and the all-year round products.
5. System also has a separate column for expired products. Any product that has expiry on the entered date is considered expired.
6. Staff can click on the expired column to check the list of expired products and see where it is located.

Alternatives:

1. A particular product is not available or out of stock so contact nearby stores to check for availability.

Exceptions:

1. Login has failed i.e. wrong username or password entered/ user has forgotten the credentials.
2. The computer located in the store is not working.

Use case ID: 4 **Name:** Check Product Availability

Pre-conditions: Staff must be logged into the system.

Post-conditions: Staff must be able to check the availability of the product in the store. If the product is unavailable, the system must show the nearest store that has the product.

Initiated by: GAP Customer Care Staff

Triggering Event: GAP Customer Care Staff enters the product name or ID he wishes to search.

Additional Actors: Logistic Staff, Store Manager.

Sequence of Events:

1. Staff enters the product name or product ID in the search bar of the desktop or mobile application.
2. The system displays the relevant product details
3. If the product does not exist in the database or the product ID is invalid then an appropriate system message is displayed to the user.
4. Staff can re-enter the product name again if he wants.
5. The system displays product availability on the screen next to the "Number of pieces available:"

Alternatives:

1. A particular product is not available or out of stock so contact nearby store to check for availability.

Exceptions:

1. Login has failed i.e. wrong username or password entered/ user has forgotten the credentials.
2. The computer located in the store is not working.

7. Business Data Model and Data Dictionary

7.a. Business Data Model

The Business Data Model is represented by the class diagram. This class diagram represents all the GAP stores situated nation-wide.
The four classes are classified as:

1. Product
2. Employee
3. GAP Store
4. Server

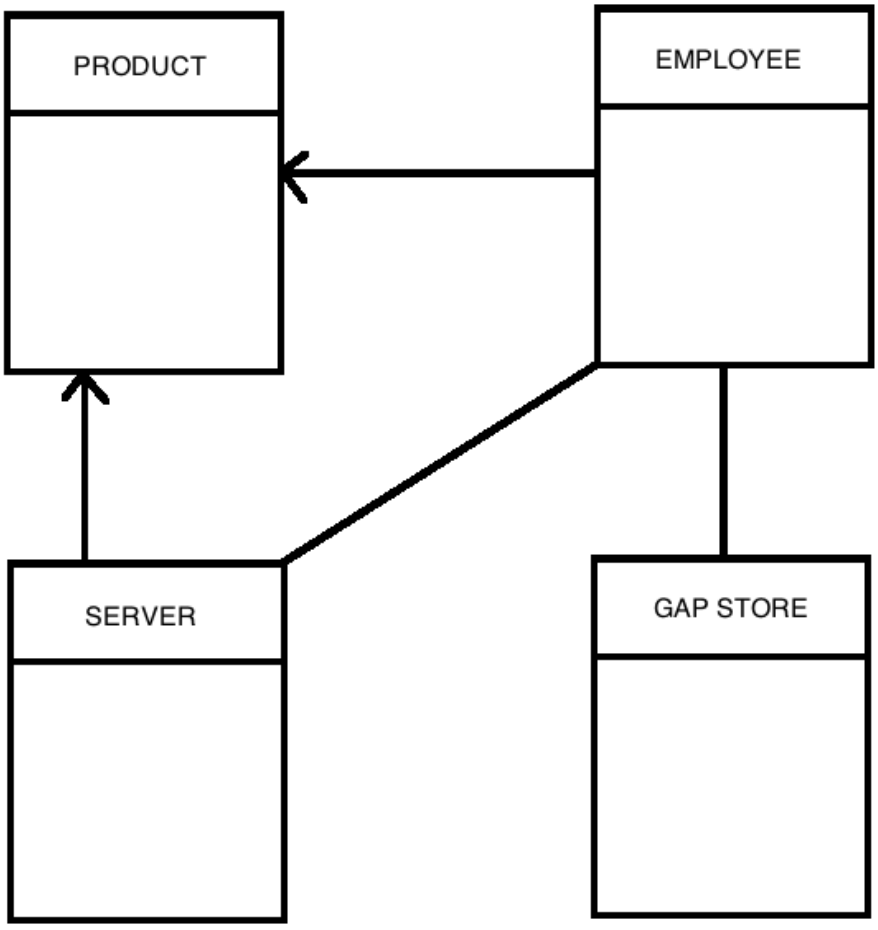

Fig 4: Business Data Model

7.b. Data Dictionary

Name	Content	Type
Customer	Customer registered	Class
Manager	Manages entire store information-customer, product, employee information	Class
Employee	Uses the application for retrieving customer and product information	Class
Product	Contains product information	Class
Server	Maintains the database	Class
Updation	Saves all the changes to the database and these changes are reflected on the dashboard.	Dataflow
Employee_id	It is the id of the employee. Every employee has a unique id.	Attribute/Element
Employee_name	Name of the employee is saved as Employee name.	Attributes
Employee_password	Password field for the employee is saved here.	Attribute/Element
Manager_password	Password field for the manager is saved here.	Attribute/Element
role	For privileged users role is set to 1(i.e. the manager).	Attribute/Element
created_on	Creation Time is saved.	Attribute/Element
created_by	Name of the account creator is saved.	Attribute/Element
last_updated	Last modified time is stored in this field.	Attribute/Element
Price	The price of each product	Attribute/Element
Quantity	The number of pieces of a certain product in the store.	Attribute/Element
Type	Type of product – seasonal or non-seasonal	Attribute/Element

8. The Scope of The Product

8a. Product Boundary

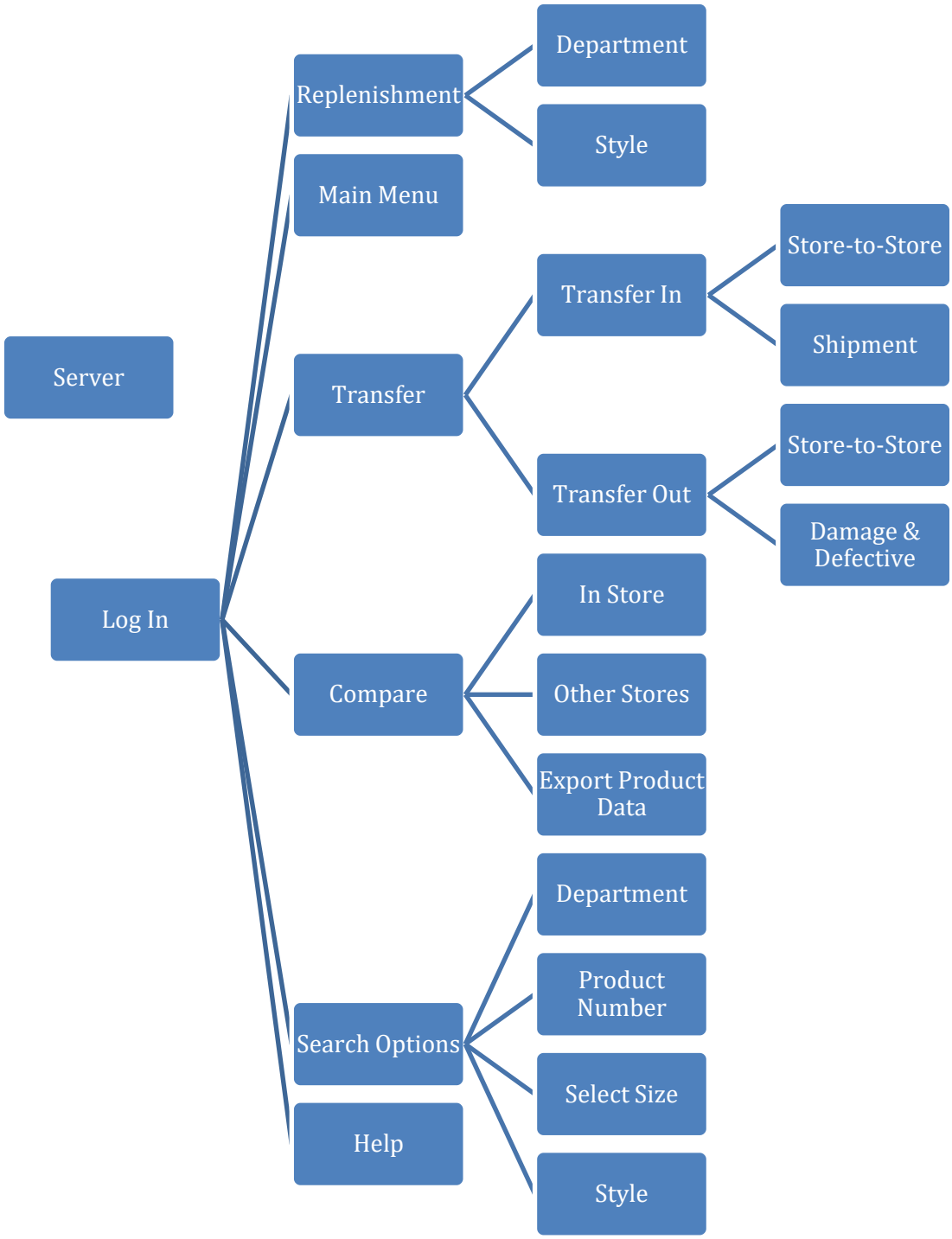

Fig 5: Product Boundary

8b. Product Use Case Table

PUC No.	PUC Name	Actor(s)	Action
1	Login	Employee (E) / Manager (M) /Store Director (SD)	User enters credentials
2	Main Menu	E / M / SD	User may has a variety of options to select
3	Replenishment Menu	E / M / SD	User may select from replenishment options
4	Select Department	E / M / SD	User may select desired department (Men's Women's, etc) from Replenishment Menu, Compare Menu, or Search Options
5	Select Style	E / M / SD	User may select desired style (outerwear, bottoms, etc) from Replenishment Menu, Compare Menu, or Search Options
6	Transfer Menu	E / M / SD	User may select from transfer options
7	Transfer In	M / SD	User may begin scanning product to append to the store's inventory list from another Gap store
8	Transfer Out	E / M / SD	User may begin scanning product to remove from the store's inventory list to send to another store.
9	Damaged & Defective	M / SD	User may begin to scan damaged/defective

			product to remove from the store's inventory list and send it back to the warehouse.
10	Compare Menu	E / M / SD	User may select from comparison options
11	In Store	E / M / SD	User may look up product counts in current store
12	Other Stores	E / M / SD	User may look up product counts in other Gap stores
13	Store-to-Store	E / M / SD	User may select store-to-store inventory transfer
14	Export Product Stats	M / SD	User may export product data
15	Search Options	E / M / SD	User may select from various search options
16	Product Number	E / M / SD	User may search using a product's unique style number
17	Help	E / M / SD	User may see the software's function map
18	Log Out	E / M / SD	User may log out and return to the main menu
19	Return	E / M / SD	User may return to previous menu
20	Shipment	M / SD	User may select to start a shipment transfer
21	Select Size	E / M / SD	User may select desired size
22	Store Number	E / M / SD	User may select store number

8c.

1.

Use Case Name	Login
Participating Actors	Employee, Manager, Store Director
Flow of Events	1. Actor opens the app, if not already opened 2. Actor is prompted for a Username and PIN 3. On successful login, the actor is redirected to the main menu, otherwise it will remain on login page, requesting credentials
Entry Condition	• App is opened • Actor must be using the store's iPod • Actor must enter valid credentials • iPod must have internet connection
Exit Condition	• Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power suddenly
Quality Requirements	• Only employees with valid credentials shall be able to log in • Employee must only be able to access their security clearance level-allowed menus • Information on the iPods must be updated within specified time frame

2.

Use Case Name	Main Menu
Participating Actors	Employee, Manager, Store Director
Flow of Events	1. Actor opens app, if not already open 2. The actor logins to the app 3. Actor selects desired option from the following: a. Replenishment b. Transfer c. Compare d. Search Options e. Help f. Log Out 4. Actor is redirected to new menu

Entry Condition	• App is opened • Actor must be using store's iPod • iPod must have internet connection
Exit Condition	• Actor selects "Log Out" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Employees can only access their security clearance level-allowed menus • Product data must be updated within specified time frame.

3.

Use Case Name	Replenishment menu
Participating Actors	Employees, Managers, Store Directors
Flow of Events	1. Actor opens the app, if not already opened 2. Actor enters valid credentials 3. Actor selects "Replenishment" from options in Main Menu 4. Actor is redirected to Replenishment menu which consists of two drop down menus a. Department b. Style
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

4.

Use Case Name	Select Department
Participating Actors	Employee, Manager, Store Director
Flow of Events	1. Actor opens the app, if not already opened

	2. Actor enter valid credentials 3. Actor selects “Replenishment”, “Compare”, or “Search Options” menus 4. Actor must select from the drop down menu: a. Women b. Men c. Kids d. Baby e. Body f. Fit g. Third-Party
Entry Condition	• App is opened • Actor must be using the store’s iPod • iPod must have internet connection • Actor must be logged in • Actor must have selected menu from Event 3
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can view this information • Product data must be updated within specified time frame • Actor must enter valid ‘Department’ and ‘Style’ combination to continue

5.

Use Case Name	Select Style
Participating Actors	Employee, Manager, Store Director
Flow of Events	1. Actor opens the app, if not already opened 2. Actor enters valid credentials 3. Actor selects “Replenishment”, “Compare”, or “Search Options” menus 4. Actor makes selection from “Department” drop down menu 5. Actor must select from the drop down menu (note that not all style options are available to all departments): a. Tops b. Bottoms c. Outerwear d. Dresses

	e. Sweaters f. Shoes g. Accessories h. Seasonal
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in • Actor must have selected menu from Event 3 • Actor must have made selection in Event 4
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can view this information • Product data must be updated within specified time frame • Actor must enter valid 'Department' and 'Style' combination to continue

6.

Use Case Name	Transfer Menu
Participating Actors	Employees, Manager, Store Director
Flow of Events	1. Actor opens the app, if not already opened 2. Actor enters valid credentials 3. Actor selects "Transfer" from options in Main Menu 4. Actor is redirected to Transfer menu: a. Transfer In b. Transfer Out 5. Actor makes a selection to continue
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified

	time frame

7.

Use Case Name	Transfer In
Participating Actors	Managers, Store Director
Flow of Events	1. Actor opens app, if not already opened 2. Actor enters valid credentials 3. Actor selects “Transfer” from Main Menu 4. Actor is redirected to “Transfer In” menu: a. Store-to-Store b. Shipment 5. Actor makes a selection to continue
Entry Condition	• App is opened • Actor has security clearance of manager or higher • Actor must be using the store’s iPod • iPod must have internet connection • Actor must be logged in
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can view this information • Product data must be updated within specified time frame • Data must be erased and restarted on loss of power (to avoid confusion)

8.

Use Case Name	Transfer Out
Participating Actors	Employees, Managers, Store Director
Flow of Events	1. Actor opens app, if not already opened 2. Actor enters valid credentials 3. Actor selects “Transfer” from Main Menu 4. Actor selects “Transfer Out” and is given the following options: a. Store-to-Store b. Damaged and Defective 5. Actor makes selection to continue
Entry Condition	• App is opened • Actor must be using the store’s iPod

	• iPod must have internet connection • Actor must be logged in
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can view this information • Product data must be updated within specified time frame • Data must be erased and restarted on loss of power (to avoid confusion)

9.

Use Case Name	Damaged & Defective
Participating Actors	Managers, Store Director
Flow of Events	1. Actor opens app, if not already opened 2. Actor enters valid credentials 3. Actor selects “Transfer” from Main Menu 4. Actor selects “Transfer Out” 5. Actor selects “Damaged and Defective” a. If Actor does not have manager credentials, Actor will be notified its required 6. Actor can scan damaged/defective items to transfer out
Entry Condition	• App is opened • Actor must be using the store’s iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees with manager credentials can view this information • Product data must be updated within specified time frame • Data must be erased and restarted on loss of power (to avoid confusion)

10.

Use Case Name	Compare menu
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

11.

Use Case Name	In Store
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

12.

Use Case Name	Other Stores
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod

	• iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

13.

Use Case Name	Store-to-Store
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store’s iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

14.

Use Case Name	Export Product Stats
Participating Actors	Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store’s iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod

	• iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

15.

Use Case Name	Search Options
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

16.

Use Case Name	Product Number
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

17.

Use Case Name	Help
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Help menu should not require additional help menu • Search keywords

18.

Use Case Name	Log Out
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

19.

Use Case Name	Return
Participating Actors	Employees, Managers, Store Director
Flow of Events	
Entry Condition	• App is opened

	• Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees can access this menu • Product data must be updated within specified time frame

20.

Use Case Name	Shipment
Participating Actors	Managers, Store Director
Flow of Events	
Entry Condition	• App is opened • Actor must be using the store's iPod • iPod must have internet connection • Actor must be logged in with manager credentials
Exit Condition	• Actor selects "Return" • Actor presses 'Home' button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Only employees with manager credentials can view this information • Product data must be updated within specified time frame • Data must be erased and restarted on loss of power (to avoid confusion)

21.

Use Case Name	Select Size
Participating Actors	Employees, Managers, Store Director
Flow of Events	1. Actor opens the app, if not already opened 2. Actor enters valid credentials 3. Actor selects "Compare" or "Search Options" from options in Main Menu 4. Actor selects "Compare" a. Actor must select "In Store" or "Other

	Stores” b. Actor is prompted to enter 4 digit size code 5. Actor selects “Search Options” Actor is prompted to enter 4-digit size code
Entry Condition	• App is opened • Actor must be using the store’s iPod • iPod must have internet connection
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Product data must be updated within specified time frame

22.

Use Case Name	Store Number
Participating Actors	Employees, Managers, Store Director
Flow of Events	1. Actor opens the app, if not already opened 2. Actor enters valid credentials 3. Actor selects either “Transfer” or “Compare” from options in Main Menu 4. Actor Selects “Transfer” a. Actor selects “Transfer In” or “Transfer Out” b. Actor selects “Store-to-Store” c. Actor is prompted for 4-digit store number 5. Actor Selects “Compare” a. Actor selects “Other Stores” b. Actor is prompted for 4-digit store number
Entry Condition	• App is opened • Actor must be using the store’s iPod • iPod must have internet connection
Exit Condition	• Actor selects “Return” • Actor presses ‘Home’ button on iPod • Actor turns off iPod • iPod dies or loses power
Quality Requirements	• Product data must be updated within specified time frame

	• Product is to have fail-safe that will save data in event of crash or power loss

DESIGN PATTERNS

28. Design Goals

The design goals represent the desired qualities of the GAP Invenstock application and provide a consistent set of criteria that must be considered when making design decisions. Our main purpose is to develop robust, maintainable, well-designed and reusable software with Object-Oriented analysis and design. We are determined to define and visualize each and every perspective of the system explicitly in order to completely materialize our Object-Oriented approach. Next, we also pay attention to how to diminish the influence and impact of alterations, how to keep the elements of our design understandable, manageable, and focused and who is when behavior differs by type.

The design goals identified in details are as follows:

- **Adaptability:** Java is one of the few programming languages, which provides cross platform portability. This attribute of Java enables our system to work in all JRE installed platforms; therefore, user will not have to worry about the operating system requirements. In order to fulfill this adaptability feature, we preferred to program the application in Java by sacrificing some of the performance advantages of other programming languages such as C or C++.

- **Efficiency:** The system is going to be responsive and able to run with high performance. The application will be retrieving information efficiently – accurate data and fast response as a local server is being used. This is the most important design goal because performance of the application has a crucial role for users' excitement. In order to reach the optimum application performance, rather than trying to minimize the memory usage, we allocated memory for each individual objects so that they will be responsible for their own tasks.

- **Reliability:** System will be bug-free and consistent in the boundary conditions. The system should not crash with unexpected inputs. To achieve this goal, the testing procedures will continue simultaneously with each stage of the development. Besides, boundary conditions will be evaluated very carefully not to miss any unconsidered situation, which may crash the system.

- **Usability:** Easiness in the usage is an important design goal in terms of user's comfort. It makes the application more friendly and attractive.

Therefore, the system will be designed such that user can easily interact with the system without any prior knowledge.

- **Extensibility:** Object oriented architecture of the application enables system customizations without causing any bugs during modifications. The application should be able to run with several staff concurrently using it a point of time. Therefore, this design architecture minimizes the possibility to cause malfunctioning in other classes.

29. Design Patterns applicable to Invenstock

29a. Overview of the design patterns applicable to Invenstock

Architecture of the Application

The application uses the MVC (Model-View-Controller) architecture. Each layer has set of design patterns, which has a different intent.

View: Composite, Memento, Decorator, Iterator, Abstract Factory, Command, State
Model: Strategy, Adapter, Singleton
Controller: Observer, Proxy

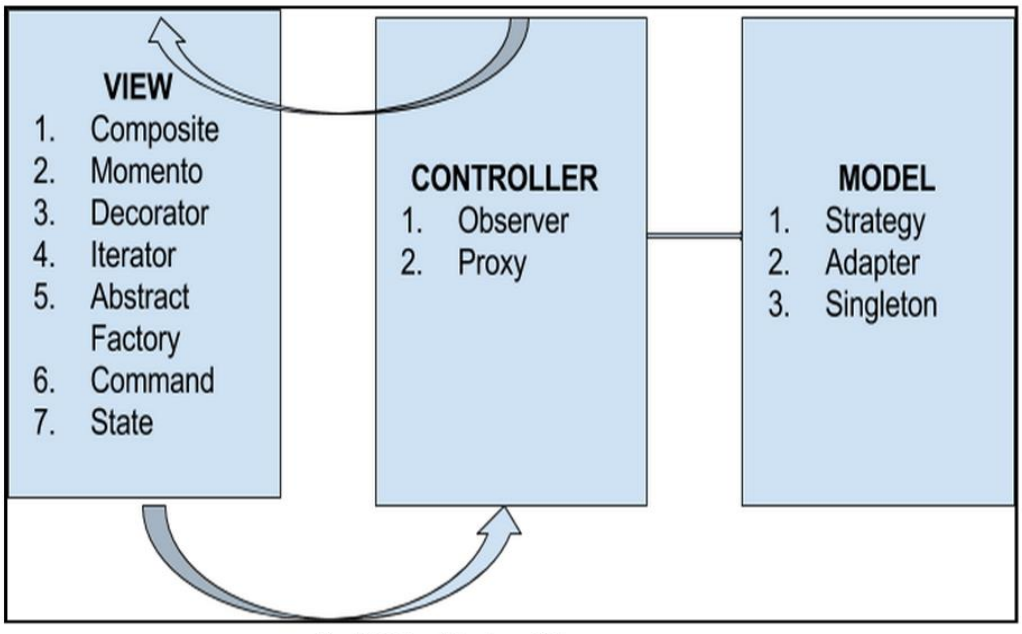

Fig 6: MVC Architecture with Design Patterns

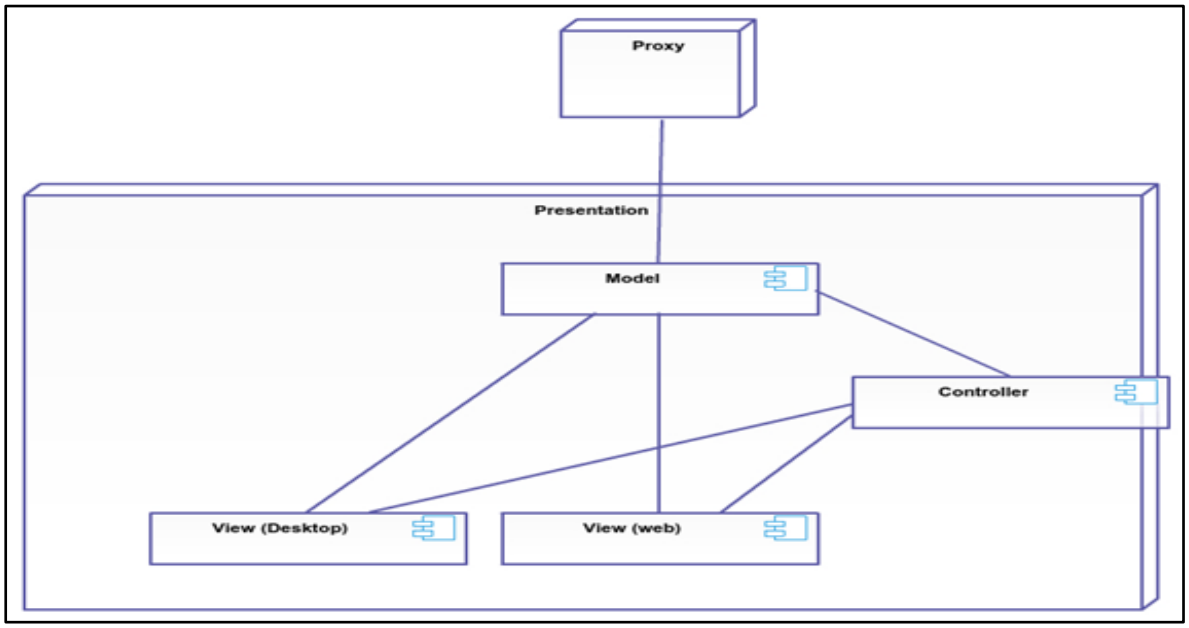

Fig 7: Higher Level Representation of the design patterns in MVC architecture

29b. Creational Patterns

I. Abstract Factory Design Pattern

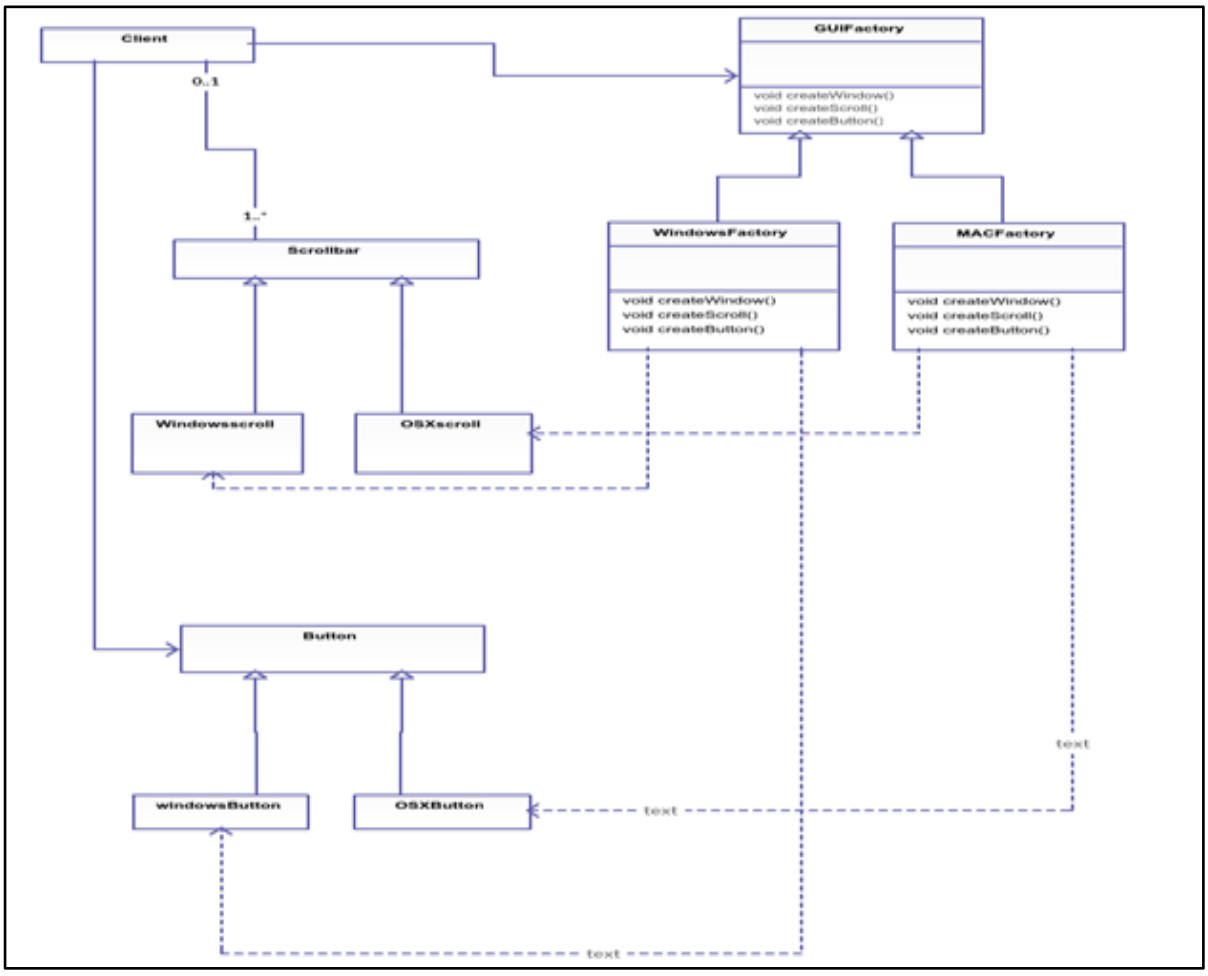

Fig 8: Abstract Factory Design

Description:

The purpose of using the abstract factory design pattern here is to create multiple different widgets, such as a new window, a scrollbar and a button. Using the abstract factory design pattern will help simplify the process. The GUI Factory class, an abstract class, declares an interface for creating each kind of widget. Then, there is also an abstract class for each kind of widget, such as Windows or Mac OS X.

The clients access the GUI Factory interface to obtain a window, scrollbar or a button for each of the abstract widget class. The clients do not directly call the concrete classes at any point; in fact they are not even aware of them.

II. Builder Design Pattern(CHANGE)

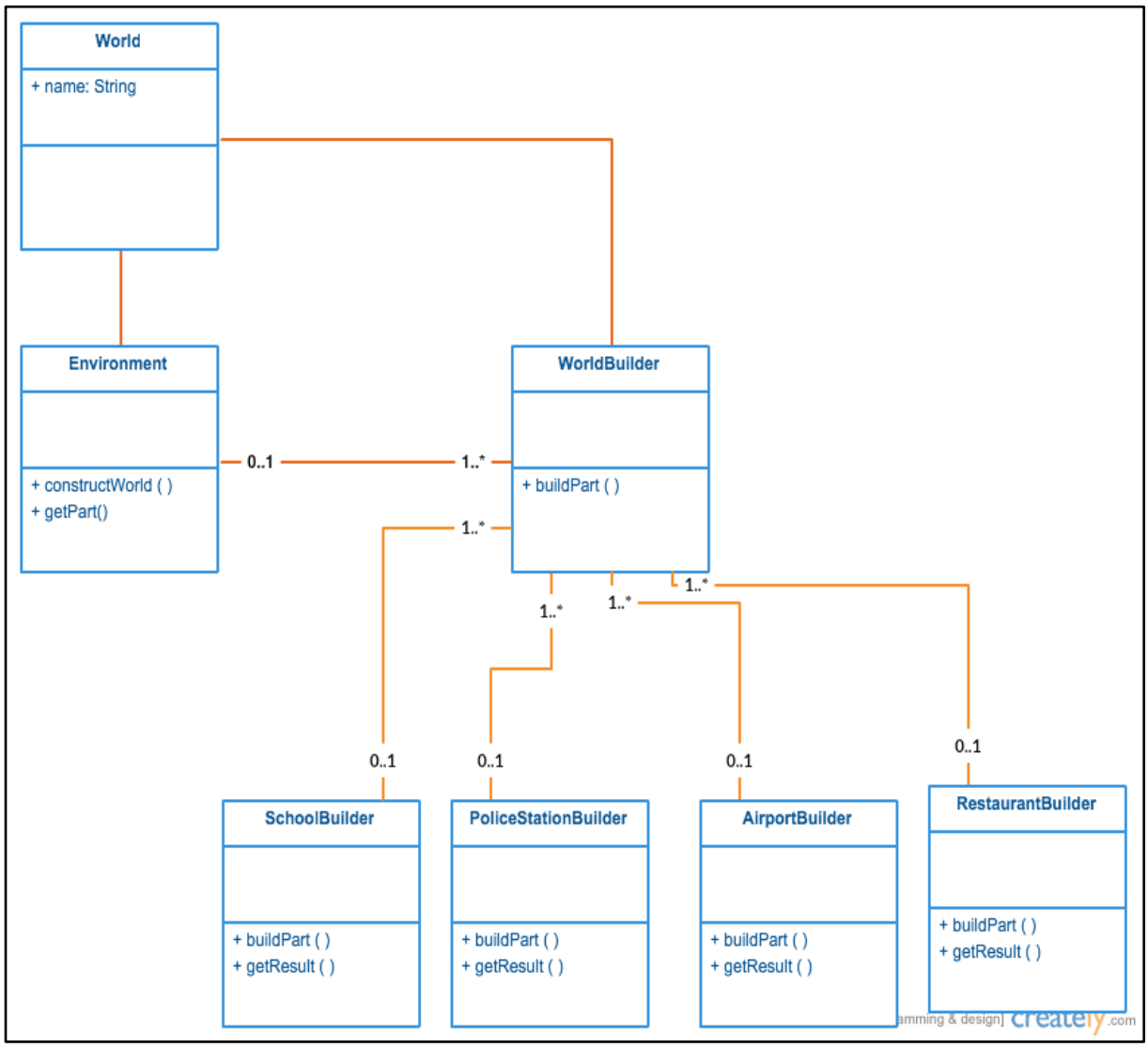

Fig 10: Builder Design Pattern

Description:

Server divided into local and remote server. Individual functions for each of them. Local- 1hr update
Remote-24hr update

The purpose of using the Builder pattern is for incremental construction of a large complex application. Using this pattern, it will greatly simplify the creation of the complex parts of the application like , and so on. The WorldBuilder class specifies an abstract interface for creating parts of the product objects of the School, PoliceStation, Airport, and Restaurant.

The SchoolBuilder, PoliceStationBuilder, AirportBuilder, and RestaurantBuilder, constructs and puts together parts of our product by implementing the WorldBuilder interface. This will define and keep track of the representation it creates and provides an interface for saving the products.

The Environment class constructs the complex objects using the WorldBuilder interface. The products, i,e., School, PoliceStation, Airport, and Restaurants represents the complex objects that are being built.

All of this is tied together with the World class. The world class is from where we know that these products must conform to the current world. For example, if the world is Mexico, then all of our products will have the setting of Mexico and likewise with other worlds.

III. Singleton Design Pattern

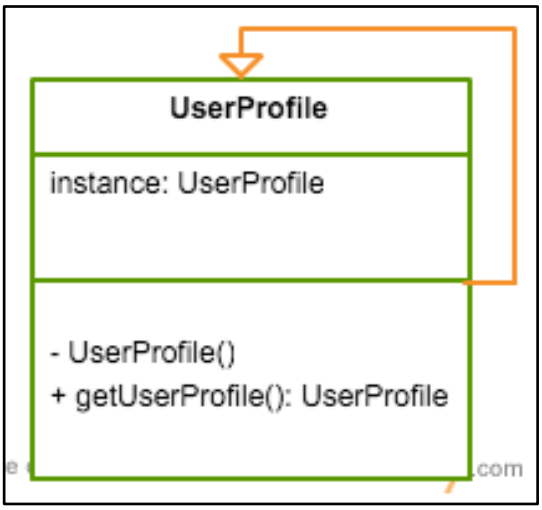

Fig 11: Singleton Design Pattern

Description:

The reason for the use of the Singleton pattern is that we can have only one instance of the User's profile. This way, no matter from where the User is accessing his/her account, they can get a handle to the same saved instance of their profile from any device. Singleton pattern allows us to do this.

1 product

29c. Behavioral Patterns

1. chain of responsibility:

sequence diagram

1. login- check user – show functions depending on which user - logout

II. Iterator Design Pattern

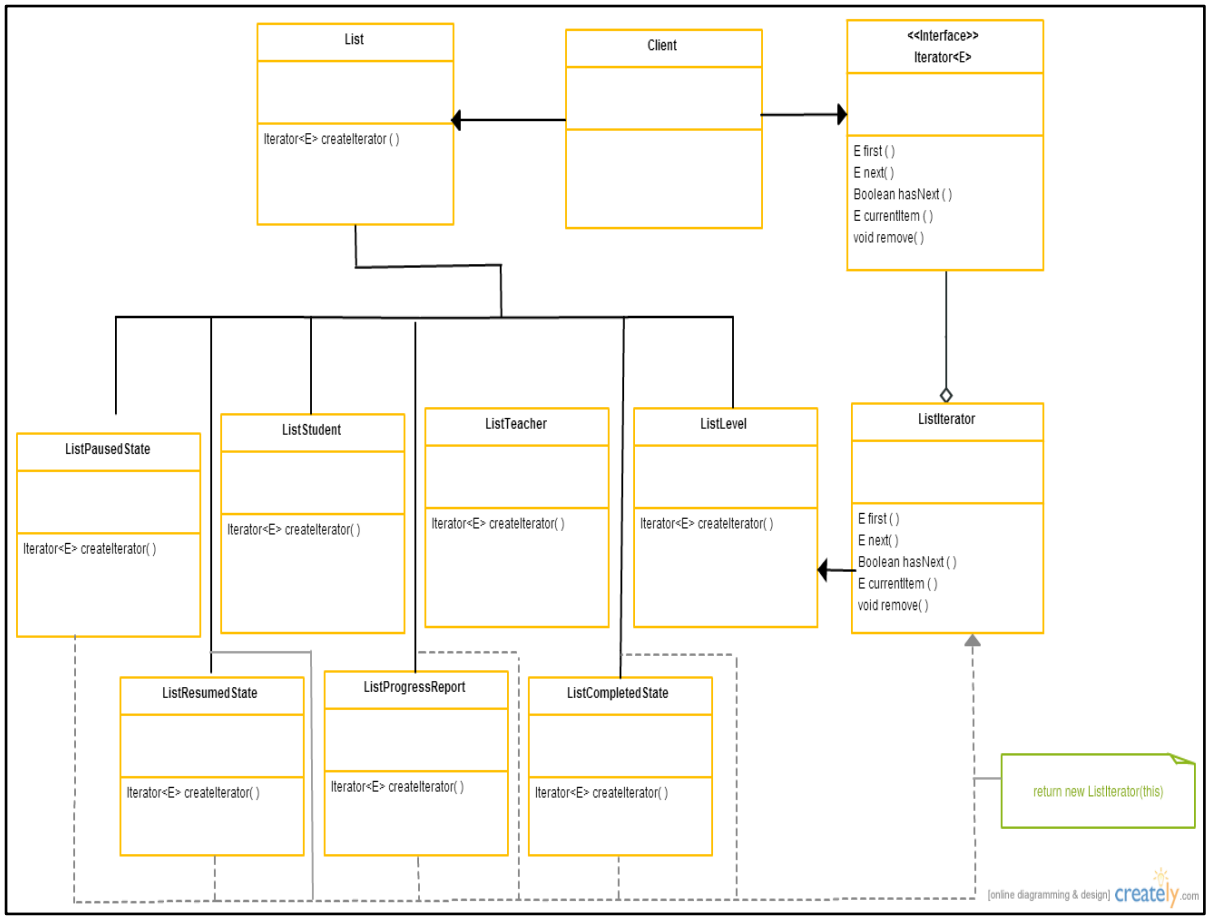

Fig 13: Iterator Design Pattern

Description:list down the list of products

The intent to use this design pattern is to navigate through the elements of different lists- Student, Teacher, Completed State, Paused State, Level, Resumed State, Progress Report, In-Progress State. So the iterator method is called from these lists and allows the user to access the different collections sequentially.Method call:

```
Iterator<E> createIterator()  {
   return new listIterator(this);
}
```

Different lists can access the methods of the iterator, thus making the access clean and simple. The ListIterator has the following methods:

first(): access the first element of list

next(): access the next element from the current position

hasNext(): checks if the list has element or not and returns boolean value

remove(): removes the list

currentItem(): Return the current element.

III. Memento Design Pattern

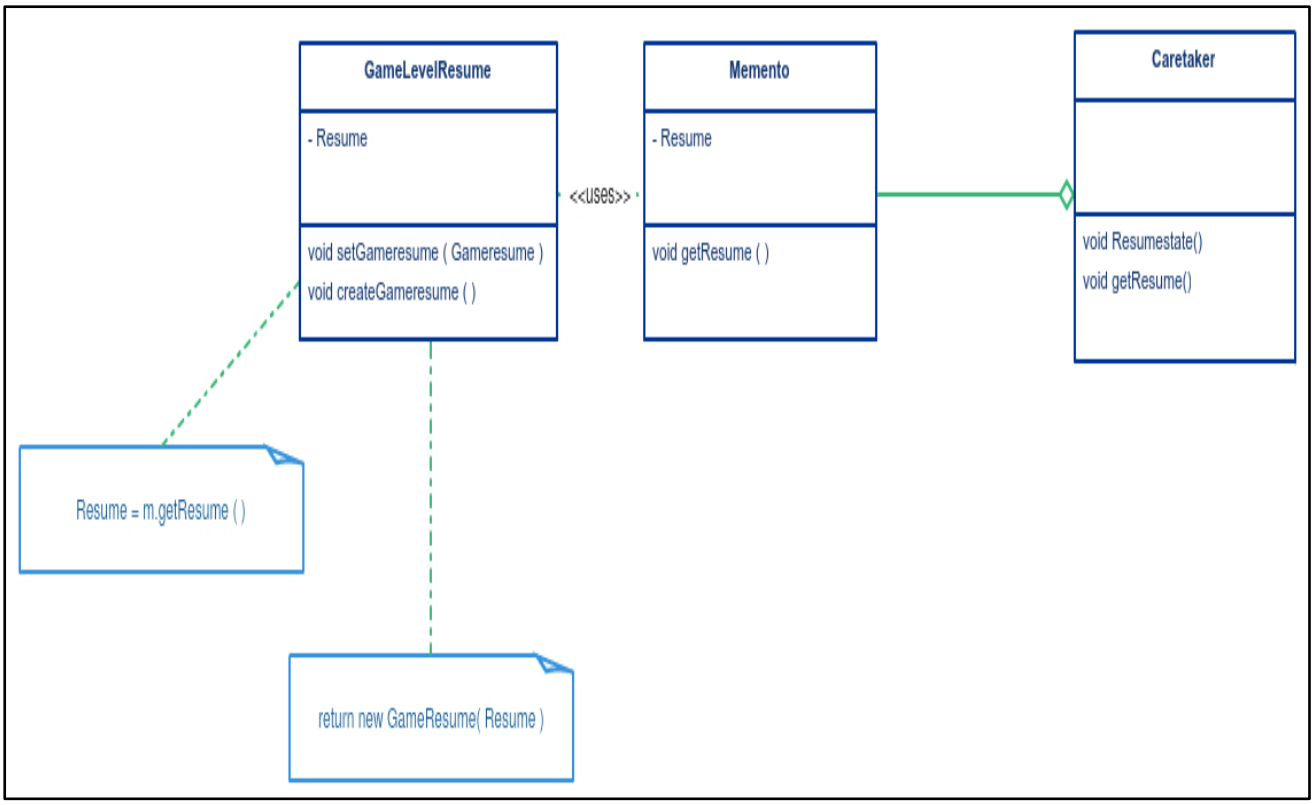

Fig 14: Memento Design Pattern

Description:

During shutdown, save the previous state.

The purpose of using the Memento design pattern is to address our goal for students to replay a level without old progress being overwritten, since the Teacher needs to be appraised of Student's attempts as well as progress.

Memento Pattern is used to save the state of a level for each attempt a Student makes to complete it. In our Memento Design pattern Resume is the state. Memento class captures the internal state of GameLevelResume.

IV. Observer Design Pattern

The main purpose of observer design pattern is to notify the users of the change. In the game, as the student starts playing the game, scores are updates and so is the progress report. To implement these two tasks, Observer design pattern is used.

1. updation for user should be notified to the system
2. updation regarding the shipment

a) Progress Report Observer

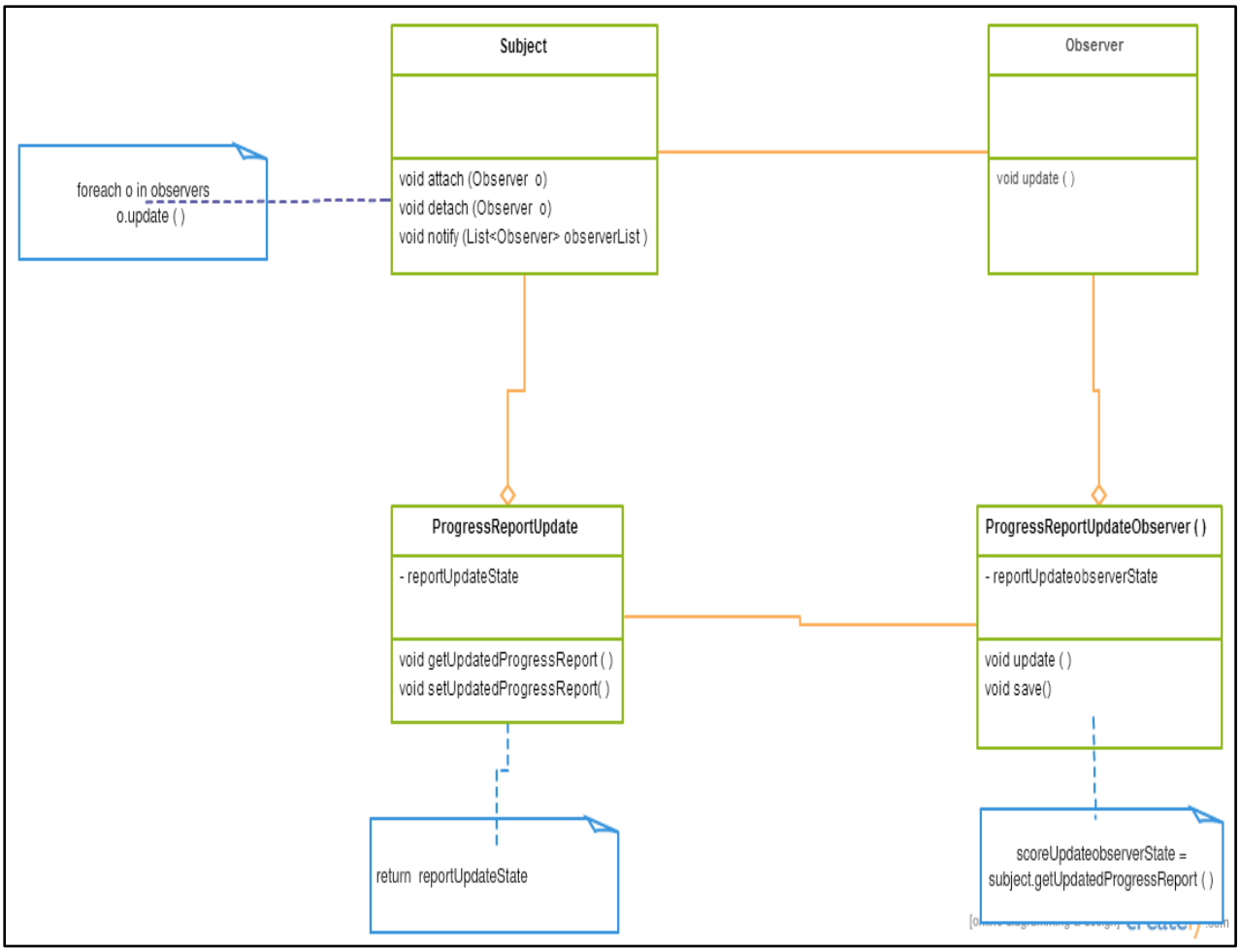

Fig 15: Report Progress Observer Design Pattern

Description:

When the player completes the level of the game, the progress report of the student is updated. ProgessReportUpdate's notify() method is called to inform all the users subscribed to ProgressReportUpdate that a new event has occurred. The progress report of the student is updated in the database as well as the dashboard of the student and the teacher.

b) Score Update Observer

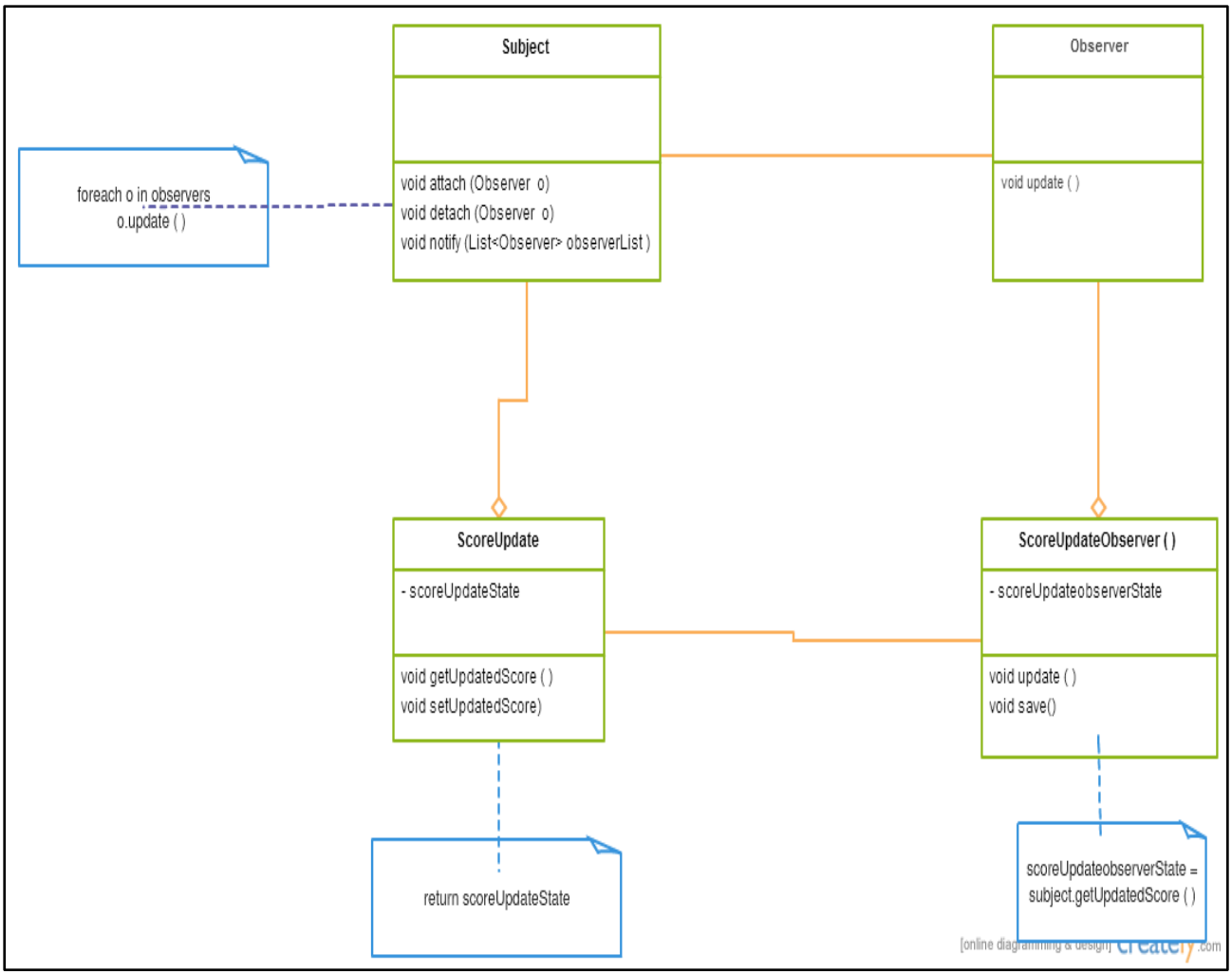

Fig 16: Score Update Observer Design Pattern

Description:

When the player completes the level of the game, the score of the student is updated. ScoreUpdate's notify() method is called to inform all the users subscribed to ScoreUpdate that a new event has occurred.The scores are then updated in the database and correct scores are displayed on the screen.

V. State Design Pattern

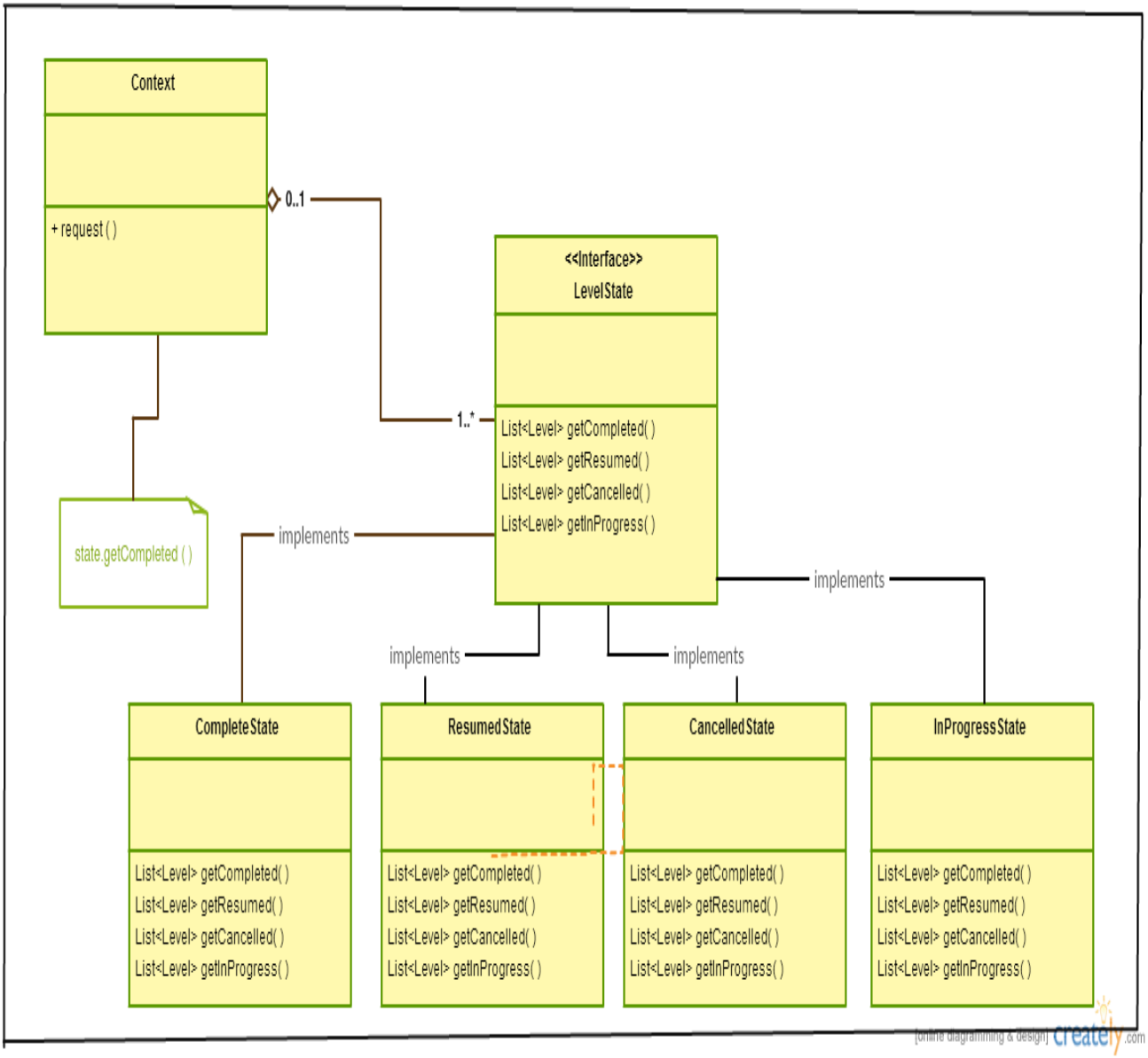

Fig 17: State Design Pattern

Description:

States of the items- full_price, sales_prices, outlet_sales_price

States of the employee – currently_working, on-break, currently_on_lunch

The State pattern will be used to track the current state of the level. The four states will be Completed, Resumed, Cancelled, and Incomplete(In Progress).The LevelSate will appear to change its state to CompletedState, ResumedState, CancelledState, InProgressState.

LevelState provides interface to four classes: CompletedState, ResumedState, CancelledState, InProgressState.

VI. Strategy Design Pattern

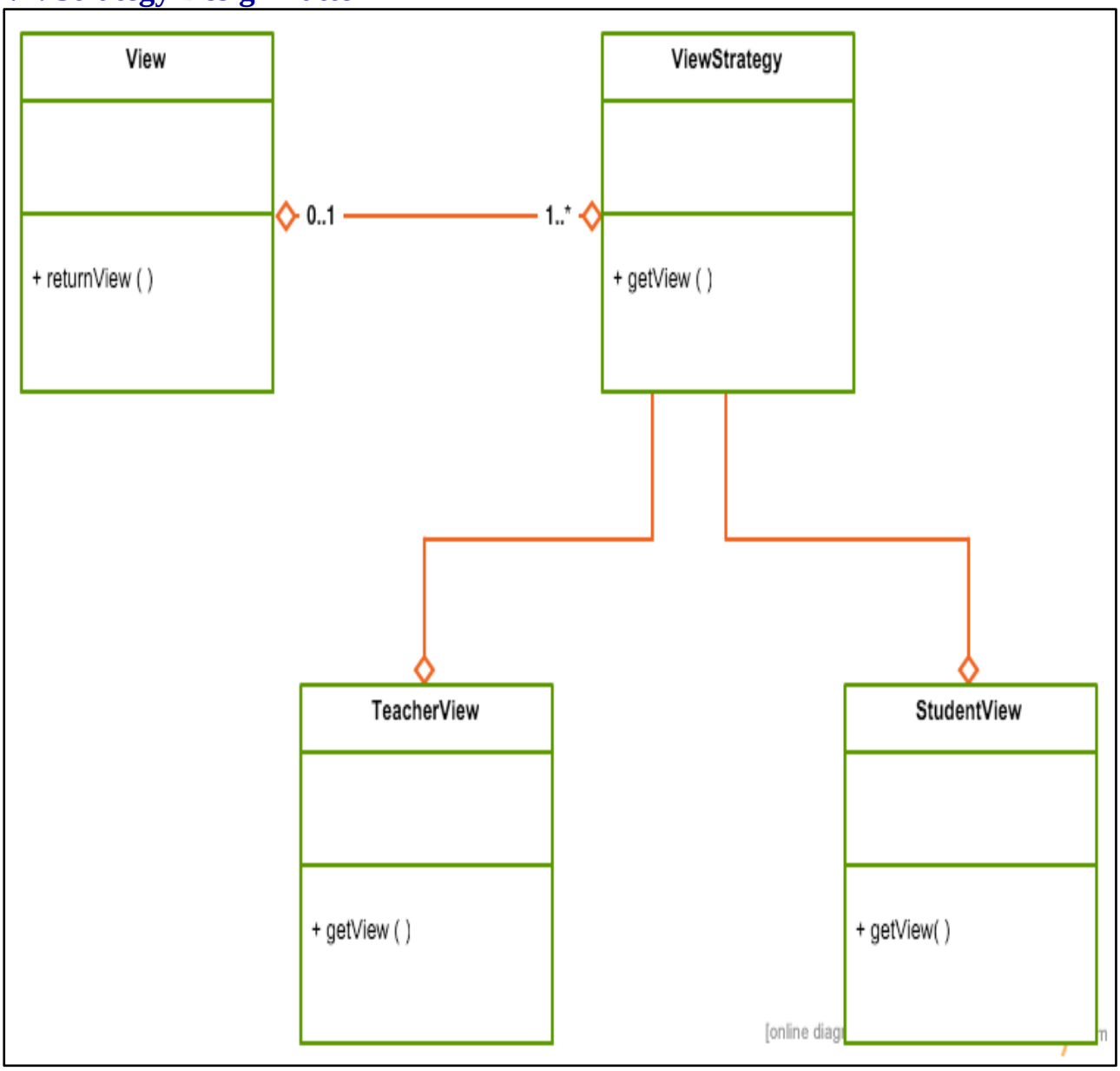

Fig 18: Strategy Design Pattern

Description:

The purpose of the Strategy pattern is to choose between which views to show to the User, depending on the type of account the User holds. If the user logging in has a Manager's account, then the view will be specific to the Manager, and vice-versa for the Staff's account. This way we will have the same name of the method for showing two different screens.

29d. Structural Patterns

I. Composite Design Pattern

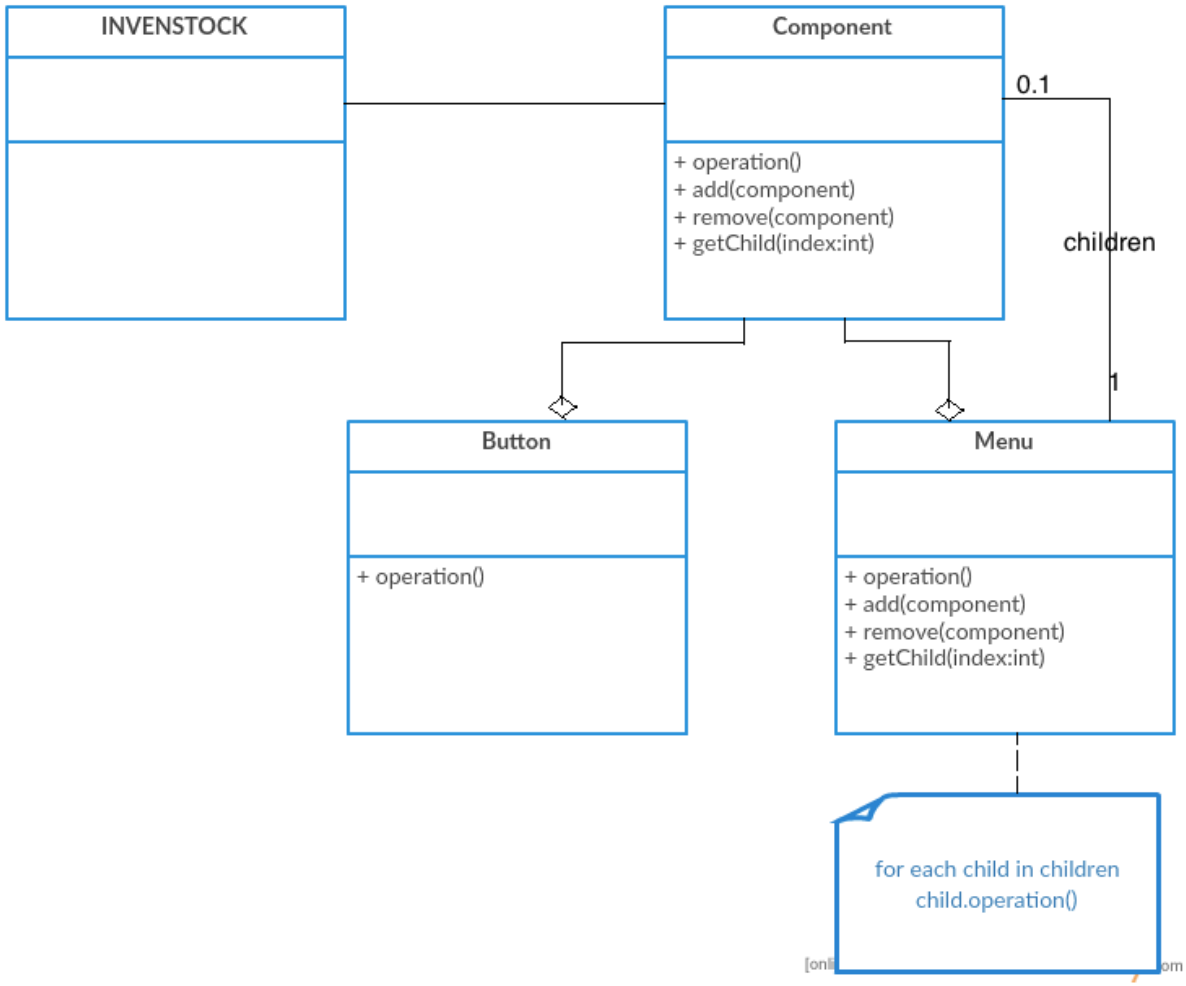

Fig 19: Composite Design Pattern

Description:

Search bar, scanner, change the operations

Composite pattern will be used to create application buttons and menus. With this pattern, it will be very simple to create such a structure. The Component class is an abstraction for Button and Menu. It defines an interface that will be implemented by the objects in the composition, i.e., Button and Menu.

The Button will be the object that will not have any children. It will implement services that the Component interface describes. The Menu holds the child

components (Buttons) in addition to implementing methods that our Component interface defines. It does this by delegating responsibility to the child.

II. Decorator Design Pattern(CHANGE)

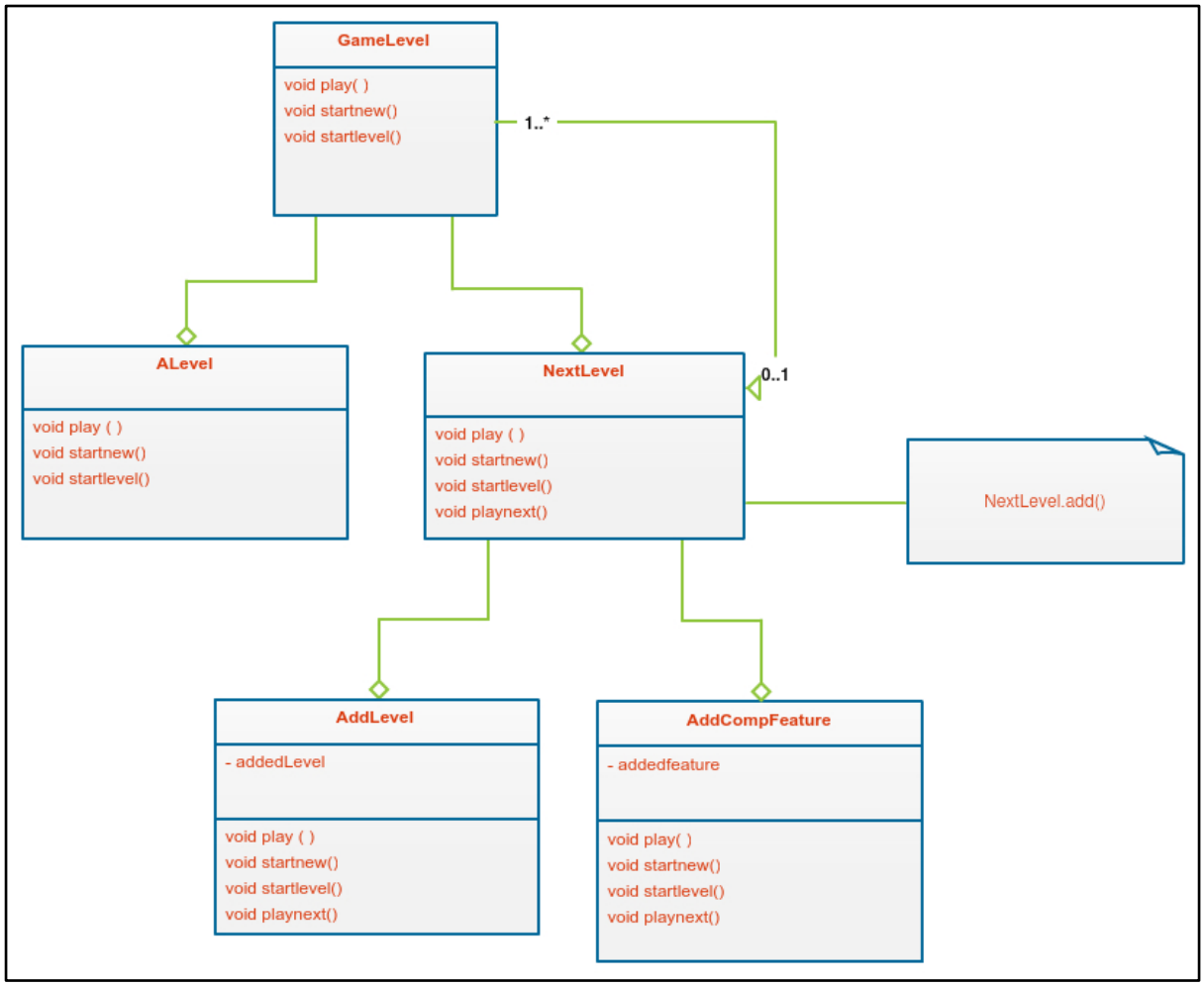

Fig 20: Decorator Design Pattern

Description:

Seasonal and non-seasonal products

~~The purpose of using Decorator design pattern is to design the look and feel of the application, how each game level is designed. Decorators help in attaching additional responsibilities to an object dynamically. In our Game, we intended to add next level and increase in complexity dynamically for which decorators have proven excellent records.~~

~~In our Decorator pattern, NextLevel class is the decorator and AddLevel, AddCompFeature are concrete decorators.~~

III. Proxy Design Pattern

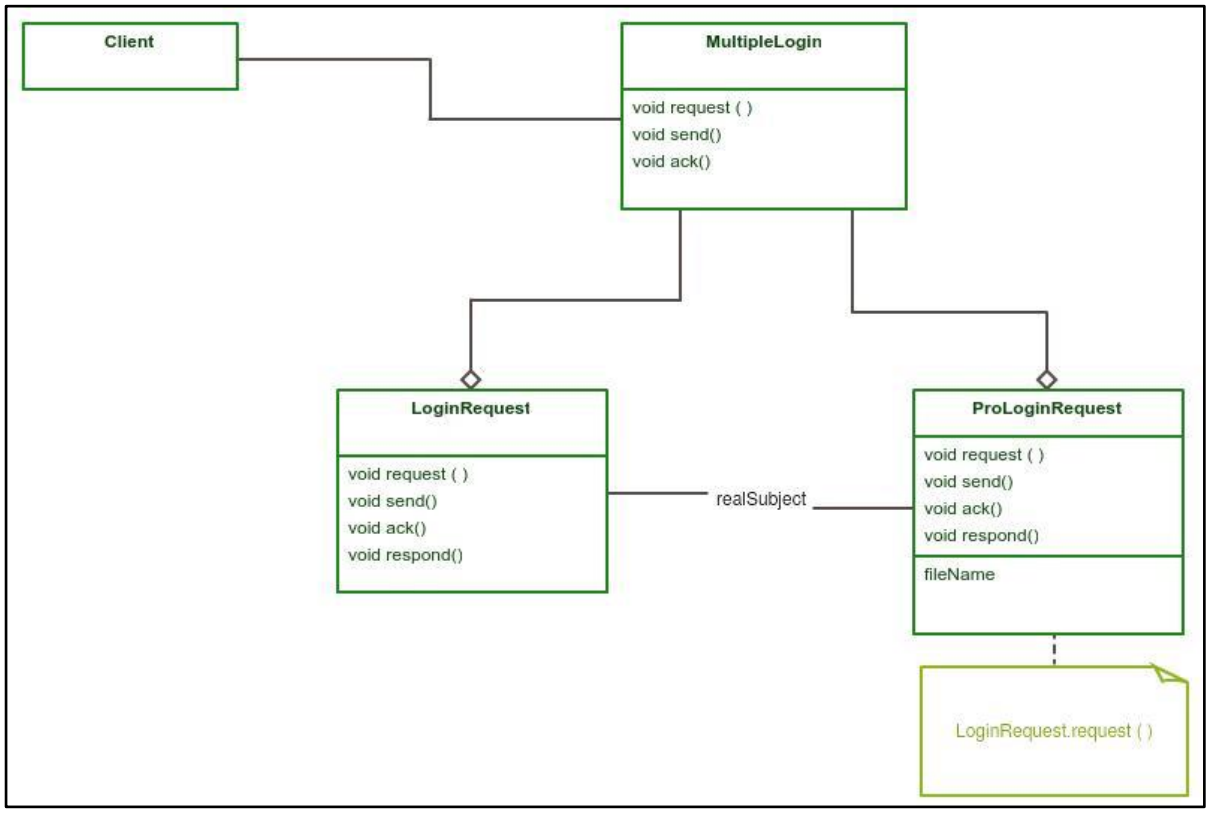

Fig 21: Proxy Design Pattern

Description:

The purpose of using Proxy design pattern is to serve our design goal for handling the login for both Staff and Manager. Proxy is used to handle the login for both Staff and Manager, as well as provides a layer of abstraction for a Manager modifying the client account permissions.

By using the Proxy design pattern we intend to use protection proxy to control the access to the original object. As Managers, Staff have different privileges and access rights, protection proxies address our goal in this application.