Credit Sesame

Sales Tax Challenge
Software Architecture Document

Version 0.01

Document history

Date	Version	Description	Author
6/17/2016	0.01	Initial version	Nahrin

Distribution

Name	0.01													
Nahrin	x													

Document approval

Credit Sesame

Nahrin Reihaneh

…………………

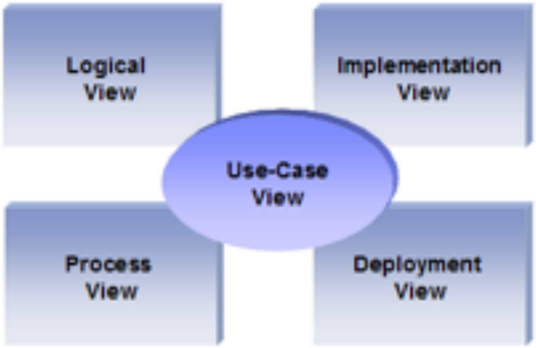
Contents

Introduction

1.Purpose of this document

The Software Architecture Document (SAD) contains the architectural description of Sales Tax Challenge developed by Nahrin Reihaneh. This description consists of various architectural views of the system, in order to highlight the different aspects of it.

The description makes use of the well-known 4+1 view model.



The 4+1 view model enables various stakeholders to establish the impact of the chosen architecture from their own perspective. The Process View (communication of processes) is not a separate chapter but can be found in the chapters 3.3 and 5.

2.References

Title	Version	Author	Location
PRD	0.1	Credit Sesame	
vision			
Product Acceptance Plan			
Use Case Model			

3.Document Overview

Chapter	Reader	Objective
2 Architectural requirements	Software Architect	Overview of architecturally relevant requirements.
3 Logical View	Developer	Knowledge of the application’s conceptual structure, as a basis for technical designs.
4 Implementation View	Developer	Knowledge of the application’s technical structure.
5 Deployment View	System Administrator roles	Knowledge of the way in which the application is deployed and (internal and external) communication takes place.

2.Architectural requirements

2.1.Non-functional requirements

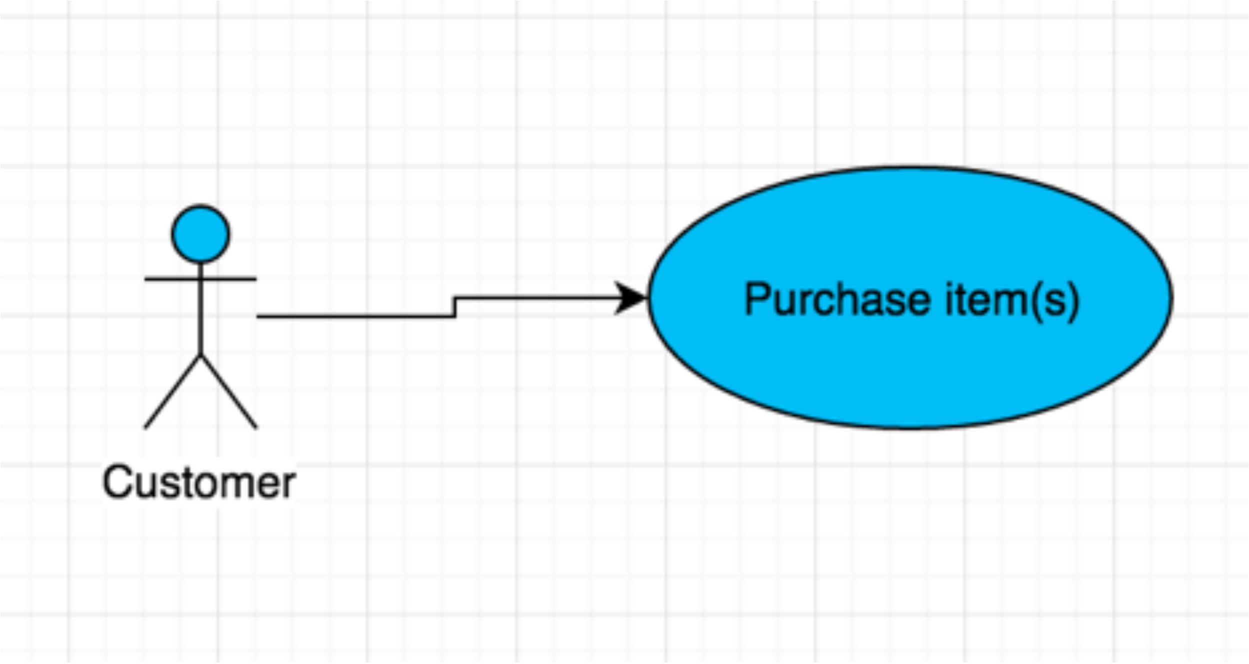
Non-functional requirement example are security, privacy, third-party products, system dependencies, distribution and reuse. Also environmental factors such as context, design, implementation strategy, team composition, development tools, time to market, use of legacy code may be addressed.

Sales Tax Challenge is simple project and the only non-functional requirements for that are development tool you need to use to change or run the program:

Source	Name	Architectural relevance	Addres-sed in:
git	git	source control	https://github.com/nahrinrs/SalexTaxes
Java	Java	Code is developed in Java	
Maven	Maven	To compile and run the project	https://maven.apache.org/download.cgi

2.2.Use Case View (functional requirements)

The use case is simple. A user purchases item(s) and system calculates the sales tax and print the receipt.



3.Logical View

The current application is simple one tier console app with no database and no fancy UI and no Rest API. Mainly represent the business logic and STDIN to input the params.

3.1.Tiers

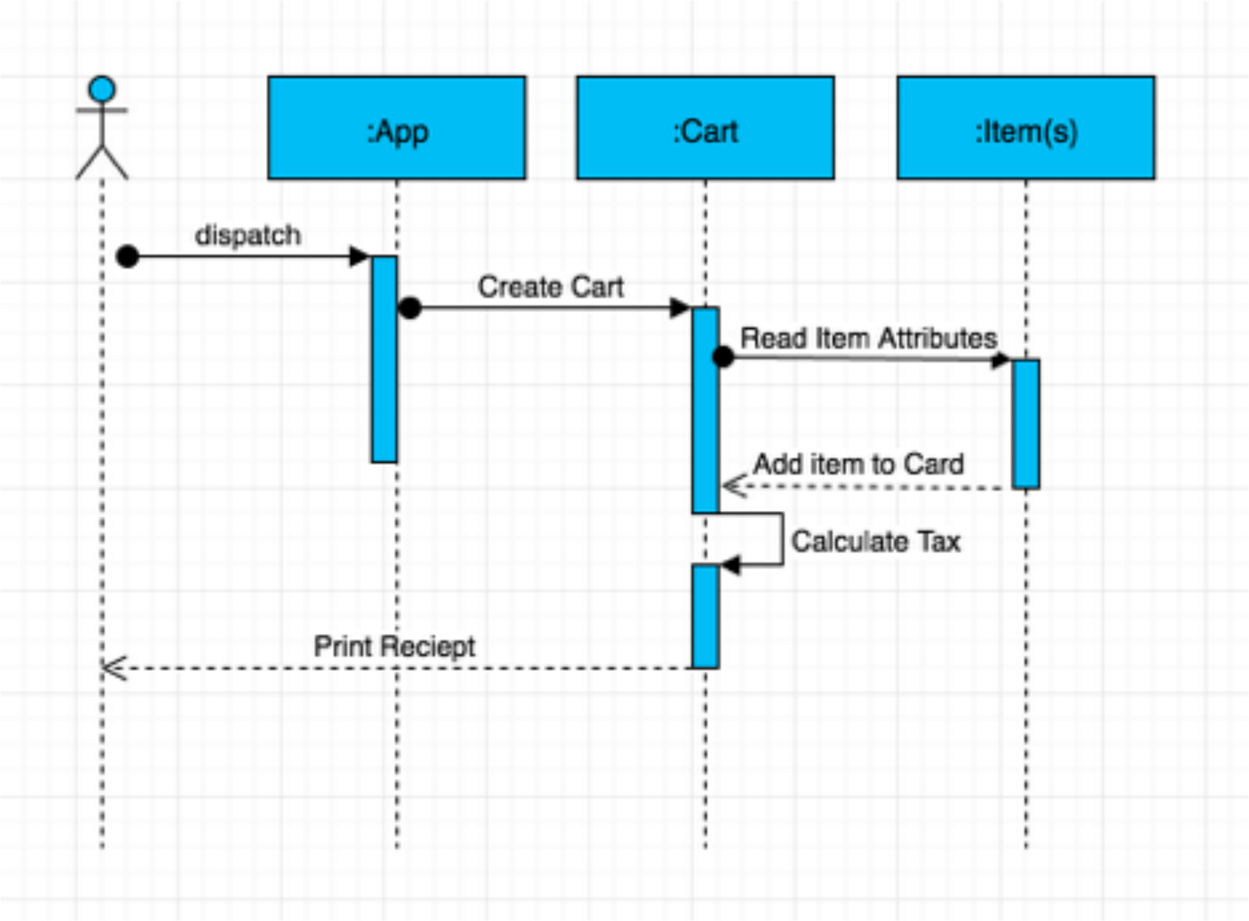
- STDIN to accept the params interactively
- Business Tier to calculate and sales tax and print the receipt

3.2.Subsystems

N/A

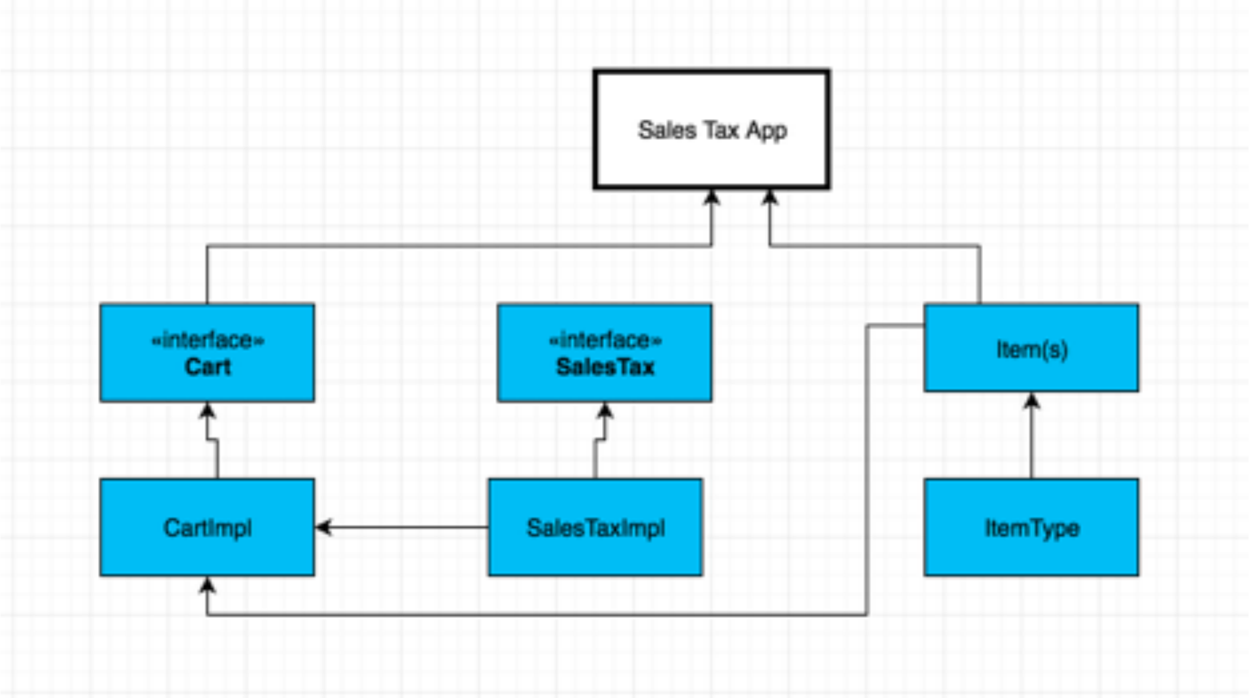
3.3.Use Case Realizations

User using STDIN answer some interactive questions about the item. At the end of each list of questions, item object is created and added to Cart. when all items are created, user enters ‘exit’ or ‘print’ which in any case, receipt will be calculated with sales taxes and printed. if user enters print, the receipt will be printed and system ready to accepts items for next user. See sequence diagram below:



4.Implementation View

This is the class diagram for salesTax Application.



4.1.Structure of the packages

The application is one tier but multiple packages defined in business tier to help to scale the code:

com.creditsesame.challange

com.creditsesame.challange.cart

com.creditsesame.challange.item

com.creditsesame.challange.tax

4.2.Realization of tiers

N/A

4.3.(Re)use of components and frameworks

N/A

5.Deployment View

Deployment is easy for this application. It is considered to run standalone. Check on https://github.com/nahrinrs/SalexTaxes to learn how you can run it locally.

Name	Type	Description
Sales Tax Application	local	See https://github.com/nahrinrs/SalexTaxes