

Donut on Demand, Made to Order Doughnuts on MVC Architecture and UI Testing

Titouan Le Mao
Computer Science
Asian Institute of Technology
st122014@ait.asia

Cedric Le Mercier
Computer Science
Asian Institute of Technology
st122004@ait.asia

I. Introduction

Doughnuts (or Donuts) are made from fried dough confectionery, and have its origin in “Olykoek”, a Dutch cake made of sweetened dough. The word itself means oil cakes, which are usually consumed as dessert. Made in homes or purchased in outlets, they are very popular among kids and young people in general. Donuts can be eaten anytime, anyplace and with about anything, from sweet, chocolate flavored snack donut to paired to the bun on a chicken sandwich. Additionally, they can be complemented with a wide range of topics, including chocolate, cereal to pork. According to the report on the Global Doughnuts Market [1], the global market of doughnuts reached USD 40 billion in 2016, with Asia Pacific reaching USD 8 billion, putting Asia Pacific as the second largest region after North America. One of the main factors of growth is the increase in disposable income in households, increase in habit in munching, changing lifestyle in urban cities, while restricting factors include a healthy lifestyle. Doughnuts vendors can increase the number of visits at their outlets by offering a larger number of selections, such as flavour, topping, type and more.

While doughnut comes in all forms and flavours and some types can be bought almost anywhere and anytime, we can't always get the flavour we want once we get in store because they often run out of stock. Additionally, there is no centralized place to browse a larger selection or for local store owners to offer handmade products and get themselves known.

As donut lovers, our company Donuts TC™ aims to make the order and pick up of donuts easier while offering customers a centralized place to explore different options to suit different tastes and occasions. We want to set up an innovative service for our customers, at the crossroads of the eShop and the drive-in. This

service, called “Donuts on Demand” (DoD), will allow customers to make their selection from our range of donuts, place their order and specify the date and time they will pick them up. Ordering on Donuts on Demand ensures that you always get your favorite cookie recipe, right out of the oven, on time. No more queues in the store, no more unpleasant surprises such as chocolate being out of stock. Donuts always fresh, available in less than two hours and recovered while still hot for even more gourmet family moments

II. Related Work

A. The Doughnut Industry and Competitors

In Thailand, one of the closest examples we want to achieve in a different industry is the industry of cakes and cupcakes. For example, GoTasty offers a selection of [custom made cupcakes](#) to suit various occasions such as birthdays. Hobby Cake offers a high customization of birthday cakes and can be ordered online for delivery or pickup. The process is as simple as describing the cake with words on LINE chat, sending a reference picture or even just a drawing.

One of the most well-known donut sellers in Thailand and internationally is Krispy Kreme, who offers a [good selection of doughnuts online](#). However, the number of options are standardized, limited and not handmade.

One of the closest examples of “Donut on Demand” is US-based [Duck Donuts](#), a specialised made-to-order donut company who offers a large selection of existing recipes as well as an option to make our own donuts by choosing our dough, coating, topping, drizzle and more. The donuts ordered can be delivered or picked up at a store.

Our work will implement a similar functionality for our customers to browse a large selection of doughnuts not only from one manufacturer or seller but from many, from local handmade stores

to larger chains. Our application will focus on customers in Thailand but can be adapted for other parts of Asia Pacific in the future.

B. High level design issues in e-commerce applications development

The application that we aim to build can be compared to a typical e-commerce store. As a result, we expect to encounter the same architectural design issues that a typical online store will have. These issues can however, be addressed in various ways. Ehikioya and Guillemot [3] discussed high level design issues in e-commerce systems development and as well as common workarounds.

Security

Security is paramount in ecommerce applications as they typically require users to create accounts and store their personal information in the server's database, including financial details. Many customers log in from unsafe access points [3] (e.g cafés, public Wifi, airport, shopping malls, workplaces...) as well as unsafe devices, so it is important to take every step possible to prevent attacks. These measures can be demanding on computational power, so every transaction or data do not need to be highly secured. For example, during shopping, the shopping cart information does not necessarily need a lot of protection, whereas the customer's financial details do need to be highly protected. According to Sherif [4], there are three issues to be addressed by designers in security: authentication, nonrepudiation (denying being an actor in transaction) and confidentiality (unauthorised access to certain data) with sophisticated cryptographic techniques. Finally, a good principle to follow is the ACID protocol, which stands for atomicity, consistency, isolation and durability.

Communications and Networking

ECommerce applications are also typically very network intensive which makes factors such as performance and bandwidth to be highly important as well as network security. From a back-end perspective, it is important to limit the number of API calls or database queries on a single webpage, as well as limiting the number of media (images, videos, files) displayed on the front-end of the application. According to Gehrke and Turban [5], the most common complaint

among customers is the loading speed of webpages, which can be the determinant factor in whether a first time visitor will convert into a paying customer or go visit a competitor. To address these issues, it is suggested to implement caching mechanisms for most frequently accessed pages or queries [3], offering two versions of a website for desktop and mobile (e.g with a domain starting with an "m", "m.website.com") or maintaining user session details by passing parameters from one page to the next such as cookies.

Languages and tools

E-Commerce often uses a wide range of languages, with Java, ASP.NET, Javascript, PHP and HTML as the most common ones [6]. The choice of language is important in architecture design because a designer has to foresee future long term changes in the system, and with new languages, tools, frameworks that are emerging rapidly, it becomes more difficult to choose the correct combination.

Database

Among Relational Database Management Systems, MySQL is the most common one and is used to store data persistently. Every e-Commerce application must adhere to the ACID principles, but atomicity is the most difficult one [3]. For example a checkout process involves moving items from a shopping cart table to the order table, which splits the checkout transaction into two insert transactions. Doing so can put the database into an inconsistent state [3]. To address this, some companies are using a separate transaction server, which also helps in alleviating the load on the rest of the application system. Some other database models such as NoSQL (MongoDB or MariaDB) are gaining popularity as well and can offer better performance than traditional relational databases for certain types of data. Finally, e-Commerce applications usually record large numbers of transactions, often too much for one server. To counter this, we can use distributed databases, which allows for better throughput by for example, moving copies of tables closer geographically to certain customers and implementing caching.

III. Design

A. Three-Tier Architecture

Our system will implement one of the most predominant software architectures in client-server, a three-tier architecture [2]. This will device our software into three logical tiers:

The presentation tier or user interface

This will be our top-level tier, our user interface and what our stakeholders will see. The main purpose of this tier is to display information and allows the user to interact indirectly with the application logic tier below. Our top level tier will run in a web browser and will be fully responsive to the user's device. It will use HTML but with a Java templating engine, CSS and some Javascript. For example, this layer will:

- Display a list of products to customers
- Display a dashboard for managers and supervisors

The application tier

Also called the controller, our application tier provides business logic and data access. It is the heart of our application where the information collected from the presentation tier is processed against the data tier using business logic.

For example, when a customer visits the website homepage, the application layer retrieves a relevant list of products from the data tier and transmits the data to the presentation layer which presents the information in a readable format.

Our application tier will be fully developed in Java with the Spring Boot Framework.

This will include controllers, services, models, JPA and Data Access Object.

The data tier

The database tier or backend is where all the information entered by stakeholders, customers or managers is stored in a permanent way. It is important to note that for security reasons as well as best practices, the presentation and data tier will not be able to communicate directly with each other.

We will use a Relational Database Management system PostgreSQL as it will fit the database structure of a typical online store.

B. Testing

Although our application project resembles a typical e-commerce application with its usual architectural challenges, we did not focus on a design issue in particular. However, as we discussed in related works, security is particularly important in e-commerce websites, as well as performance. Performance will be important in the first days of launch because it can determine if a prospective customer will buy on our site or try visiting a competitor.

For testing, we focused on feature testing, which is testing our app like how a user would. This will be particularly useful in the future for checking if certain users can access or do certain actions on pages. Testing this

Basic database performance testing will be done in JMeter, while web user interface testing using Selenium WebDriver.

Selenium is a browser automation tool that can work in conjunction with various unit test frameworks such as Python's unittest, which is the one we will be using. Python is the scripting language we chose and we are going to run tests on Google Chrome's browser. Our unit tests are going to run in Page Objects which are going to do webdriver calls and run an executable that is going to run our commands in Chrome Browser. (Figure 1)

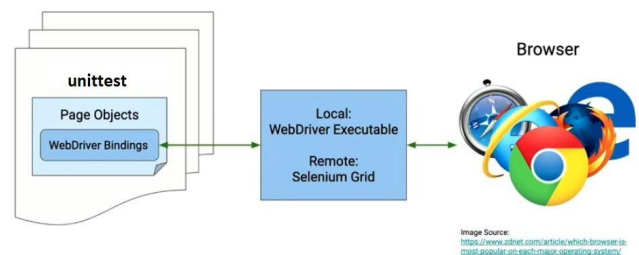


Figure 1. Working with Selenium and Unittest

IV. Results

Even if our application is a simple e-commerce platform and does not focus on a specific software architecture problem, we are still globally satisfied with the final result of our application since it is functional and well tested. Indeed, the different functionalities planned at the beginning have been implemented and tested from start to finish.

V. Discussion & Conclusion

A very complete project that allowed us to acquire a lot of knowledge in many areas of software development. We were able to put into practice almost all of the knowledge taught during this course and it brought us a lot on a personal and professional level. Moreover, the learning of Selenium was very interesting and will certainly be useful for future projects. Finally, we also learned how to organize ourselves for the development of an application with the help of different very interesting and useful tools.

References

- [1] Goldstein Market Intelligence, Global Doughnuts Market Report 2030, in *On the basis of products, On the basis of types, By Region With COVID-19 impact | Forecast Period 2017-2030*, 2020.
- [2] IBM Cloud Education, Three-Tier Architecture, 2020, <https://www.ibm.com/cloud/learn/three-tier-architecture>
- [3] Ehikioya, SA, Guillemot, E. A critical assessment of the design issues in e-commerce systems development. *Engineering Reports*. Vol 2. Issue 4, 2020
- [4] Sherif MH. Protocols for Secure Electronic Commerce. 3rd ed. Boca Raton, FL: CRC Press; 2016.
- [5] Gehrke D, Turban E, *Determinants of Successful Website Design: Relative Importance and Recommendations for Effectiveness*, Proceedings of the 32nd Hawaii International Conference on System Sciences, IEEE, 1999
- [6] Reynolds M. *Beginning E-Commerce with Visual Basic*, ASP, SQL Server 7.0 and MTS. Birmingham, England: Wrox Press Ltd; 2000
- [7] Laudon KC, Traver CG *.e-Commerce 2018: Business, Technology, Society*. 14th ed. Boston, MA: Pearson; 2019.