2014

Gordon Gear Gift Card POS System

Ehsan Rahman
Grand Valley State University

Gordon Gear Gift Card POS System

Sponsor by

Gordon®
FOOD SERVICE

By
Ehsan Rahman
December, 2014

Gordon Gear Gift Card POS System

By

Ehsan Rahman

A project submitted in partial fulfillment of the requirements for the degree of

Master of Science in

Computer Information Systems

at

Grand Valley State University

December, 2014

Dr. Tao Yonglei **Date**

Table of Contents

Abstract

Gordon Gear Gift Card POS system is a point of sale system that was created for the purpose of being able to make purchases efficiently and effectively during Gordon Food Service Annual Meeting. At the Annual Meeting, each employee receives a gift card in the amount of sixty five dollars, a gift from Gordon family. Employees have an option to use the gift card at the store that was temporarily set up at the Annual Meeting location for the sole purpose of employees to shop while they are enjoying food and entertainment. The Gordon Gear store offers a huge collection of all traditional GFS logo items. This temporary store consists of such things as, Gordon Gear apparel, sporting goods, toys, dishware, electronics, etc. If an employee chooses not to use their gift card at the temporary store created for the Annual Meeting, they have the option to spend it in any Gordon Food Marketplace Stores. When employees are done shopping, they take their purchase to one of the many cashiers. The cashier scans the item, then the point of sale Gift Card POS system calculates the amount owed by the employee and provides options for the employee to make payment, either by gift card, payroll deduction, or both. Once verified by the employee that the purchase items are correct, system process transactions real time to Gordon Food Service database.

Thousands of employees are making purchases with their card at the same time, which can cause a lot of chaos. Before this solution, the checkout application was a desktop application written in Visual Basic. Visual basic is no longer supported by Microsoft and newer operating system does not support Visual Basic installation, which is why another solution had to be implemented. This is why Web Gordon Gear POS system was created. Web Gordon Gear Apparel presentation layer written with the newest technologies, AngularJS, html5 and deployed under Weblogic 12c application server and followed top 10 Open Web Application Security Project (OWASP) guidelines.

Introduction

The Gordon Gear Store concept, first introduced in the year of 2003 was a big success. Back in 2003, the Gordon Gear Store setup was only in a one room hall with 50 items, mostly GFS brand apparel, and few miscellaneous items. There were only 2 cashiers that checked out employees at the Gordon Gear store. Since then, Gordon Food Service has grown to become the third largest privately held Food Distribution Company in the USA. The Application was written in Visual Basic 5, which later upgraded to Visual Basic 6. During the checkout process, cashiers hand keyed item numbers for the products, which was very time consuming and not efficient. After the store closes, one person goes to an individual checkout machine and transmits files using dialup connection to GFS server. Since then, technology has come a long way, but due to resource constraint, Gordon Gear POS system stayed the same. In the year of 2008, Microsoft ended their extended support contract for Visual Basic 6, which is still compatible with Windows XP.

With the tremendous growth of Gordon Food Service, there is a need for an efficient and scalable POS system to support thousands of employees that are going through the checkout process. New Gordon Gear Web POS system is written in a way to allow cashiers to check out faster and more efficiently and reliably.

Program Requirements

Business Requirements:

1. Application has to be able to run on both tablet and laptop
2. Ability to scan a gift card barcode
3. Ability to lookup employee information from barcode scan
4. Ability to scan product barcode
5. Ability to update and delete an item from the list
6. Ability to do payroll deduction, gift card deduction or both
7. Ability to update transactions real time to the database
8. Ability to send transaction receipt through email to the employee

Information Technology Requirements:

1. Web application has to be platform independent
2. Web application has should follow responsive design
3. Application should use LDAP for authentication
4. All traffic should be SSL, HTTPS

5. Application should use HTML5 and angular js
6. Password to the database layer has to be encrypted

Implementation

1.1. Development Deliveries

Gordon Gear POS system written in JDK7 and W3C compliant. HTML5 used for presentation layer with AngularJS and JQuery framework. Application uses Spring framework heavily for security and connection between component layer and Database Access layer. Application follows Twitter Bootstrap responsive design pattern and JUnit and Easy Mock for testing framework. Clover tool used for code coverage statistics and maven 3.0 to build the application and to deploy under WebLogic 12c environment. Source code for this application is version control using Subversion.

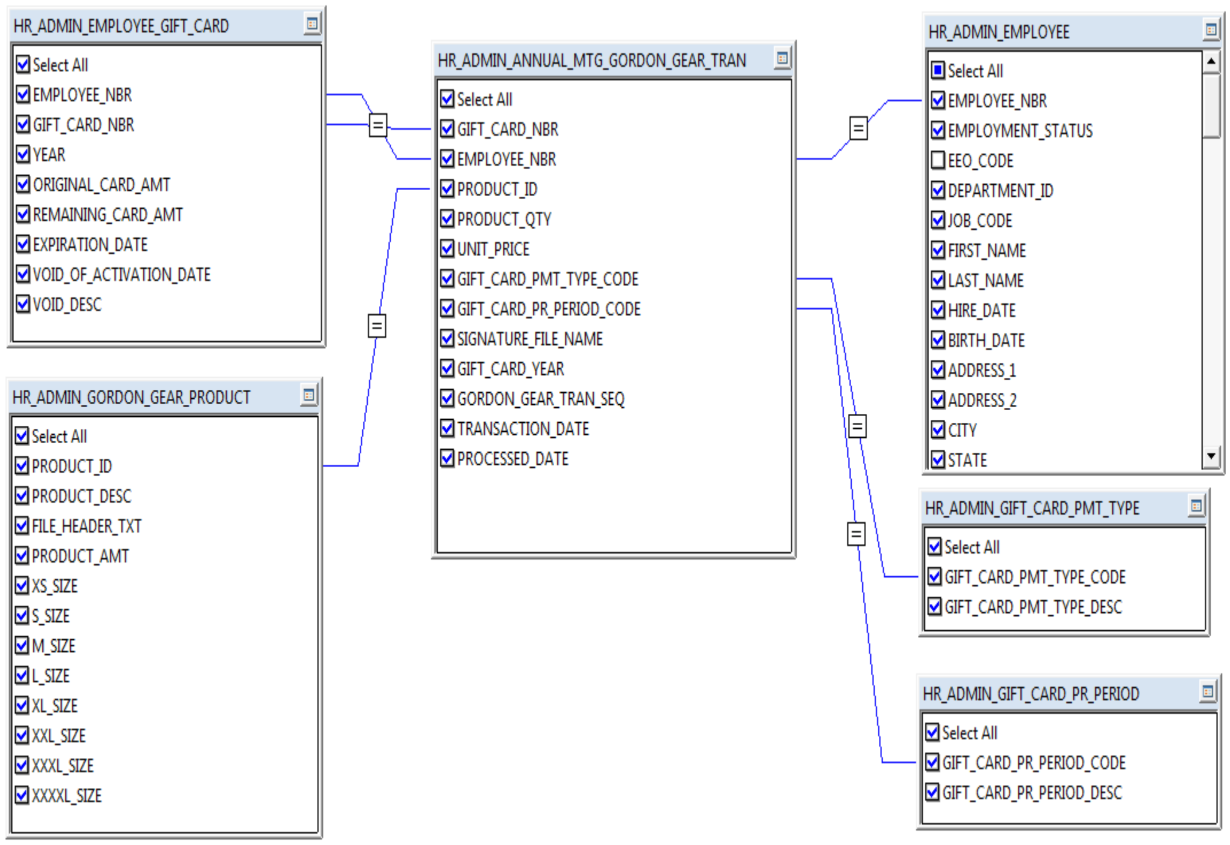
1.2. Required Deliveries

Access to the application is controlled by DSEE (LDAP) both for authentication and authorization. There are 2 different roles that user can get access to, Support user, that has access to all application privilege and General user role has access to checkout an employee. Backend database of the application is Oracle 11g with a special user account strictly access to only Gordon Gear application schema and tables.

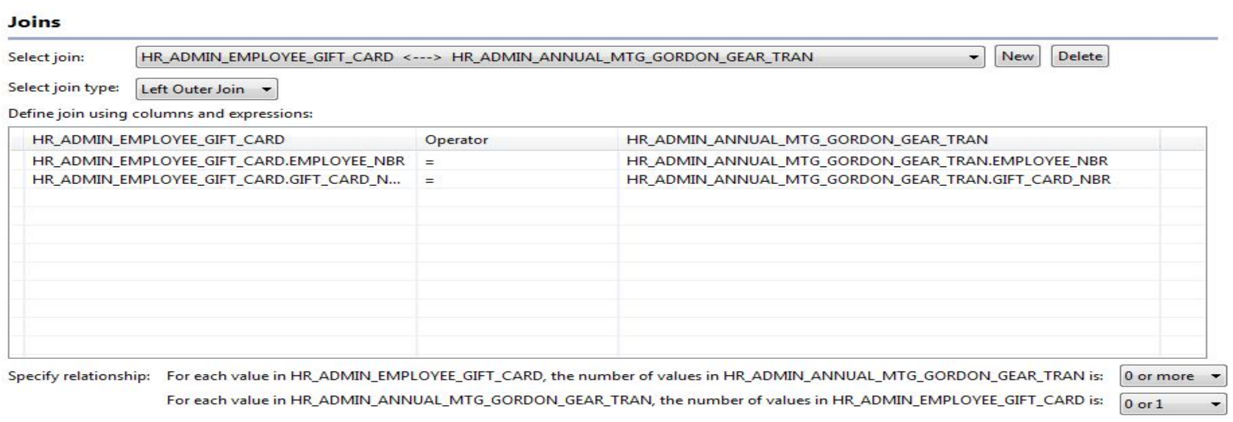
1.3. Entity Relationship Diagram

Below entity–relationship model (ER model) is a data model for for Gordon Gear POS System describing the data and information aspects of a business domain and its process requirements.

1.3.1. Diagram



1.3.2. Cardinalities



Joins

Select join: HR_ADMIN_EMPLOYEE_GIFT_CARD <---> HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN [New] [Delete]

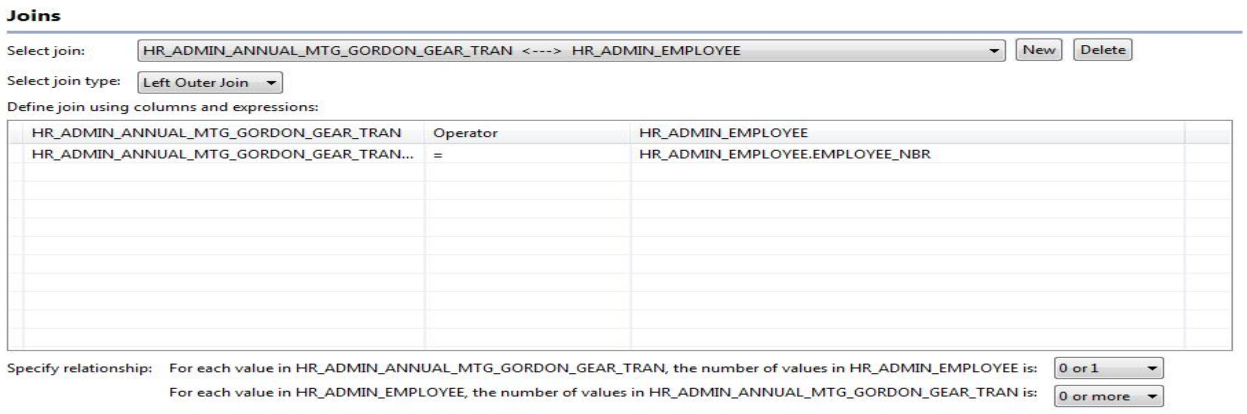
Select join type: Left Outer Join

Define join using columns and expressions:

HR_ADMIN_EMPLOYEE_GIFT_CARD	Operator	HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN
HR_ADMIN_EMPLOYEE_GIFT_CARD.EMPLOYEE_NBR	=	HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN.EMPLOYEE_NBR
HR_ADMIN_EMPLOYEE_GIFT_CARD.GIFT_CARD_N...	=	HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN.GIFT_CARD_NBR

Specify relationship: For each value in HR_ADMIN_EMPLOYEE_GIFT_CARD, the number of values in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN is: 0 or more

For each value in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN, the number of values in HR_ADMIN_EMPLOYEE_GIFT_CARD is: 0 or 1



Joins

Select join: HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN <---> HR_ADMIN_EMPLOYEE

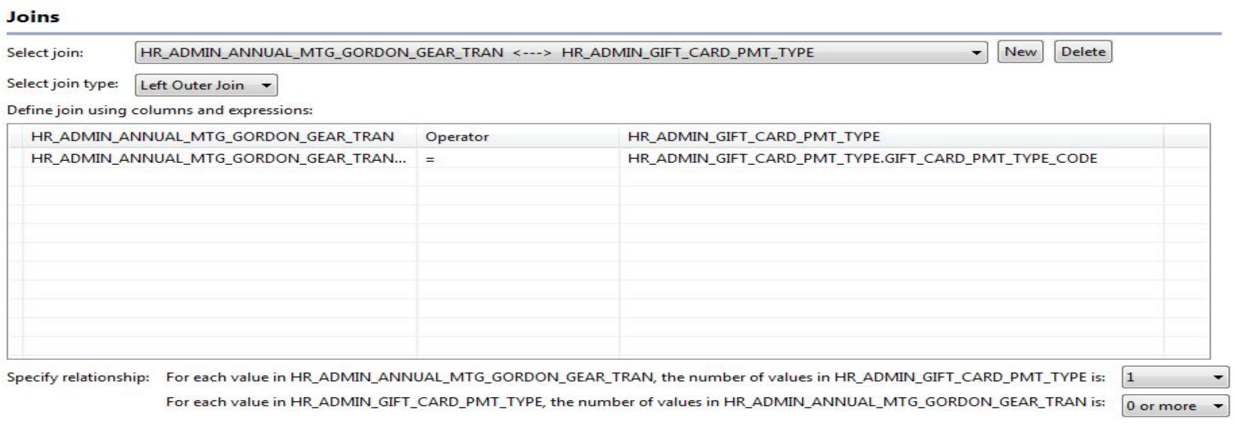
Select join type: Left Outer Join

Define join using columns and expressions:

HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN	Operator	HR_ADMIN_EMPLOYEE
HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN...	=	HR_ADMIN_EMPLOYEE.EMPLOYEE_NBR

Specify relationship: For each value in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN, the number of values in HR_ADMIN_EMPLOYEE is: 0 or 1

For each value in HR_ADMIN_EMPLOYEE, the number of values in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN is: 0 or more



Joins

Select join: HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN <---> HR_ADMIN_GIFT_CARD_PMT_TYPE

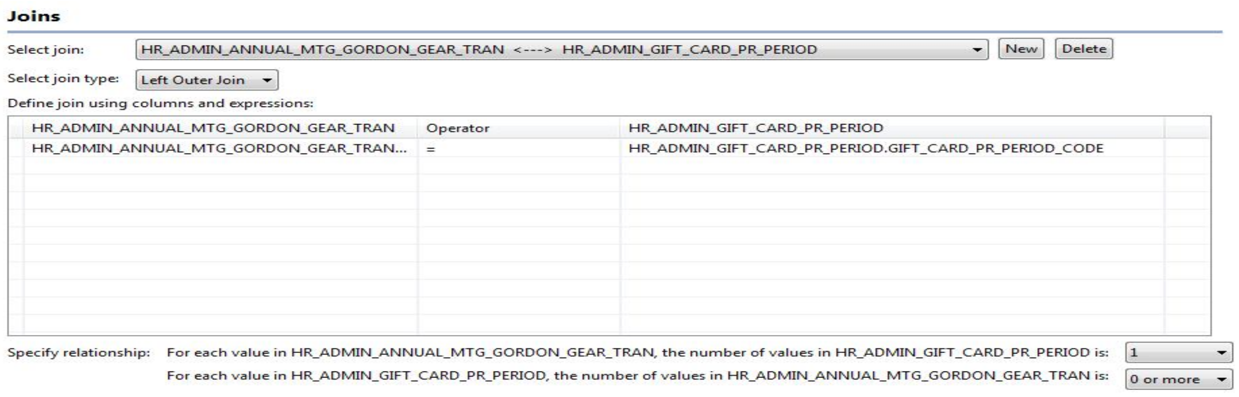
Select join type: Left Outer Join

Define join using columns and expressions:

HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN	Operator	HR_ADMIN_GIFT_CARD_PMT_TYPE
HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN...	=	HR_ADMIN_GIFT_CARD_PMT_TYPE.GIFT_CARD_PMT_TYPE_CODE

Specify relationship: For each value in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN, the number of values in HR_ADMIN_GIFT_CARD_PMT_TYPE is: 1

For each value in HR_ADMIN_GIFT_CARD_PMT_TYPE, the number of values in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN is: 0 or more



Joins

Select join: HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN <---> HR_ADMIN_GIFT_CARD_PR_PERIOD

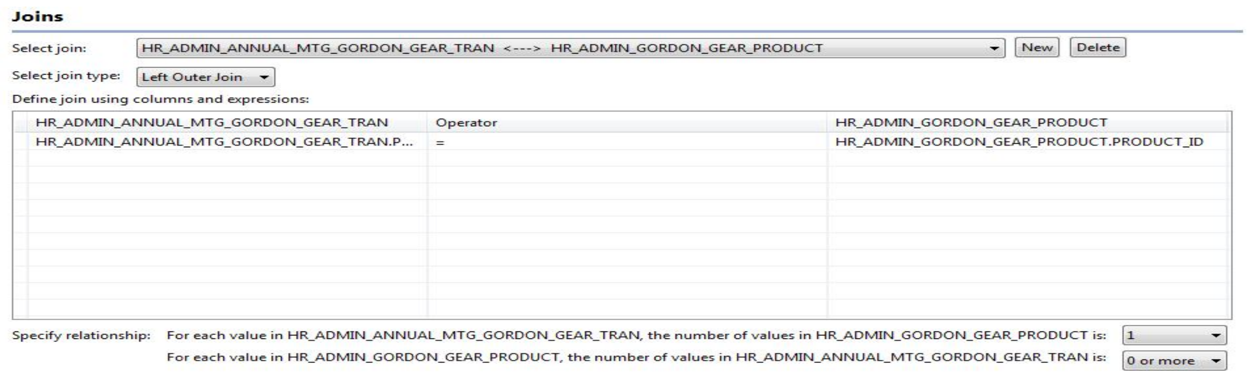
Select join type: Left Outer Join

Define join using columns and expressions:

HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN	Operator	HR_ADMIN_GIFT_CARD_PR_PERIOD
HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN...	=	HR_ADMIN_GIFT_CARD_PR_PERIOD.GIFT_CARD_PR_PERIOD_CODE

Specify relationship: For each value in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN, the number of values in HR_ADMIN_GIFT_CARD_PR_PERIOD is: 1

For each value in HR_ADMIN_GIFT_CARD_PR_PERIOD, the number of values in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN is: 0 or more



Joins

Select join: HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN <---> HR_ADMIN_GORDON_GEAR_PRODUCT [New] [Delete]

Select join type: Left Outer Join

Define join using columns and expressions:

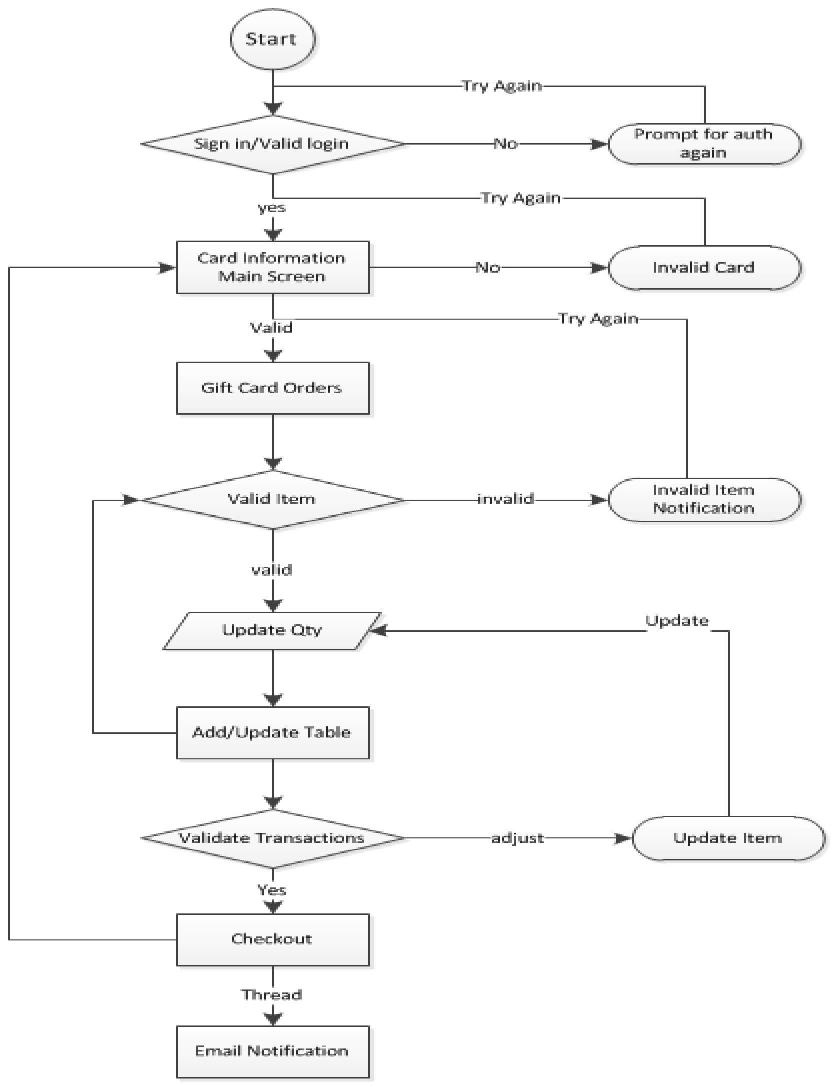
HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN	Operator	HR_ADMIN_GORDON_GEAR_PRODUCT
HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN.P...	=	HR_ADMIN_GORDON_GEAR_PRODUCT.PRODUCT_ID

Specify relationship: For each value in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN, the number of values in HR_ADMIN_GORDON_GEAR_PRODUCT is: 1

For each value in HR_ADMIN_GORDON_GEAR_PRODUCT, the number of values in HR_ADMIN_ANNUAL_MTG_GORDON_GEAR_TRAN is: 0 or more

1.4. Flow Chart

Gordon Gear POS system flow chart diagram displays basic relationship between each components and decision making process.



1.5. HTA Diagram

Hierarchical Task Analysis (HTA) diagram was drawn to show the task analysis of App in HTA a high-level task is decomposed into the hierarchy of subtasks. The HTA diagram helped in defining the overall task goals and sub-goals of the Gordon Gear POS System.

Home Page

The interface of the Home Page is very simple and gives straight forward options to the users of the App. This page provides three options to the users: add a new project, view all projects and search projects with help of different criteria. Login screen allows 2 options, enter employee number or scan gift card barcode. If an employee do not have gift card, employee number can be used to purchase Gordon Gear items. Application validates employee number and gift card against oracle database to make sure user has a valid card. If card or employee number could not be found, invalid card popup will be display.

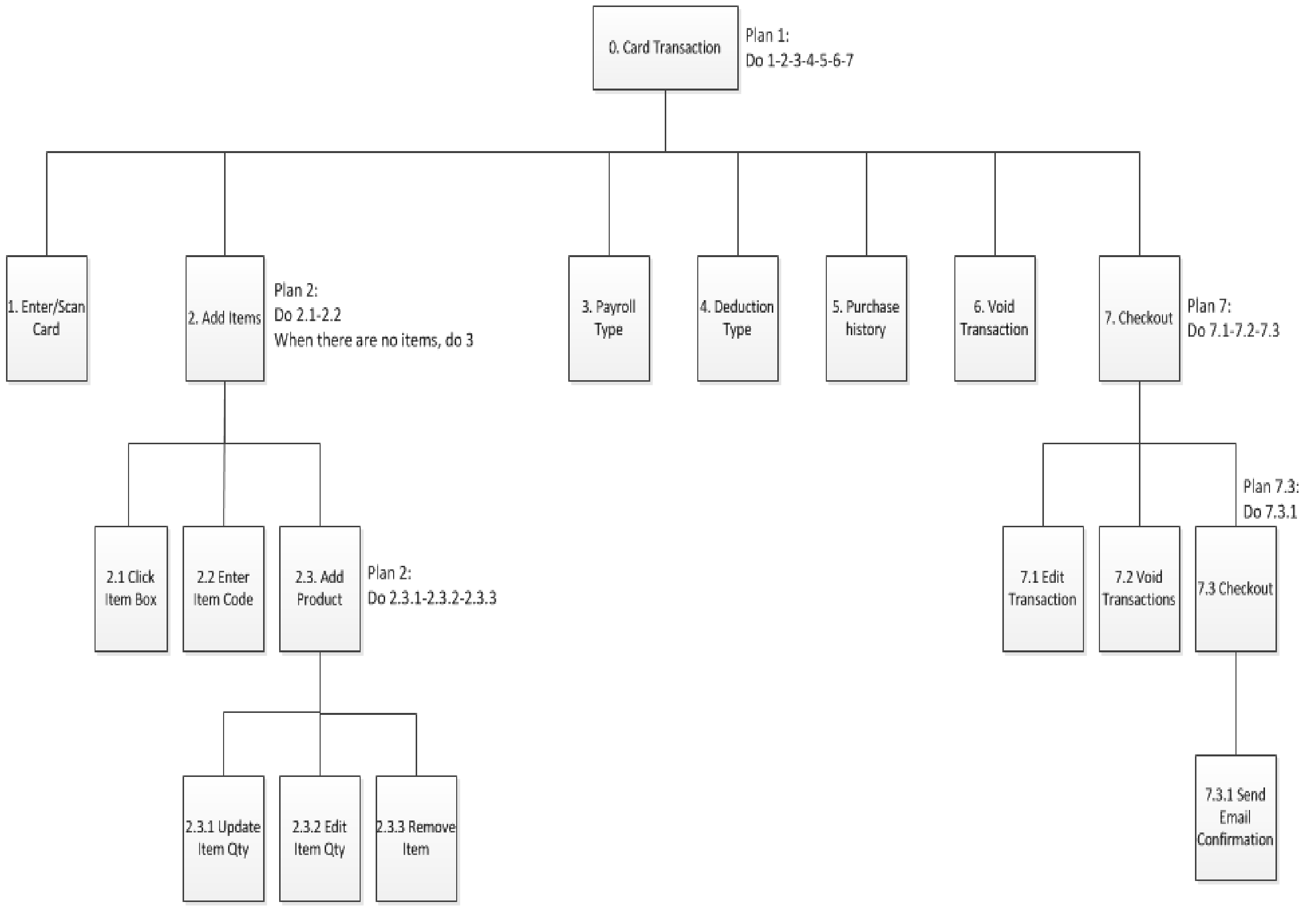
Add Items to the Page

Once a successful login occurs, data entry page appears to start scanning process. The page is divided into three rows. The first row autofocuses on the input field for scanning a Gordon Gear product. This product input field is required for the "add product" button to become active. A product ID can be entered manually as 4 digits, or, if scanned in, the program detects the length, parses the number, and adds the product to the table automatically while clearing and refocusing on the input box so that the next item can be immediately scanned.

The second row contains a dynamic table that updates as products are scanned in or removed, with the ability to enter custom amounts, increment, or decrement the quantity of each item in the list. Under the table is displayed the current order total, and how much money will remain on the gift card if the purchase is finalized. The remaining balance is green if money will remain on the card after the purchase and red if the order total exceeds the remaining amount on the gift card.

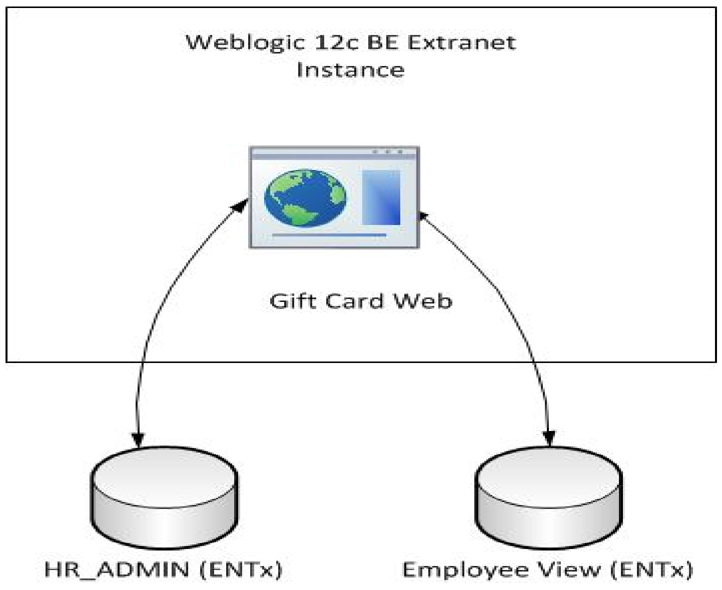
Checkout

The checkout page is an order confirmation page that displays the current products being purchased, the order total, and the payment type. If the payment type is payroll deduction only or the order total is large enough to warrant a necessary payroll deduction after using the gift card. Will also display the payroll deduction period selected and the total amount of money that will be deducted. If employee wish to cancel his/her purchase, transactions can voided with the “void transaction” button, which will take the user back to the login screen and clear their data. Otherwise, the user can click the “edit” button to go back to the home page to add or remove items, or change payment options. Clicking the checkout button will ask for one final order confirmation, and then write the transaction to the database. Checkout will also send employee a confirmation email with the purchase history.



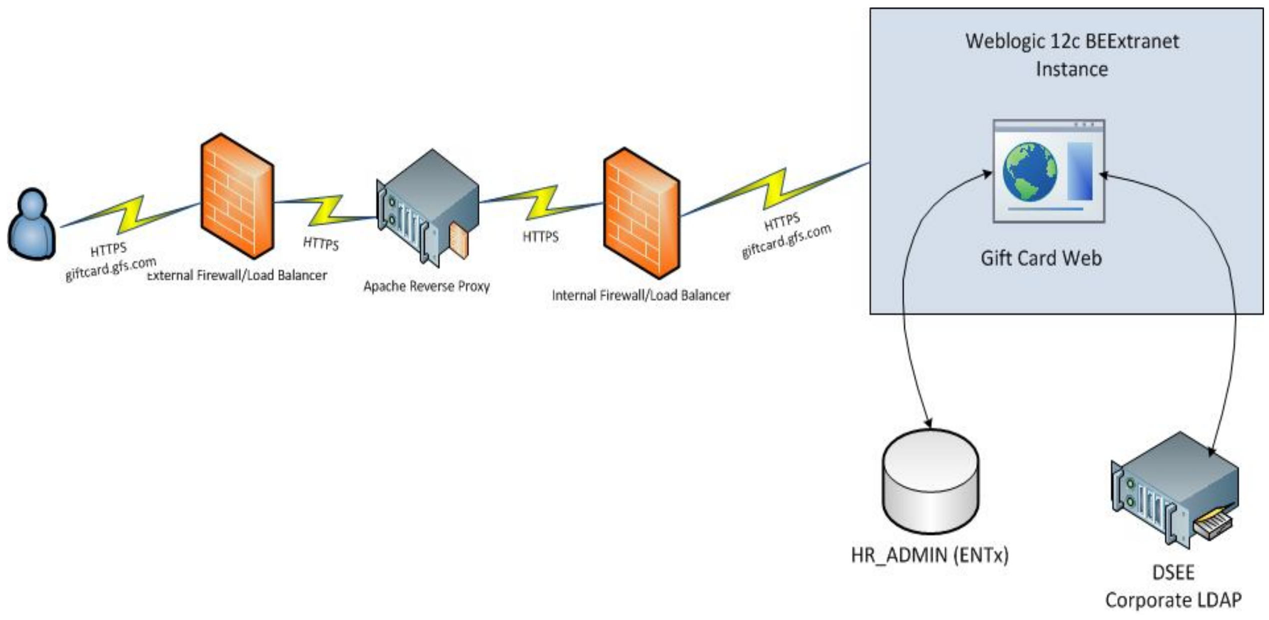
1.6. System Integration Diagram

Gordon Gear POS system is deployed under WebLogic application server. POS system integrates with its own instance of the database and Enterprise Employee table to retrieve basic employee information.



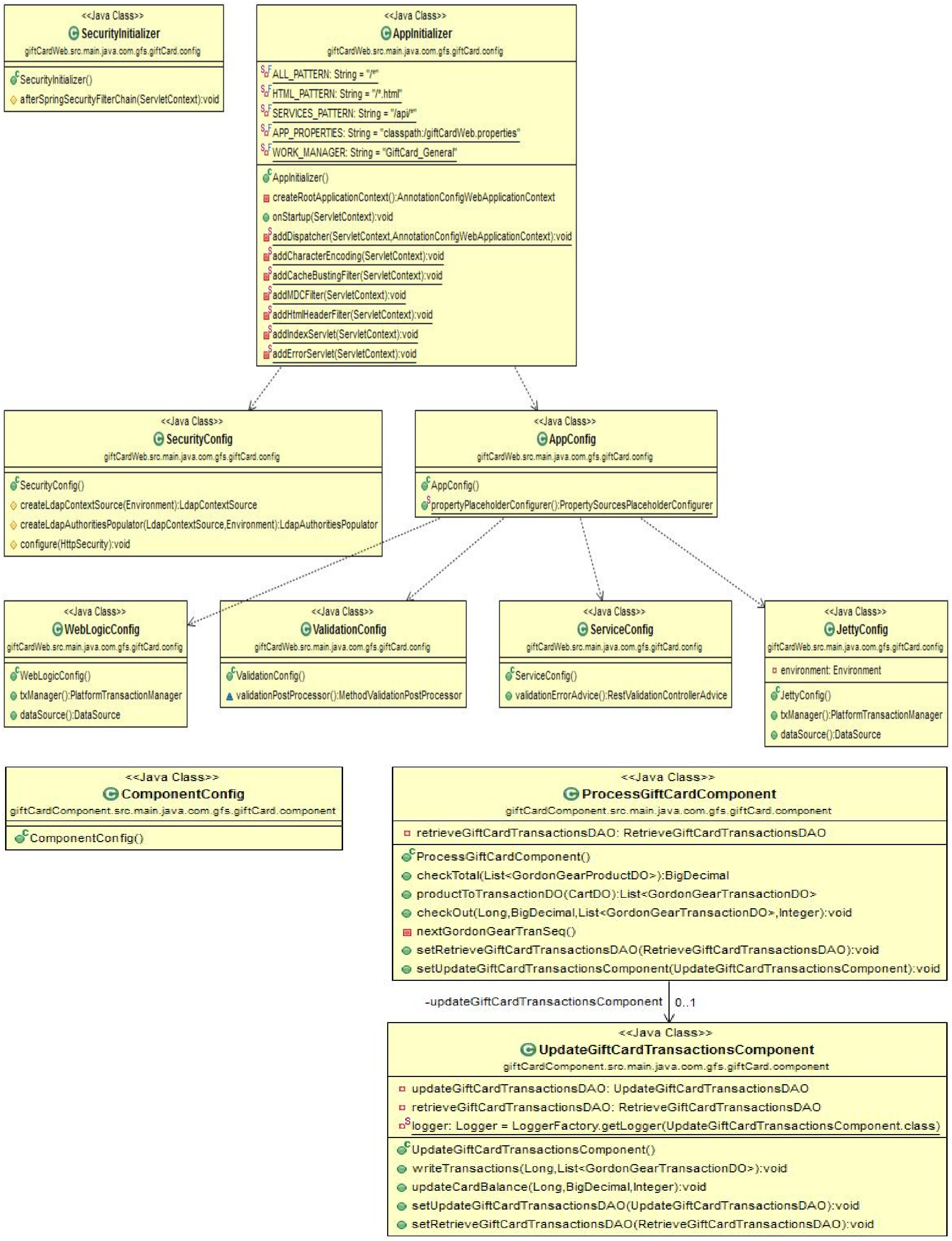
1.7. Deployment Model

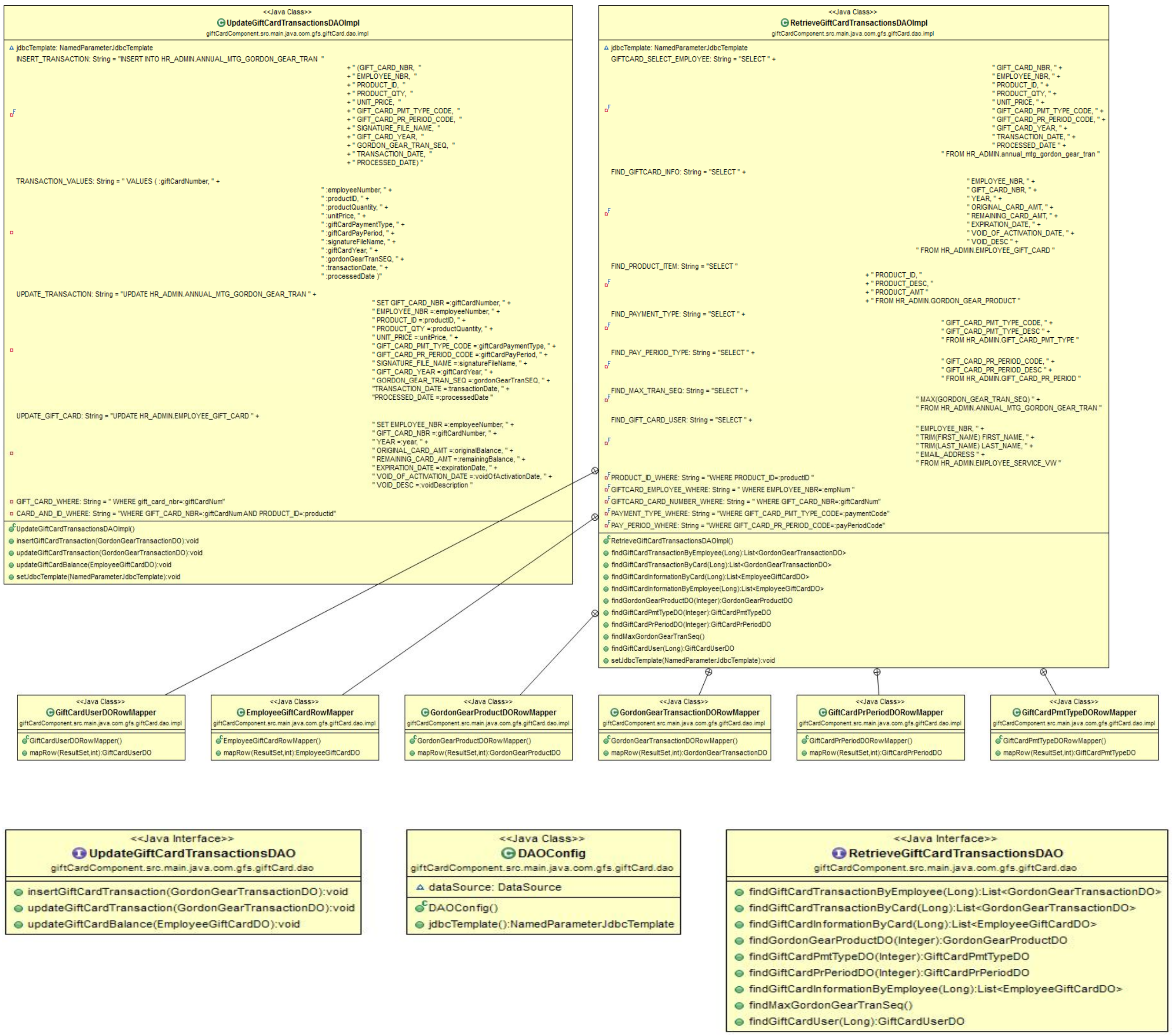
Once user enters https://giftcard.gfs.com, external Gordon Food Service load balancer terminates s SSL and traffic is re-encrypted between F5 (LB) and WebLogic. Once the request passes into the internal Load Balancer, apache whitelist the service URL for the application and reject requests on all other service URLs. Apache also limits the threads available to service requests to protect internal systems. System then popups basic authentication credentials to verify if the user is allowed to access the application via DSEE.



1.8. Class Diagram

Class diagram shows the relationship between classes and hierarchy.


<<Java Class>>
SecurityInitializer
giftCardWeb.src.main.java.com.gfs.giftCard.config
SecurityInitializer()
afterSpringSecurityFilterChain(ServletContext):void
<<Java Class>>
AppInitializer
giftCardWeb.src.main.java.com.gfs.giftCard.config
ALL_PATTERN: String = "/*"
HTML_PATTERN: String = "/*.html"
SERVICES_PATTERN: String = "/api/*"
APP_PROPERTIES: String = "classpath:/giftCardWeb.properties"
WORK_MANAGER: String = "GiftCard_General"
AppInitializer()
createRootApplicationContext():AnnotationConfigWebApplicationContext
onStartup(ServletContext):void
addDispatcher(ServletContext,AnnotationConfigWebApplicationContext):void
addCharacterEncoding(ServletContext):void
addCacheBustingFilter(ServletContext):void
addMDCFilter(ServletContext):void
addHtmlHeaderFilter(ServletContext):void
addIndexServlet(ServletContext):void
addErrorServlet(ServletContext):void
<<Java Class>>
SecurityConfig
giftCardWeb.src.main.java.com.gfs.giftCard.config
SecurityConfig()
createLdapContextSource(Environment):LdapContextSource
createLdapAuthoritiesPopulator(LdapContextSource,Environment):LdapAuthoritiesPopulator
configure(HttpSecurity):void
<<Java Class>>
AppConfig
giftCardWeb.src.main.java.com.gfs.giftCard.config
AppConfig()
propertyPlaceholderConfigurer():PropertySourcesPlaceholderConfigurer
<<Java Class>>
WebLogicConfig
giftCardWeb.src.main.java.com.gfs.giftCard.config
WebLogicConfig()
txManager():PlatformTransactionManager
dataSource():DataSource
<<Java Class>>
ValidationConfig
giftCardWeb.src.main.java.com.gfs.giftCard.config
ValidationConfig()
validationPostProcessor():MethodValidationPostProcessor
<<Java Class>>
ServiceConfig
giftCardWeb.src.main.java.com.gfs.giftCard.config
ServiceConfig()
validationErrorAdvice():RestValidationControllerAdvice
<<Java Class>>
JettyConfig
giftCardWeb.src.main.java.com.gfs.giftCard.config
environment: Environment
JettyConfig()
txManager():PlatformTransactionManager
dataSource():DataSource
<<Java Class>>
ComponentConfig
giftCardComponent.src.main.java.com.gfs.giftCard.component
ComponentConfig()
<<Java Class>>
ProcessGiftCardComponent
giftCardComponent.src.main.java.com.gfs.giftCard.component
retrieveGiftCardTransactionsDAO: RetrieveGiftCardTransactionsDAO
ProcessGiftCardComponent()
checkTotal(List<GordonGearProductDO>):BigDecimal
productToTransactionDO(CartDO):List<GordonGearTransactionDO>
checkOut(Long,BigDecimal,List<GordonGearTransactionDO>,Integer):void
nextGordonGearTranSeq()
setRetrieveGiftCardTransactionsDAO(RetrieveGiftCardTransactionsDAO):void
setUpdateGiftCardTransactionsComponent(UpdateGiftCardTransactionsComponent):void
-updateGiftCardTransactionsComponent
0..1
<<Java Class>>
UpdateGiftCardTransactionsComponent
giftCardComponent.src.main.java.com.gfs.giftCard.component
updateGiftCardTransactionsDAO: UpdateGiftCardTransactionsDAO
retrieveGiftCardTransactionsDAO: RetrieveGiftCardTransactionsDAO
logger: Logger = LoggerFactory.getLogger(UpdateGiftCardTransactionsComponent.class)
UpdateGiftCardTransactionsComponent()
writeTransactions(Long,List<GordonGearTransactionDO>):void
updateCardBalance(Long,BigDecimal,Integer):void
setUpdateGiftCardTransactionsDAO(UpdateGiftCardTransactionsDAO):void
setRetrieveGiftCardTransactionsDAO(RetrieveGiftCardTransactionsDAO):void



```
<<Java Class>>
UpdateGiftCardTransactionsDAOImpl
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

jdbcTemplate: NamedParameterJdbcTemplate
INSERT_TRANSACTION: String = "INSERT INTO HR_ADMIN.ANNUAL_MTG_GORDON_GEAR_TRAN  "
                                        + " (GIFT_CARD_NBR,  "
                                        + " EMPLOYEE_NBR,  "
                                        + " PRODUCT_ID,  "
                                        + " PRODUCT_QTY,  "
                                        + " UNIT_PRICE,  "
                                        + " GIFT_CARD_PMT_TYPE_CODE,  "
                                        + " GIFT_CARD_PR_PERIOD_CODE,  "
                                        + " SIGNATURE_FILE_NAME,  "
                                        + " GIFT_CARD_YEAR,  "
                                        + " GORDON_GEAR_TRAN_SEQ,  "
                                        + " TRANSACTION_DATE,  "
                                        + " PROCESSED_DATE) "

TRANSACTION_VALUES: String = " VALUES ( :giftCardNumber, " +
                                      " :employeeNumber, " +
                                      " :productID, " +
                                      " :productQuantity, " +
                                      " :unitPrice, " +
                                      " :giftCardPaymentType, " +
                                      " :giftCardPayPeriod, " +
                                      " :signatureFileName, " +
                                      " :giftCardYear, " +
                                      " :gordonGearTranSEQ, " +
                                      " :transactionDate, " +
                                      " :processedDate )"

UPDATE_TRANSACTION: String = "UPDATE HR_ADMIN.ANNUAL_MTG_GORDON_GEAR_TRAN " +
                                        " SET GIFT_CARD_NBR =:giftCardNumber, " +
                                        " EMPLOYEE_NBR =:employeeNumber, " +
                                        " PRODUCT_ID =:productID, " +
                                        " PRODUCT_QTY =:productQuantity, " +
                                        " UNIT_PRICE =:unitPrice, " +
                                        " GIFT_CARD_PMT_TYPE_CODE =:giftCardPaymentType, " +
                                        " GIFT_CARD_PR_PERIOD_CODE =:giftCardPayPeriod, " +
                                        " SIGNATURE_FILE_NAME =:signatureFileName, " +
                                        " GIFT_CARD_YEAR =:giftCardYear, " +
                                        " GORDON_GEAR_TRAN_SEQ =:gordonGearTranSEQ, " +
                                        "TRANSACTION_DATE =:transactionDate, " +
                                        "PROCESSED_DATE =:processedDate "

UPDATE_GIFT_CARD: String = "UPDATE HR_ADMIN.EMPLOYEE_GIFT_CARD " +
                                        " SET EMPLOYEE_NBR =:employeeNumber, " +
                                        " GIFT_CARD_NBR =:giftCardNumber, " +
                                        " YEAR =:year, " +
                                        " ORIGINAL_CARD_AMT =:originalBalance, " +
                                        " REMAINING_CARD_AMT =:remainingBalance, " +
                                        " EXPIRATION_DATE =:expirationDate, " +
                                        " VOID_OF_ACTIVATION_DATE =:voidOfActivationDate, " +
                                        " VOID_DESC =:voidDescription "

GIFT_CARD_WHERE: String = " WHERE gift_card_nbr=:giftCardNum"
CARD_AND_ID_WHERE: String = "WHERE GIFT_CARD_NBR=:giftCardNum AND PRODUCT_ID=:productid"

UpdateGiftCardTransactionsDAOImpl()
insertGiftCardTransaction(GordonGearTransactionDO):void
updateGiftCardTransaction(GordonGearTransactionDO):void
updateGiftCardBalance(EmployeeGiftCardDO):void
setJdbcTemplate(NamedParameterJdbcTemplate):void
```

```
<<Java Class>>
RetrieveGiftCardTransactionsDAOImpl
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

jdbcTemplate: NamedParameterJdbcTemplate
GIFTCARD_SELECT_EMPLOYEE: String = "SELECT " +
                                        " GIFT_CARD_NBR, " +
                                        " EMPLOYEE_NBR, " +
                                        " PRODUCT_ID, " +
                                        " PRODUCT_QTY, " +
                                        " UNIT_PRICE, " +
                                        " GIFT_CARD_PMT_TYPE_CODE, " +
                                        " GIFT_CARD_PR_PERIOD_CODE, " +
                                        " GIFT_CARD_YEAR, " +
                                        " TRANSACTION_DATE, " +
                                        " PROCESSED_DATE " +
                                  " FROM HR_ADMIN.annual_mtg_gordon_gear_tran "

FIND_GIFTCARD_INFO: String = "SELECT " +
                                     " EMPLOYEE_NBR, " +
                                     " GIFT_CARD_NBR, " +
                                     " YEAR, " +
                                     " ORIGINAL_CARD_AMT, " +
                                     " REMAINING_CARD_AMT, " +
                                     " EXPIRATION_DATE, " +
                                     " VOID_OF_ACTIVATION_DATE, " +
                                     " VOID_DESC " +
                               " FROM HR_ADMIN.EMPLOYEE_GIFT_CARD "

FIND_PRODUCT_ITEM: String = "SELECT "
                           + " PRODUCT_ID, "
                           + " PRODUCT_DESC, "
                           + " PRODUCT_AMT "
                           + " FROM HR_ADMIN.GORDON_GEAR_PRODUCT "

FIND_PAYMENT_TYPE: String = "SELECT " +
                                  " GIFT_CARD_PMT_TYPE_CODE, " +
                                  " GIFT_CARD_PMT_TYPE_DESC " +
                                  " FROM HR_ADMIN.GIFT_CARD_PMT_TYPE "

FIND_PAY_PERIOD_TYPE: String = "SELECT " +
                                  " GIFT_CARD_PR_PERIOD_CODE, " +
                                  " GIFT_CARD_PR_PERIOD_DESC " +
                                  " FROM HR_ADMIN.GIFT_CARD_PR_PERIOD "

FIND_MAX_TRAN_SEQ: String = "SELECT " +
                               " MAX(GORDON_GEAR_TRAN_SEQ) " +
                               " FROM HR_ADMIN.ANNUAL_MTG_GORDON_GEAR_TRAN "

FIND_GIFT_CARD_USER: String = "SELECT " +
                               " EMPLOYEE_NBR, " +
                               " TRIM(FIRST_NAME) FIRST_NAME, " +
                               " TRIM(LAST_NAME) LAST_NAME, " +
                               " EMAIL_ADDRESS " +
                               " FROM HR_ADMIN.EMPLOYEE_SERVICE_VW "

PRODUCT_ID_WHERE: String = "WHERE PRODUCT_ID=:productID "
GIFTCARD_EMPLOYEE_WHERE: String = " WHERE EMPLOYEE_NBR=:empNum "
GIFTCARD_CARD_NUMBER_WHERE: String = " WHERE GIFT_CARD_NBR=:giftCardNum"
PAYMENT_TYPE_WHERE: String = "WHERE GIFT_CARD_PMT_TYPE_CODE=:paymentCode"
PAY_PERIOD_WHERE: String = "WHERE GIFT_CARD_PR_PERIOD_CODE=:payPeriodCode"

RetrieveGiftCardTransactionsDAOImpl()
findGiftCardTransactionByEmployee(Long):List<GordonGearTransactionDO>
findGiftCardTransactionByCard(Long):List<GordonGearTransactionDO>
findGiftCardInformationByCard(Long):List<EmployeeGiftCardDO>
findGiftCardInformationByEmployee(Long):List<EmployeeGiftCardDO>
findGordonGearProductDO(Integer):GordonGearProductDO
findGiftCardPmtTypeDO(Integer):GiftCardPmtTypeDO
findGiftCardPrPeriodDO(Integer):GiftCardPrPeriodDO
findMaxGordonGearTranSeq()
findGiftCardUser(Long):GiftCardUserDO
setJdbcTemplate(NamedParameterJdbcTemplate):void
```

```
<<Java Class>>
GiftCardUserDORowMapper
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

GiftCardUserDORowMapper()
mapRow(ResultSet,int):GiftCardUserDO
```

```
<<Java Class>>
EmployeeGiftCardRowMapper
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

EmployeeGiftCardRowMapper()
mapRow(ResultSet,int):EmployeeGiftCardDO
```

```
<<Java Class>>
GordonGearProductDORowMapper
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

GordonGearProductDORowMapper()
mapRow(ResultSet,int):GordonGearProductDO
```

```
<<Java Class>>
GordonGearTransactionDORowMapper
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

GordonGearTransactionDORowMapper()
mapRow(ResultSet,int):GordonGearTransactionDO
```

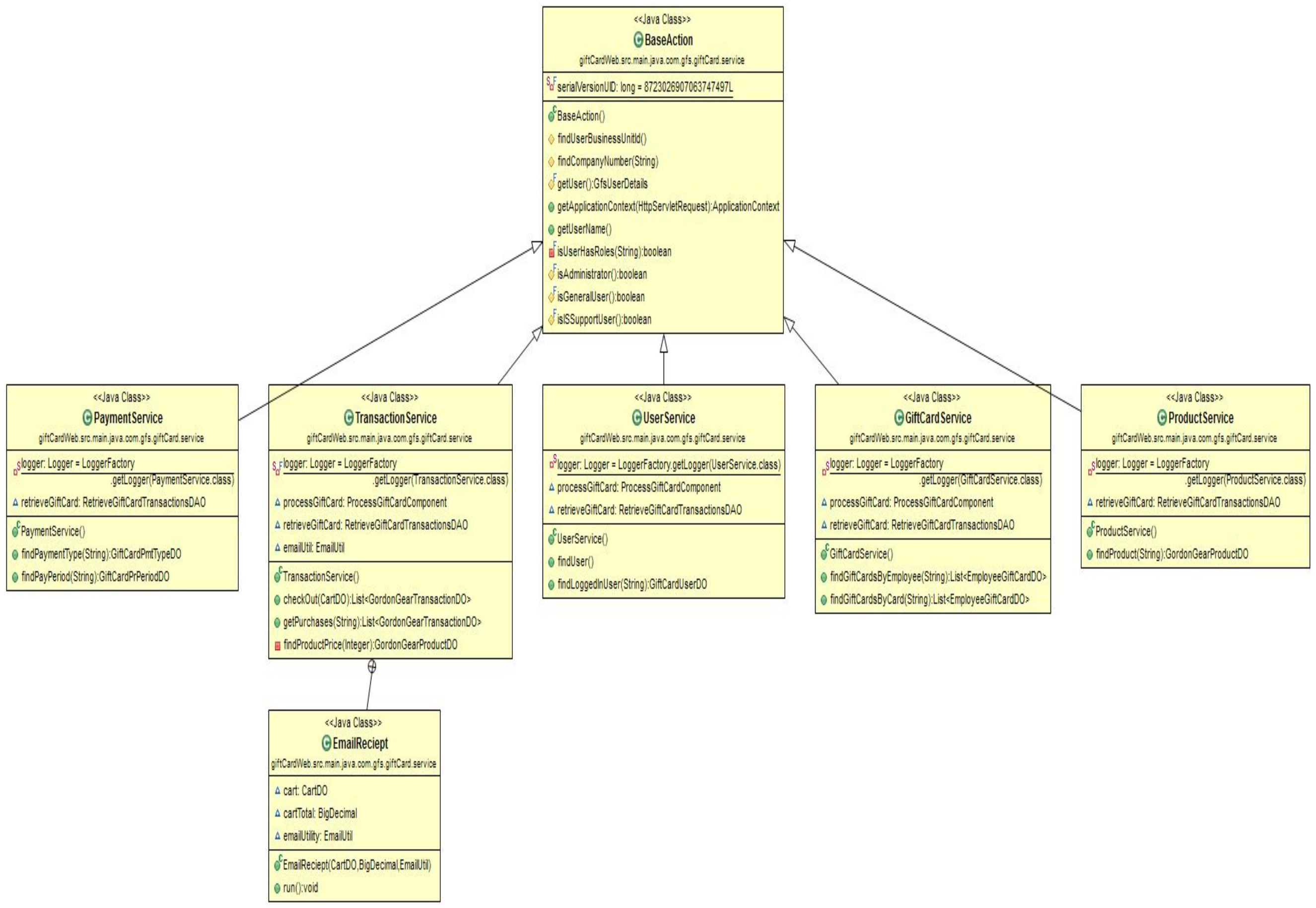
```
<<Java Class>>
GiftCardPrPeriodDORowMapper
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

GiftCardPrPeriodDORowMapper()
mapRow(ResultSet,int):GiftCardPrPeriodDO
```

```
<<Java Class>>
GiftCardPmtTypeDORowMapper
giftCardComponent.src.main.java.com.gfs.giftCard.dao.impl

GiftCardPmtTypeDORowMapper()
mapRow(ResultSet,int):GiftCardPmtTypeDO
```

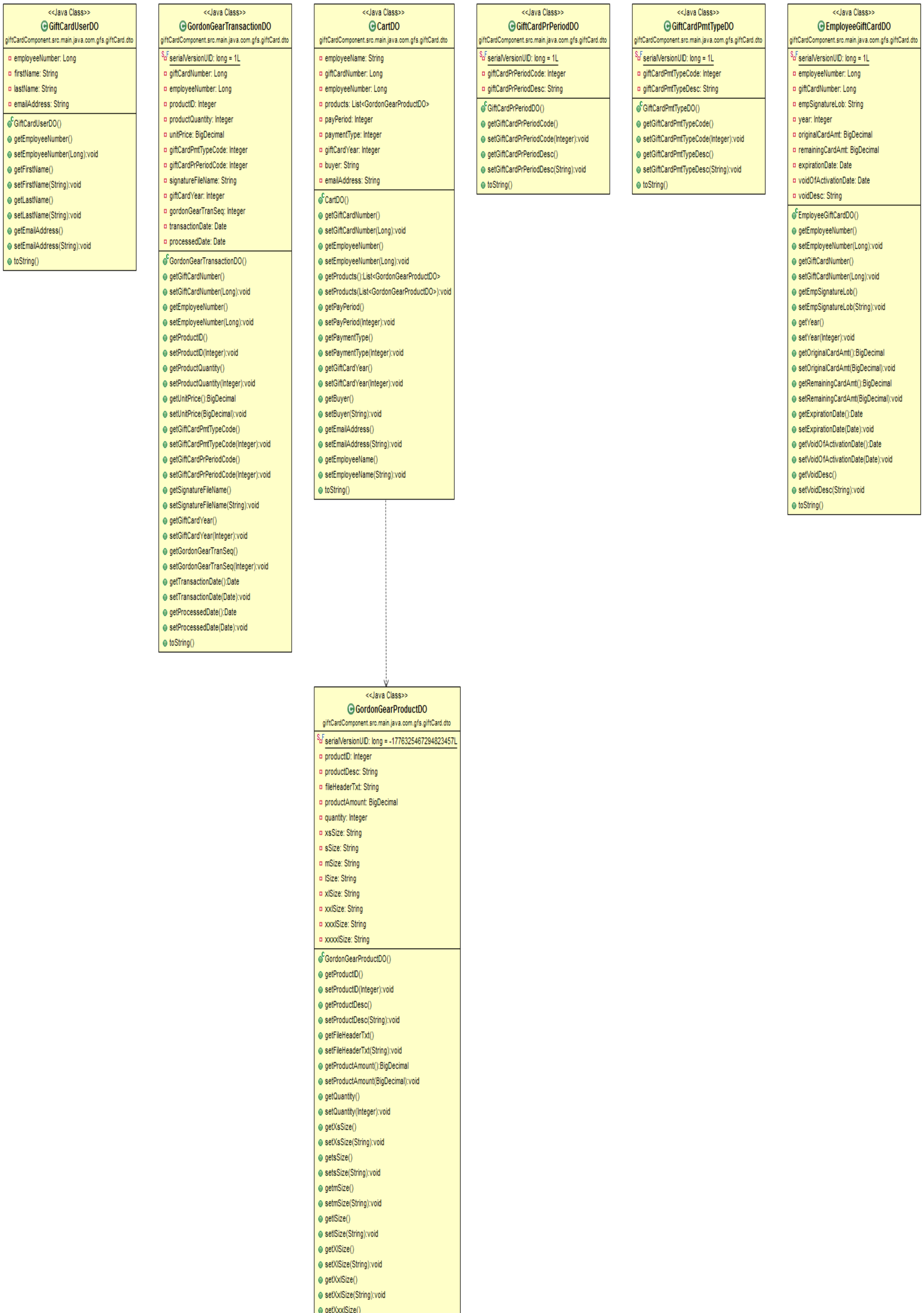
```
<<Java Interface>>
UpdateGiftCardTransactionsDAO
giftCardComponent.src.main.java.com.gfs.giftCard.dao

insertGiftCardTransaction(GordonGearTransactionDO):void
updateGiftCardTransaction(GordonGearTransactionDO):void
updateGiftCardBalance(EmployeeGiftCardDO):void
```

```
<<Java Class>>
DAOConfig
giftCardComponent.src.main.java.com.gfs.giftCard.dao

dataSource: DataSource
DAOConfig()
jdbcTemplate():NamedParameterJdbcTemplate
```

```
<<Java Interface>>
RetrieveGiftCardTransactionsDAO
giftCardComponent.src.main.java.com.gfs.giftCard.dao

findGiftCardTransactionByEmployee(Long):List<GordonGearTransactionDO>
findGiftCardTransactionByCard(Long):List<GordonGearTransactionDO>
findGiftCardInformationByCard(Long):List<EmployeeGiftCardDO>
findGordonGearProductDO(Integer):GordonGearProductDO
findGiftCardPmtTypeDO(Integer):GiftCardPmtTypeDO
findGiftCardPrPeriodDO(Integer):GiftCardPrPeriodDO
findGiftCardInformationByEmployee(Long):List<EmployeeGiftCardDO>
findMaxGordonGearTranSeq()
findGiftCardUser(Long):GiftCardUserDO
```



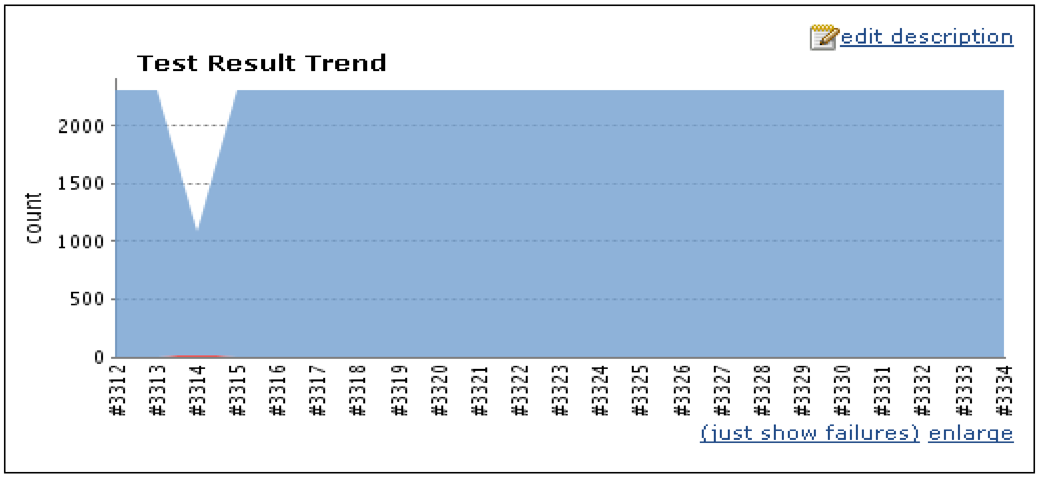
<<Java Class>>
BaseAction
giftCardWeb.src.main.java.com.gfs.giftCard.service
serialVersionUID: long = 8723026907063747497L
BaseAction()
findUserBusinessUnitId()
findCompanyNumber(String)
getUser():GfsUserDetails
getApplicationContext(HttpServletRequest):ApplicationContext
getUserName()
isUserHasRoles(String):boolean
isAdministrator():boolean
isGeneralUser():boolean
isISSupportUser():boolean
<<Java Class>>
PaymentService
giftCardWeb.src.main.java.com.gfs.giftCard.service
logger: Logger = LoggerFactory .getLogger(PaymentService.class)
retrieveGiftCard: RetrieveGiftCardTransactionsDAO
PaymentService()
findPaymentType(String):GiftCardPmtTypeDO
findPayPeriod(String):GiftCardPrPeriodDO
<<Java Class>>
TransactionService
giftCardWeb.src.main.java.com.gfs.giftCard.service
logger: Logger = LoggerFactory .getLogger(TransactionService.class)
processGiftCard: ProcessGiftCardComponent
retrieveGiftCard: RetrieveGiftCardTransactionsDAO
emailUtil: EmailUtil
TransactionService()
checkOut(CartDO):List<GordonGearTransactionDO>
getPurchases(String):List<GordonGearTransactionDO>
findProductPrice(Integer):GordonGearProductDO
<<Java Class>>
UserService
giftCardWeb.src.main.java.com.gfs.giftCard.service
logger: Logger = LoggerFactory.getLogger(UserService.class)
processGiftCard: ProcessGiftCardComponent
retrieveGiftCard: RetrieveGiftCardTransactionsDAO
UserService()
findUser()
findLoggedInUser(String):GiftCardUserDO
<<Java Class>>
GiftCardService
giftCardWeb.src.main.java.com.gfs.giftCard.service
logger: Logger = LoggerFactory .getLogger(GiftCardService.class)
processGiftCard: ProcessGiftCardComponent
retrieveGiftCard: RetrieveGiftCardTransactionsDAO
GiftCardService()
findGiftCardsByEmployee(String):List<EmployeeGiftCardDO>
findGiftCardsByCard(String):List<EmployeeGiftCardDO>
<<Java Class>>
ProductService
giftCardWeb.src.main.java.com.gfs.giftCard.service
logger: Logger = LoggerFactory .getLogger(ProductService.class)
retrieveGiftCard: RetrieveGiftCardTransactionsDAO
ProductService()
findProduct(String):GordonGearProductDO
<<Java Class>>
EmailReciept
giftCardWeb.src.main.java.com.gfs.giftCard.service
cart: CartDO
cartTotal: BigDecimal
emailUtility: EmailUtil
EmailReciept(CartDO,BigDecimal,EmailUtil)
run():void



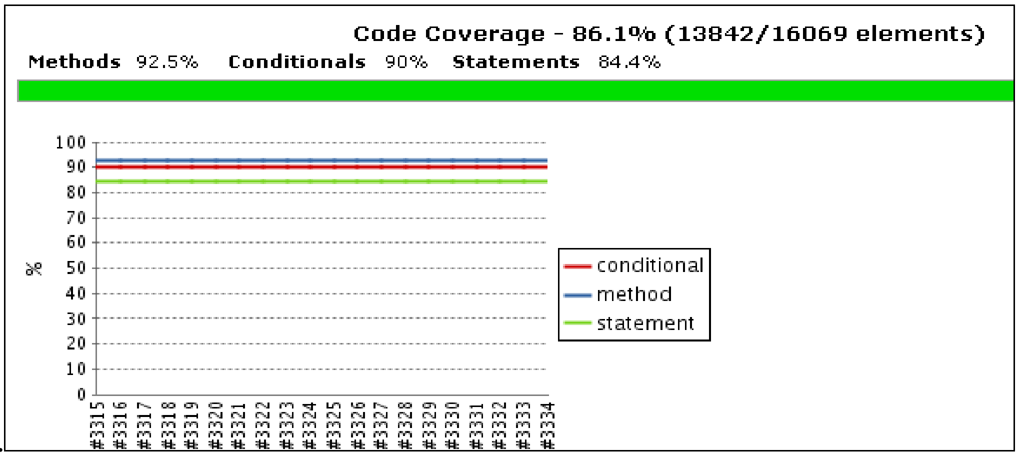
<<Java Class>>
GiftCardUserDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
employeeNumber: Long
firstName: String
lastName: String
emailAddress: String
GiftCardUserDO()
getEmployeeNumber()
setEmployeeNumber(Long):void
getFirstName()
setFirstName(String):void
getLastName()
setLastName(String):void
getEmailAddress()
setEmailAddress(String):void
toString()
<<Java Class>>
GordonGearTransactionDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
serialVersionUID: long = 1L
giftCardNumber: Long
employeeNumber: Long
productID: Integer
productQuantity: Integer
unitPrice: BigDecimal
giftCardPmtTypeCode: Integer
giftCardPrPeriodCode: Integer
signatureFileName: String
giftCardYear: Integer
gordonGearTranSeq: Integer
transactionDate: Date
processedDate: Date
GordonGearTransactionDO()
getGiftCardNumber()
setGiftCardNumber(Long):void
getEmployeeNumber()
setEmployeeNumber(Long):void
getProductID()
setProductID(Integer):void
getProductQuantity()
setProductQuantity(Integer):void
getUnitPrice():BigDecimal
setUnitPrice(BigDecimal):void
getGiftCardPmtTypeCode()
setGiftCardPmtTypeCode(Integer):void
getGiftCardPrPeriodCode()
setGiftCardPrPeriodCode(Integer):void
getSignatureFileName()
setSignatureFileName(String):void
getGiftCardYear()
setGiftCardYear(Integer):void
getGordonGearTranSeq()
setGordonGearTranSeq(Integer):void
getTransactionDate():Date
setTransactionDate(Date):void
getProcessedDate():Date
setProcessedDate(Date):void
toString()
<<Java Class>>
CartDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
employeeName: String
giftCardNumber: Long
employeeNumber: Long
products: List<GordonGearProductDO>
payPeriod: Integer
paymentType: Integer
giftCardYear: Integer
buyer: String
emailAddress: String
CartDO()
getGiftCardNumber()
setGiftCardNumber(Long):void
getEmployeeNumber()
setEmployeeNumber(Long):void
getProducts():List<GordonGearProductDO>
setProducts(List<GordonGearProductDO>):void
getPayPeriod()
setPayPeriod(Integer):void
getPaymentType()
setPaymentType(Integer):void
getGiftCardYear()
setGiftCardYear(Integer):void
getBuyer()
setBuyer(String):void
getEmailAddress()
setEmailAddress(String):void
getEmployeeName()
setEmployeeName(String):void
toString()
<<Java Class>>
GiftCardPrPeriodDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
serialVersionUID: long = 1L
giftCardPrPeriodCode: Integer
giftCardPrPeriodDesc: String
GiftCardPrPeriodDO()
getGiftCardPrPeriodCode()
setGiftCardPrPeriodCode(Integer):void
getGiftCardPrPeriodDesc()
setGiftCardPrPeriodDesc(String):void
toString()
<<Java Class>>
GiftCardPmtTypeDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
serialVersionUID: long = 1L
giftCardPmtTypeCode: Integer
giftCardPmtTypeDesc: String
GiftCardPmtTypeDO()
getGiftCardPmtTypeCode()
setGiftCardPmtTypeCode(Integer):void
getGiftCardPmtTypeDesc()
setGiftCardPmtTypeDesc(String):void
toString()
<<Java Class>>
EmployeeGiftCardDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
serialVersionUID: long = 1L
employeeNumber: Long
giftCardNumber: Long
empSignatureLob: String
year: Integer
originalCardAmt: BigDecimal
remainingCardAmt: BigDecimal
expirationDate: Date
voidOfActivationDate: Date
voidDesc: String
EmployeeGiftCardDO()
getEmployeeNumber()
setEmployeeNumber(Long):void
getGiftCardNumber()
setGiftCardNumber(Long):void
getEmpSignatureLob()
setEmpSignatureLob(String):void
getYear()
setYear(Integer):void
getOriginalCardAmt():BigDecimal
setOriginalCardAmt(BigDecimal):void
getRemainingCardAmt():BigDecimal
setRemainingCardAmt(BigDecimal):void
getExpirationDate():Date
setExpirationDate(Date):void
getVoidOfActivationDate():Date
setVoidOfActivationDate(Date):void
getVoidDesc()
setVoidDesc(String):void
toString()
<<Java Class>>
GordonGearProductDO
giftCardComponent.src.main.java.com.gfs.giftCard.dto
serialVersionUID: long = -1776325467294823457L
productID: Integer
productDesc: String
fileHeaderTxt: String
productAmount: BigDecimal
quantity: Integer
xsSize: String
sSize: String
mSize: String
lSize: String
xlSize: String
xxlSize: String
xxxlSize: String
xxxxlSize: String
GordonGearProductDO()
getProductID()
setProductID(Integer):void
getProductDesc()
setProductDesc(String):void
getFileHeaderTxt()
setFileHeaderTxt(String):void
getProductAmount():BigDecimal
setProductAmount(BigDecimal):void
getQuantity()
setQuantity(Integer):void
getXsSize()
setXsSize(String):void
getsSize()
setsSize(String):void
getmSize()
setmSize(String):void
getlSize()
setlSize(String):void
getXlSize()
setXlSize(String):void
getXxlSize()
setXxlSize(String):void
getXxxlSize()

1.9. Test Coverage

For Gordon Gear Gift Card application, I used Apache Subversion to maintain source code. This was the company's standard repository for source code control. They also created a master test plan to gather all of the information necessary to plan and control the test effort. In addition to this, unit tests, application integration tests and system integration tests were built using Java to minimize breakage of business functionality. All test cases were integrated using a product called Jenkins, an extensible continuous integration engine, which was linked to the subversion code repository. During the development stage, when any code checked in to subversion, Jenkins built the project and tests to find any regressions in the code.

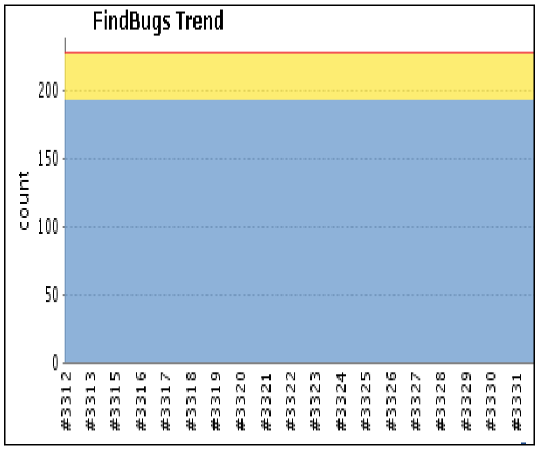


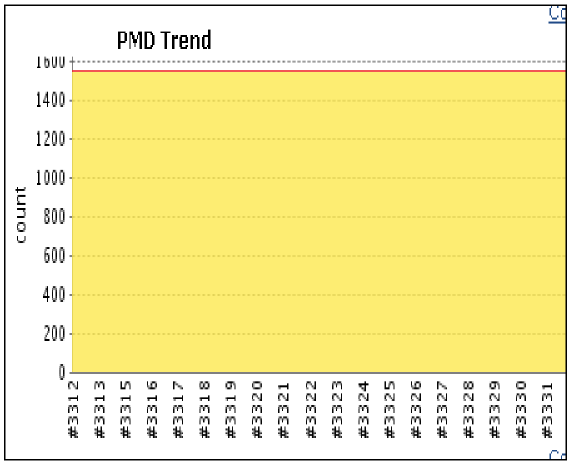
Also added an open source tool called Clover that reads through the new code and determines the percentage of the code that is covered by your test case to ensure business logic and application logic perform according to the use case specification.



.

A software tool called FindBugs was implemented which uses static analysis to look for bugs, memory leaks and code optimization in java code. This worked in conjunction with Jenkins to show how and where code optimizations could be made to increase code integrity. Jenkins was also integrated with PMD. PMD was used to scan Java source code and locate potential problems like dead code, suboptimal code, and duplicate code.

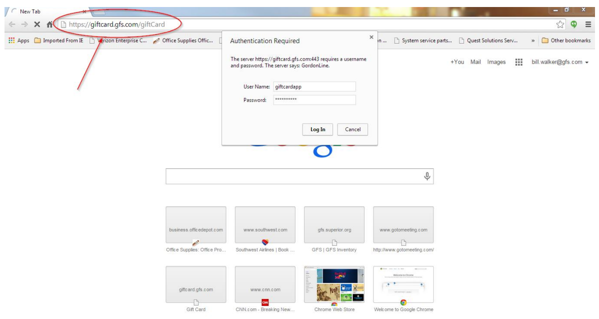
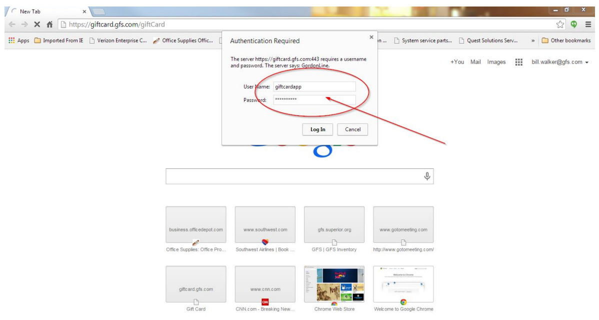
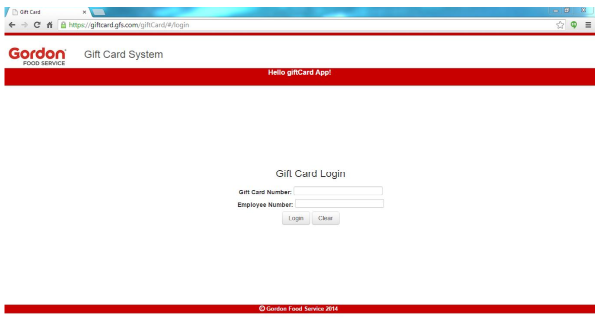
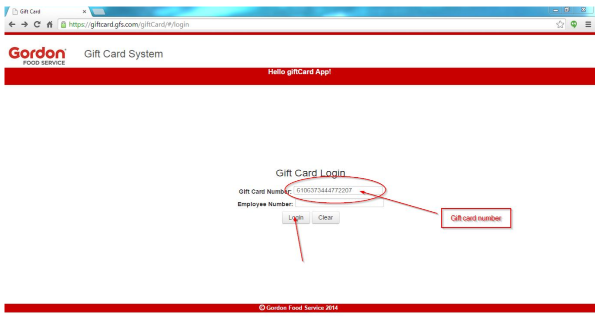


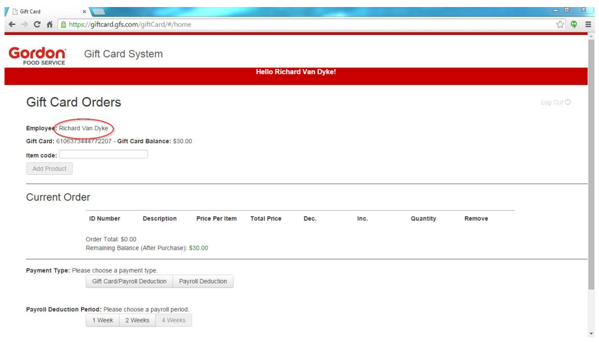
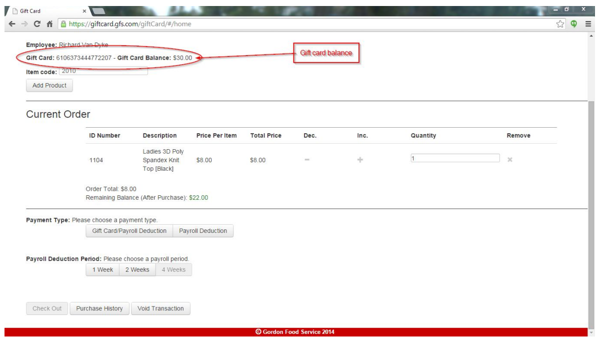
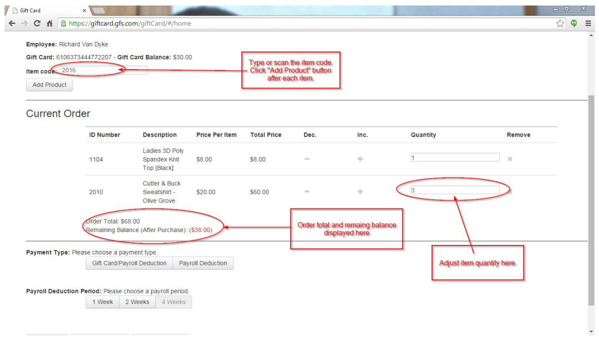
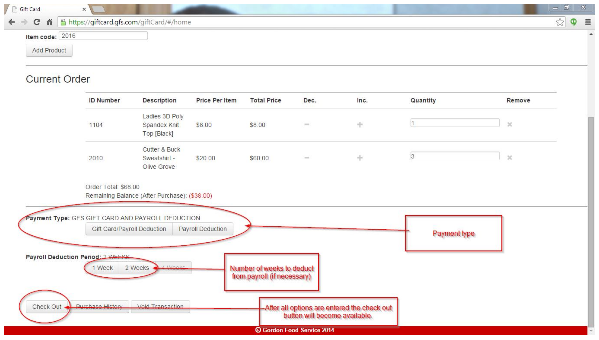


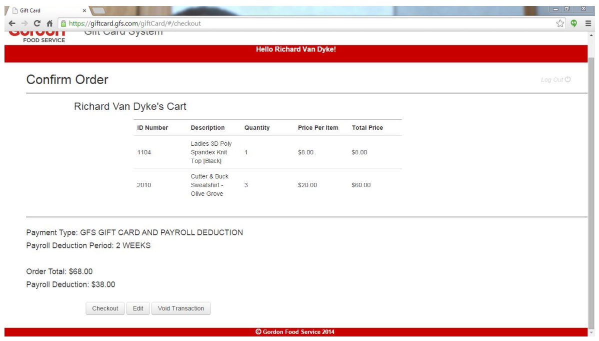
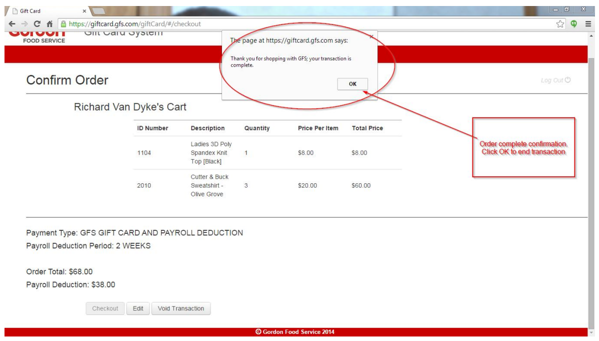
User Manual

Web Address: https://giftcard.gfs.com/giftCard (this is case sensitive)

1.	Open Google Chrome Browser	

2.	Type "https://giftcard.gfs.com/giftCard" on the address line (case sensitive, no quotes)	 https://giftcard.gfs.com/giftCard Authentication Required The server https://giftcard.gfs.com:443 requires a username and password. The server says: GordonLine. User Name: giftcardapp Password: ********** Log In Cancel
3.	Enter the username and password provided at the prompt User Name is: XXXXXXX Password is: XXXXXXX	 https://giftcard.gfs.com/giftCard Authentication Required The server https://giftcard.gfs.com:443 requires a username and password. The server says: GordonLine. User Name: giftcardapp Password: ********** Log In Cancel
4.	You are now at the main sales screen	 https://giftcard.gfs.com/giftCard/#/login Gordon FOOD SERVICE Gift Card System Hello giftCard App! Gift Card Login Gift Card Number: Employee Number: Login Clear © Gordon Food Service 2014
5.	Scan the employee's gift card bar code with the scanners supplied, or manually enter the gift card number. You can also enter the employee's employee number. Click login.	 https://giftcard.gfs.com/giftCard/#/login Gordon FOOD SERVICE Gift Card System Hello giftCard App! Gift Card Login Gift Card Number: 6106373444772207 Employee Number: Login Clear Gift card number © Gordon Food Service 2014

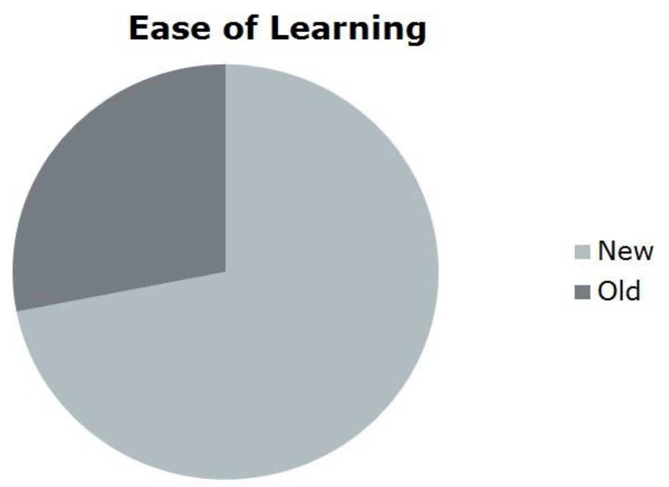
6.	You are now at the main ordering screen. Please verify the employee's name is correct before proceeding with the sale.	
7.	The remaining balance on the employee's gift card is also displayed here in real time.	
8.	Scan or type the item code(s) for the items the employee has selected. Adjust the quantity or scan item again if the employee has selected more than one of an item. The remaining balance is displayed as each item is added. Negative numbers indicate the gift card is used up and the negative amount will deducted through payroll.	 Type or scan the item code. Click "Add Product" button after each item. Order total and remaing balance displayed here Adjust item quantity here.
9.	Once all items have been entered you must select the payment type (and number of weeks if a payroll deduction is requested or required) before the "Check Out" button will become active.	 Payment type Number of weeks to deduct from payroll (if necessary) After all options are entered the check out button will become available.

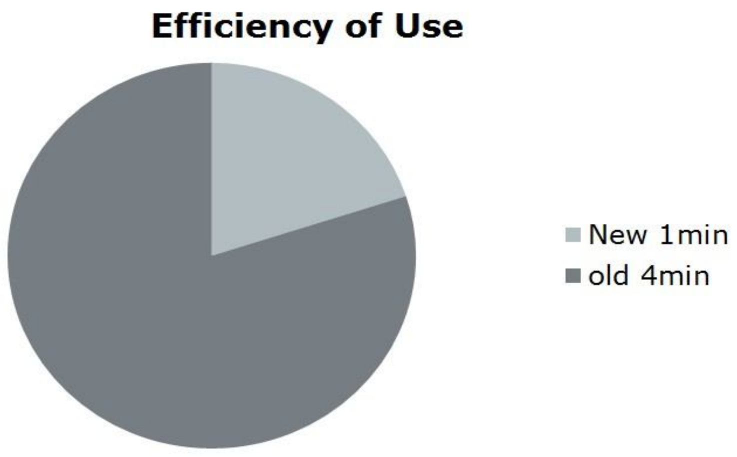
10.	A confirm order page is displayed. Verify all items are correct, then proceed to click “Check Out”	 https://giftcard.gfs.com/giftCard/#/checkout Hello Richard Van Dyke! Confirm Order Richard Van Dyke's Cart ID Number · Description · Quantity · Price Per Item · Total Price 1104 · Ladies 3D Poly Spandex Knit Top [Black] · 1 · $8.00 · $8.00 2010 · Cutter & Buck Sweatshirt - Olive Grove · 3 · $20.00 · $60.00 Payment Type: GFS GIFT CARD AND PAYROLL DEDUCTION Payroll Deduction Period: 2 WEEKS Order Total: $68.00 Payroll Deduction: $38.00 © Gordon Food Service 2014
11.	A pop window will confirm the transaction completed successfully. The employee will receive an email to their GFS address with their order details. Click OK to set up for the next employee transaction.	 The page at https://giftcard.gfs.com says: Thank you for shopping with GFS; your transaction is complete. Order complete confirmation. Click OK to end transaction Confirm Order Richard Van Dyke's Cart ID Number · Description · Quantity · Price Per Item · Total Price 1104 · Ladies 3D Poly Spandex Knit Top [Black] · 1 · $8.00 · $8.00 2010 · Cutter & Buck Sweatshirt - Olive Grove · 3 · $20.00 · $60.00 Payment Type: GFS GIFT CARD AND PAYROLL DEDUCTION Payroll Deduction Period: 2 WEEKS Order Total: $68.00 Payroll Deduction: $38.00 © Gordon Food Service 2014

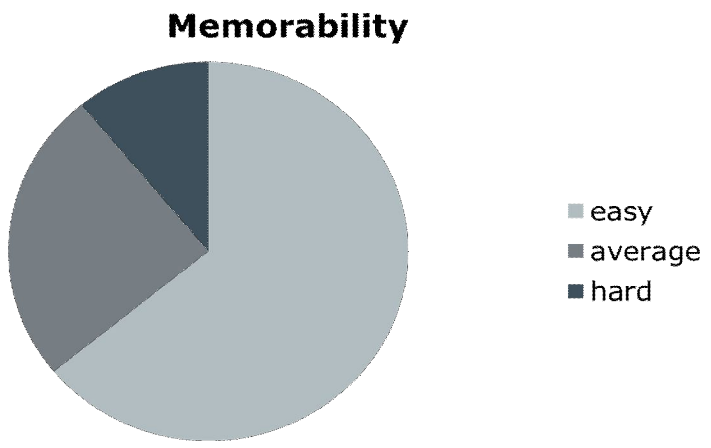
Usability Evaluation

During the testing phase, business users provided feedback on the usability of the system. Usability graphs data represent following criteria’s:

1. Visibility of System Status
2. Match Between System and the Real World
3. User Control and Freedom
4. Consistency and Standards
5. Error Prevention
6. Recognition Rather Than Recall
7. Flexibility and Efficiency of Use
8. Aesthetic and Minimalist Design
9. Help Users Recognize, Diagnose, and Recover from Errors
10. Help and Documentation


Ease of Learning
New
Old


Efficiency of Use
New 1min
old 4min


Memorability
easy
average
hard

Future Enhancements

Gordon Gear POS system meets their existing requirement but business users requested additional functionality for next year if time permits. Some of the future enhancements that are in the radar for next year are:

i. Functionality to add new items
ii. Functionality to delete items if out of stock
iii. Functionality maintain Gift Card Data through the application
iv. Functionality to import flat file with item information
v. Be able to run disconnected mode if internet is down
vi. Be able to secure FTP files to PCI server
vii. Be able to process file automatically to the bank

Lesson Learned

What went well?

- Strong business participation in testing
- Collaboration with customers, rather than contract negotiation, fueled a positive working relationship
- Consistently delivering working software allowed us to definitely know where the project team was at in the project
- Code re-factored midway through the project; this helped set the stage for the later work
- Structured approach to development (end of process toward the start) was well thought out
- Controlled scope
- Schedule focus at the end was good

What could be improved?

- Frequent releases to production require frequent involvement of Quality Assurance and business testing. These teams pushed back wanting not to get involved so frequently.
- Need a focus on testing earlier, especially the iterations that are not going to production
- Need more business participation on a daily basis
- Challenges with WebLogic 12c environment required more deployment to production
- Should have focused more on the schedule throughout the project
- Learning curve for Angualr JS was higher than expected

Conclusions

Gordon Gear POS System is architected with the newest technologies that exists in the market such as, AngularJS, HTML5, Spring Framework and is deployed under Enterprise application server, WebLogic with Oracle database backend. During the design and development phase, the application followed responsive design patterns and W3C guidelines which made the application compatible with both Android and IOS tablets, and phones. Gordon Food Service's Annual meeting committee is very pleased with the system, especially with the platform independent and ease of access from anywhere.

Bibliography

http://support2.microsoft.com/lifecycle/search/?sort=PN&alpha=Visual+basic+6&Filter=FilterNO
http://en.wikipedia.org/wiki/Responsive_web_design
http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model
http://www.gfs.com
https://angularjs.org/
http://www.w3schools.com/html/html5_intro.asp
http://spring.io/